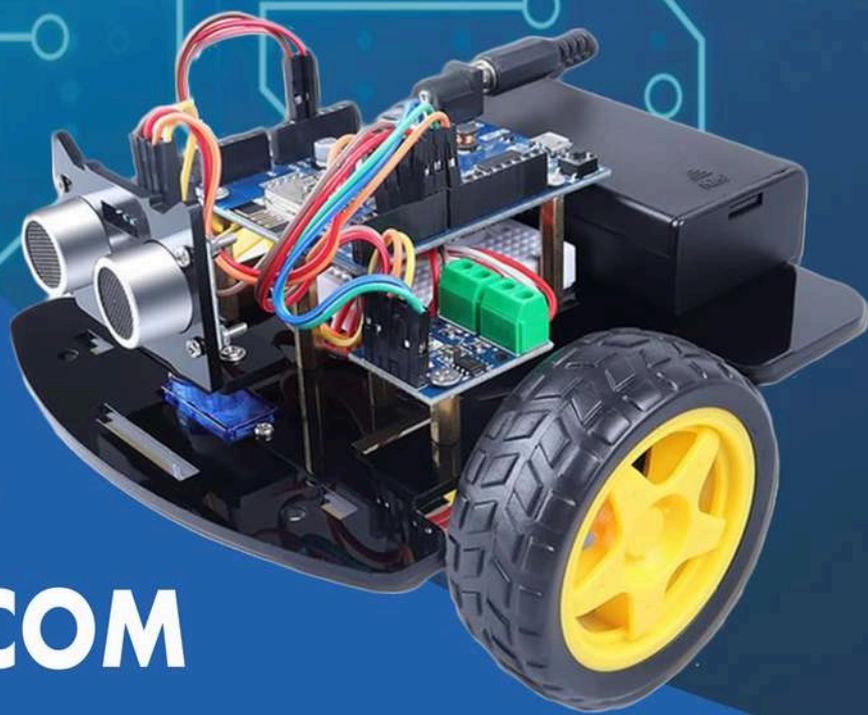


De acordo com a
BNCC da computação.

**Caderno
Metodológico**

INTRODUÇÃO À ROBOTICA COM ARDUÍNO



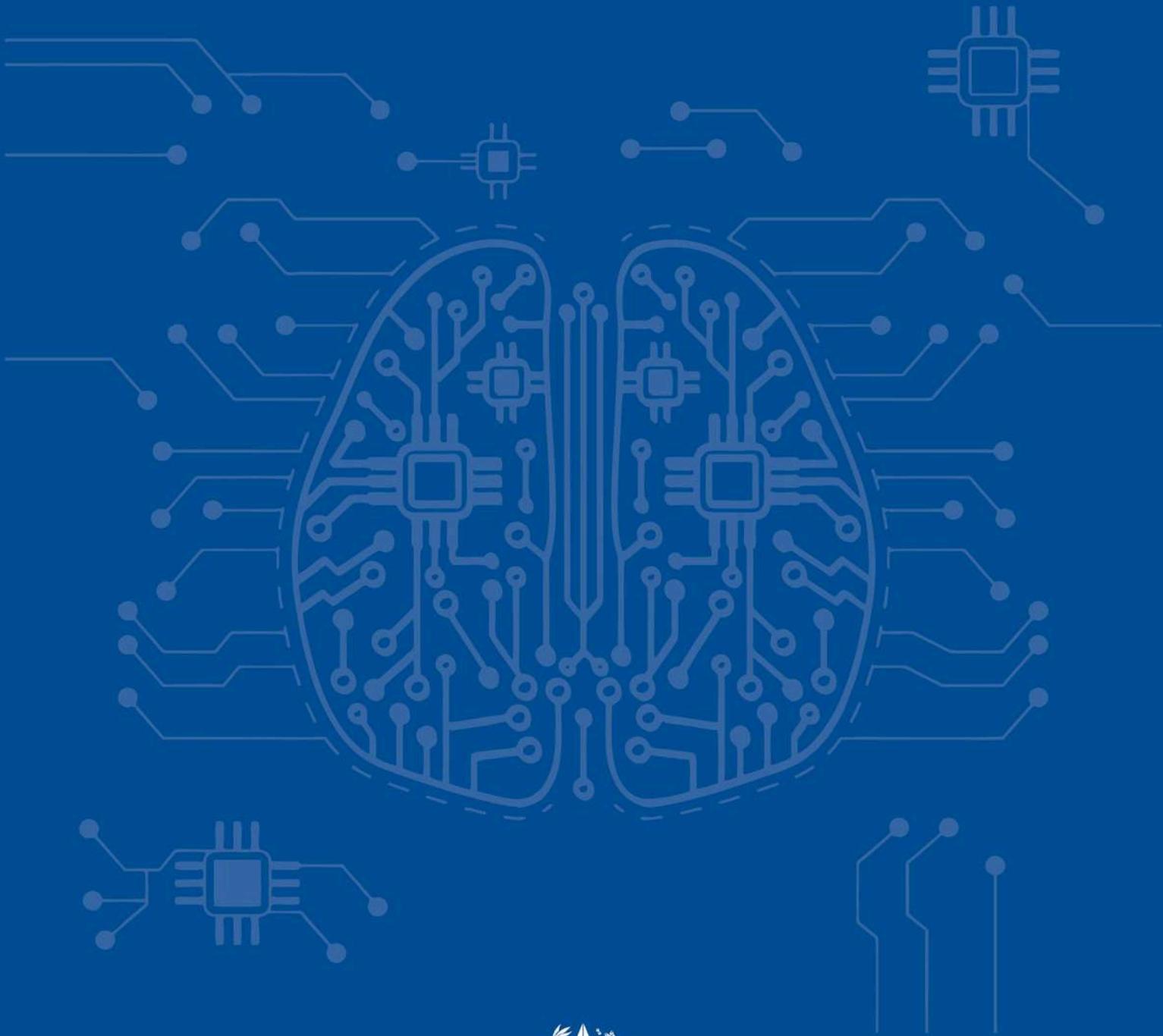
Gerência de Currículo
da Educação Básica

GOVERNO DO ESTADO
DO ESPÍRITO SANTO
Secretaria da Educação





Gerência de Currículo
da Educação Básica



GOVERNO DO ESTADO
DO ESPÍRITO SANTO
Secretaria da Educação

Dados Internacionais de Catalogação-na-publicação (CIP)
(Governo do Estado do Espírito Santo, ES, Brasil)

E77i **Espírito Santo (Estado). Secretaria de Educação.**
Introdução à robótica com arduino [livro eletrônico] / Organizadores Aleide Cristina
Camargo; Andréa Guzzo Pereira; Júlio Cesar Souza Almeida; Vitor Amorim de Ângelo.
Vitória, ES: GECEB/SEDU, 2024.

3.651 Kb - (Caderno Metodológico)
Bibliografia
ISBN: 978-65-85134-75-0

1. Educação - Espírito Santo (Estado). 2. Educação Básica. 3. Robótica. 4. Arduino
I. Camargo, Aleide Cristina. II. Pereira, Andréa Guzzo III. Almeida, Júlio Cesar Souza
IV. Ângelo, Vitor Amorim de V. Título.

CDD: 370
CDU: 37

GOVERNADOR DO ESTADO DO ESPÍRITO SANTO

José Renato Casagrande

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Vitor Amorim de Angelo

SUBSECRETÁRIA DE ESTADO DA EDUCAÇÃO BÁSICA E PROFISSIONAL

Andréa Guzzo Pereira

GERENTE DA GERÊNCIA DE CURRÍCULO DA EDUCAÇÃO BÁSICA

Aleide Cristina De Camargo

SUBGERENTES

Aldete Maria Xavier

Marcos Valério Guimarães

TÉCNICOS EDUCACIONAIS

ARTE

Inara Novaes Macedo

BIOLOGIA

Luciane da Silva Lima Vieira

Vinícius Brito Lima

EDUCAÇÃO FÍSICA

Vinnicius Camargo de Souza Laurindo

ENSINO RELIGIOSO/FILOSOFIA

Aline Eduardo Machado

FÍSICA

Julio Cesar Souza Almeida

LÍNGUA ESPANHOLA

Mônica Nadja Caniçali

GEOGRAFIA

Wanderley Lopes Sebastião

HISTÓRIA

João Evangelista de Sousa

LÍNGUA PORTUGUESA

Fernanda Maia Lyrio

Maria Eduarda Scarpat

Mariana de Castro Atallah

MATEMÁTICA

Gabriel Luiz Santos Kachel

Wellington Rosa de Azevedo

Laiana Meneguelli

QUÍMICA

Thaís Scárdua Rangel Garcia

SOCIOLOGIA

Aldete Maria Xavier

ORGANIZADORA

Aleide Cristina de Camargo

Andréa Guzzo Pereira

Vitor Amorim de Angelo

PROJETO GRÁFICO

Dianni Pereira de Oliveira

DIAGRAMAÇÃO

Julio Cesar Souza Almeida

PROFESSOR AUTOR

Julio Cesar Souza Almeida

BIBLIOTECÁRIOS

Gabriel de Menezes Oliveira

Joice Rodrigues Teixeira

Mariene Kohler

Roberta Dalfior Cola

Sarah Garcia Fernandes Vargas

Victor Barroso Oliveira

índice

Apresentação



Documento Interativo

Click na página desejada;

Boa navegação!

A Importância da Robótica com Arduino no Ensino Médio: Desenvolvendo Competências para o Futuro.....	09	
A história do Arduino.....	10	
Conhecendo a plataforma Arduino.....	12	
Conhecendo a placa Arduino.....	13	
Componentes da plataforma Arduino.....	14	
O que são resistores?.....	16	
O que é um LED?.....	18	
Sensor de Temperatura.....	19	
Sensor Ultrassônico.....	20	
Sensor LDR.....	21	
Sensor de Umidade.....	22	
Dispositivo Buzzer.....	23	
Software de Programação.....	24	
Software de Programação.....	25	



Documento Interativo

Click na página desejada;

Boa navegação!

índice

Orientações Pedagógicas

Robótica, Programação e Pensamento computacional	29
Competências em Computação da BNCC.....	31
Temas Integradores e a Robótica.....	34



Documento Interativo

Click na página desejada;
Boa navegação!

índice

Projetos

Projeto 01: Piscar um LED com Arduino.....	36
Projeto 02: Ligar Led enquanto o Push-button ficar pressionado.....	43
Projeto 03: Push-button como interruptor.....	50
Projeto 04: Criando um Semáforo.....	57
Projeto 05: Sensor de Temperatura.....	64
Projeto 06: Sensor Ultrassônico HC-SR04.....	71
Projeto 07: Sensor LDR.....	78
Projeto 08: Sensor Umidade do solo com LED.....	85
Projeto 09: Piano com Buzzer.....	92
Projeto 10: Músicas no Arduino.....	100
Projeto 11: Usando IA como programador.....	113

Apresentação

E aí, Professor,

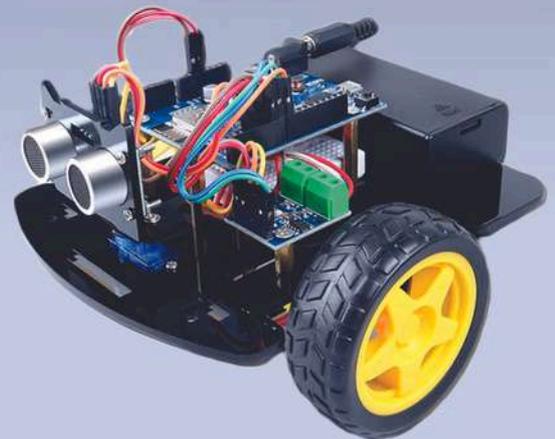
O Governo do Estado do Espírito Santo, por meio da Secretaria de Estado da Educação (SEDU-ES) e também da Gerência de Currículo da Educação Básica, vem com entusiasmo, apresentar nosso Caderno de Introdução à Robótica utilizando Arduino. Em um mundo cada vez mais impulsionado pela tecnologia, a robótica se destaca como uma das áreas mais emocionantes e inovadoras, oferecendo oportunidades infinitas para a criatividade para a descoberta.

Este material, foi cuidadosamente elaborado para proporcionar a você uma experiência imersiva e prática no universo da robótica. Com uma abordagem acessível e passo a passo, você será guiado desde os conceitos mais básicos até a construção e programação de projetos robóticos fascinantes.

Com Arduino como nossa plataforma principal, você terá a oportunidade de mergulhar de cabeça no mundo da eletrônica e da programação, desenvolvendo habilidades essenciais que o prepararão para os desafios do futuro. Iremos começar com a montagem de circuitos simples, mostrando como funcionam alguns sensores importantes dentro da plataforma Arduino, mas nos próximos cadernos pretendemos abordar a criação de robôs autônomos, além da utilização de outras plataformas como Scratch, Spike Prime. Cada projeto apresentado neste suplemento foi cuidadosamente projetado para oferecer uma experiência prática e gratificante.

Vale ressaltar que os projetos apresentados nesse caderno, estão plenamente de acordo com as diretrizes da educação propostas pela Base Nacional Comum Curricular (BNCC) e alinhados com o Currículo do Espírito Santo.

Estamos confiantes de que o material elaborado pela Gerência de currículo da Educação Básica, será uma fonte valiosa de conhecimento e inspiração para todos os entusiastas da tecnologia, sejam eles novatos ou experientes. Que esse seja um ponta pé inicial, para desenvolvermos projetos maravilhosos entre os professores da Rede Estadual de Ensino do Espírito Santo. Portanto, não hesite em explorar esse mundo de possibilidades, nesta jornada emocionante pela robótica com Arduino. O futuro está em suas mãos, é hora de começar a explorar!



A Importância da Robótica com Arduino no Ensino Médio: Desenvolvendo Competências para o Futuro

A introdução da Robótica no Ensino Médio por meio da plataforma Arduino representa um passo significativo na evolução do processo educacional. Essa abordagem vai além do tradicional ensino teórico, proporcionando aos estudantes uma experiência prática e interdisciplinar, unindo conceitos de física, matemática, eletrônica e programação.

Ao adentrar o universo da Robótica, os estudantes não apenas absorvem informações teóricas, mas também aplicam esse conhecimento na construção e programação de dispositivos reais. O principal instrumento para essa imersão é o Arduino, uma plataforma acessível e versátil que facilita a compreensão dos princípios fundamentais da Robótica.

Um dos benefícios imediatos dessa abordagem é o desenvolvimento de habilidades práticas. Os estudantes não apenas aprendem informações passivamente, mas são desafiados a resolver problemas, aprimorar o pensamento lógico e explorar a criatividade. O famoso "Hello World" da eletrônica, ao fazer um LED piscar, torna-se a porta de entrada para um vasto campo de possibilidades.

Além disso, a Robótica no Ensino Médio prepara os estudantes para os desafios do futuro. Em uma era permeada por avanços tecnológicos rápidos, compreender e moldar a tecnologia é uma habilidade crucial. A capacidade de programar, entender circuitos e interagir com dispositivos eletrônicos não apenas aumenta a alfabetização digital, mas também promove a autonomia dos estudantes.

Outro aspecto valioso é o estímulo ao trabalho em equipe. Muitos projetos robóticos exigem a colaboração de diferentes habilidades, incentivando os estudantes a compartilharem ideias e conhecimentos. Essa colaboração não apenas reforça as habilidades sociais, mas também reflete a dinâmica do ambiente profissional contemporâneo.

A Robótica no Ensino Médio transforma conceitos abstratos em realizações tangíveis. Ao conectar teoria à prática, os estudantes percebem a aplicação direta do conhecimento, tornando o aprendizado mais envolvente. Essa abordagem também fomenta a curiosidade e a inovação, incentivando os estudantes a questionar, explorar e criar soluções para desafios práticos.

Portanto, A Secretaria Estadual de Educação do Espírito Santo, apresenta através desse material, a introdução da Robótica com Arduino no Ensino Médio não apenas diversifica a abordagem educacional, mas também desenvolve habilidades essenciais para os cidadãos do futuro. Essa experiência prática, aliada ao estímulo à criatividade e à resolução de problemas, contribui para a formação de indivíduos preparados para enfrentar os desafios tecnológicos da sociedade contemporânea. Vamos, juntos, explorar as infinitas possibilidades que a Robótica com Arduino nos oferece!





A história do Arduino

A plataforma Arduino tem uma história fascinante que remonta ao início dos anos 2000, quando um grupo de pesquisadores do Interaction Design Institute Ivrea, na Itália, liderado por Massimo Banzi, David Cuartielles, David Mellis, Tom Igoe e Gianluca Martino, buscava uma maneira acessível de ensinar eletrônica para estudantes de design e arte interativa.

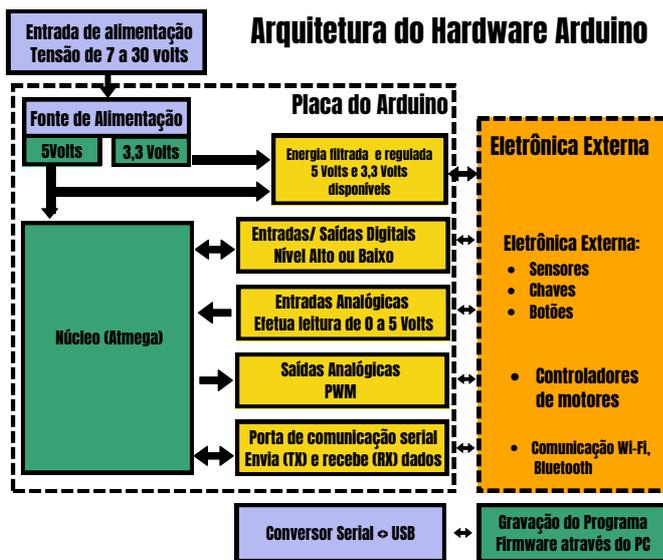
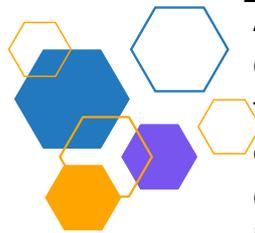
O nome "Arduino" foi inspirado no nome de um bar próximo ao instituto onde os pesquisadores se encontravam frequentemente. Em 2005, o primeiro hardware Arduino, o Arduino Diecimila, foi lançado. Ele era baseado em um microcontrolador Atmel AVR e vinha com uma interface USB integrada, tornando-o muito mais fácil de programar e usar do que as placas de desenvolvimento disponíveis na época.



Bar na comuna italiana de Ivrea

A comunidade em torno do Arduino cresceu rapidamente, impulsionada pela filosofia de código aberto e pela facilidade de uso da plataforma. Os usuários compartilhavam projetos, tutoriais e bibliotecas de código, contribuindo para a expansão e diversificação do ecossistema Arduino.

Desde então, o Arduino evoluiu para uma plataforma versátil e poderosa, com uma ampla gama de placas, módulos e acessórios disponíveis para uma variedade de aplicações. Ele é amplamente utilizado em projetos de hobby, educação, arte interativa, automação residencial, Internet das Coisas (IoT) e muito mais. Na figura ao lado temos a estrutura de desenvolvimento da plataforma Arduino.



Disponível em:

<https://www.unifatecpr.com.br/conteudo/Revista%20Tecnologica%20da%20FATEC-%20PR%20V%201%20N%205-%20JAN-DEZ%202014.pdf>

Código aberto na robótica refere-se ao uso de software e hardware cujos designs e códigos-fonte são públicos, permitindo que qualquer pessoa os utilize, modifique e distribua.



VOLTAR AO SUMÁRIO



Conhecendo a plataforma Arduino



A plataforma Arduino é muito mais do que uma simples placa de circuito; é uma caixa de ferramentas versátil que desbloqueia as portas da eletrônica criativa. Ao conhecermos os elementos fundamentais desta plataforma, estaremos abrindo as portas para um mundo de possibilidades e inovação. Vamos desvendar o que torna o Arduino tão único e como ele se torna a ponte entre a teoria e a prática.

Placa Arduino Uno

A placa Arduino é uma plataforma de prototipagem eletrônica extremamente versátil e popular, usada em uma variedade de aplicações. A sua principal função é fornecer uma interface acessível e amigável para a criação de projetos eletrônicos, permitindo que entusiastas, estudantes, e profissionais desenvolvam e testem ideias rapidamente. Aqui estão algumas das principais finalidades e usos da placa Arduino:

✓ Prototipagem Rápida:

A placa Arduino é projetada para permitir a prototipagem rápida de projetos eletrônicos. Com a sua estrutura de furos e a facilidade de conexão com jumpers, é possível criar circuitos temporários sem a necessidade de soldagem.

✓ Aprendizado de Eletrônica e Programação



É uma ferramenta educacional valiosa para o aprendizado de conceitos de eletrônica e programação. O Arduino utiliza uma linguagem de programação baseada em C/C++, tornando-a acessível para iniciantes. Ela também aceita as linguagens de programação, Python ou Java.

✓ Desenvolvimento de Projetos DIY (Faça Você Mesmo)



A comunidade DIY (**Do It Yourself**) significa “**Faça Você Mesmo**” utiliza a placa Arduino para criar uma ampla variedade de projetos, desde simples experimentos até dispositivos mais complexos, como robôs, controladores de automação residencial, sensores, entre outros.

✓ Internet das Coisas (IoT)



O Arduino é amplamente utilizado em projetos de IoT, permitindo a criação de dispositivos conectados à internet. Pode ser integrado com módulos de comunicação sem fio para enviar e receber dados pela internet.



Conhecendo a plataforma Arduino

✓ Controle de Motores:

É comumente utilizado para controlar motores, como motores de passo ou servo motores, tornando-o ideal para projetos robóticos e de automação.

✓ Experimentação com Sensores:

A placa Arduino é compatível com uma variedade de sensores, como sensores de luz, temperatura, umidade, acelerômetros, giroscópios, entre outros. Isso permite a criação de projetos que respondem ao ambiente ao seu redor.

✓ Desenvolvimento de Jogos e Interfaces Interativas:

Pode ser usado para criar jogos eletrônicos simples, interfaces interativas e instalações artísticas que envolvem interação física.

✓ Ensino e Pesquisa Acadêmica:

Muitas instituições acadêmicas utilizam a placa Arduino como uma ferramenta prática para ensino e pesquisa em cursos de engenharia, ciência da computação e disciplinas relacionadas.

✓ Automação de Processos:

Em ambientes industriais, a placa Arduino pode ser utilizada para automatizar processos simples e controlar dispositivos.

✓ Desenvolvimento de Protótipos para Startups:

Startups muitas vezes usam a placa Arduino para desenvolver protótipos iniciais de seus produtos de maneira rápida e econômica. Desta forma, a placa Arduino serve como uma base flexível e acessível para a criação de uma ampla variedade de projetos eletrônicos, promovendo a inovação, a experimentação e a aprendizagem prática.



Conhecendo a placa Arduino.

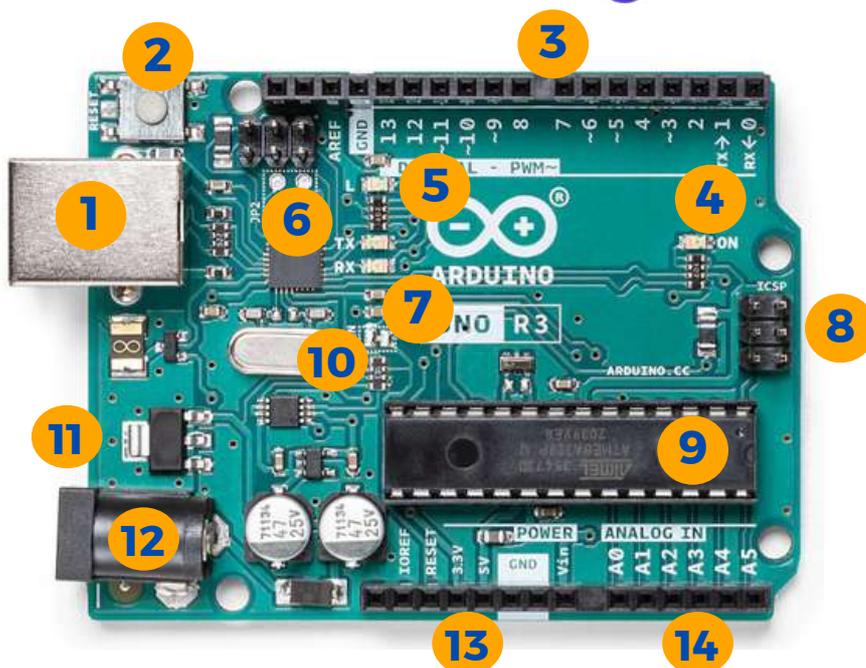


Figura Disponível em: <https://www.multilogica-shop.com/>

- 01 Conector USB para o cabo tipo AB
- 02 Botão de reset
- 03 Pinos de entrada e saída digital e PWM
- 04 LED verde de placa ligada
- 05 LED laranja conectado ao pin13
- 06 ATmega encarregado da comunicação com o computador
- 07 LED TX (transmissor) e RX (receptor) da comunicação serial
- 08 Porta ICSP para programação serial
- 09 Microcontrolador ATmega 328, cérebro do Arduíno
- 10 Cristal de quartzo 16Mhz
- 11 Regulador de voltagem
- 12 Conector fêmea 2,1mm com centro positivo
- 13 Pinos de voltagem e terra
- 14 Entradas analógicas



VOLTAR AO SUMÁRIO



Componentes da plataforma Arduino.

Microcontrolador:



Microcontrolador

No coração do Arduino está o microcontrolador, uma peça fundamental que processa as instruções dos programas. Compreenderemos como o microcontrolador é o cérebro por trás de nossos projetos, tornando-os inteligentes e interativos.

Figura Disponível em: <https://www.multilogica-shop.com/>

Protoboard

A protoboard, também conhecida como breadboard (placa de ensaio, em português), é uma ferramenta fundamental e versátil em projetos de eletrônica. Trata-se de uma placa perfurada com uma série de furos nos quais os componentes eletrônicos podem ser inseridos e conectados sem a necessidade de soldagem. Essa característica torna a protoboard uma plataforma ideal para criar protótipos e testar circuitos rapidamente.

Protoboard

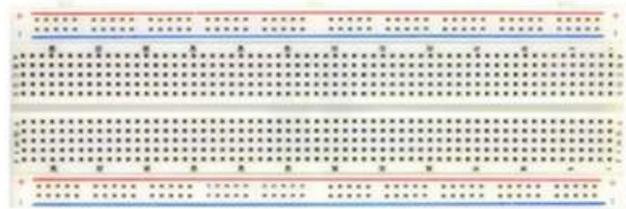


Figura Disponível em: <https://www.multilogica-shop.com/>

Características da Protoboard

✓ Furos e Trilhas:

A superfície da protoboard é coberta por uma série de furos interconectados por trilhas condutoras. Cada furo é conectado a outros furos adjacentes por meio dessas trilhas, permitindo a formação de conexões elétricas.

✓ Alimentação:

Na protoboard, há geralmente duas faixas de trilhas ao longo das bordas que são dedicadas à alimentação (geralmente 5V e GND), permitindo uma distribuição eficiente de energia para os componentes conectados.

✓ Facilidade de Uso:

Não há necessidade de soldagem na protoboard. Os componentes são encaixados nos furos, e os fios de ligação (jumpers) podem ser usados para estabelecer conexões entre os componentes, tornando o processo de montagem e desmontagem extremamente simples e rápido.

✓ Reutilização:

A natureza semipermanente da protoboard permite que os componentes sejam facilmente removidos e reutilizados em diferentes projetos. Isso é particularmente útil durante a fase de prototipagem, onde a configuração do circuito pode ser alterada frequentemente.

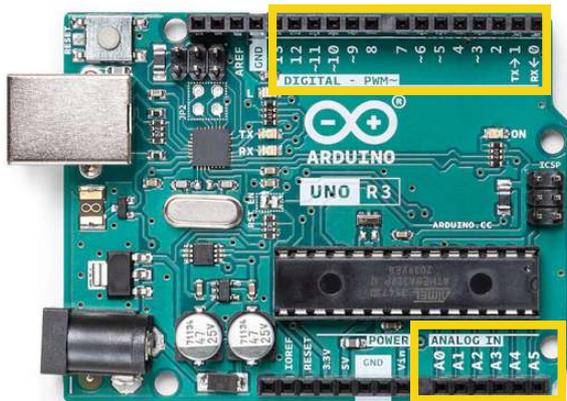


VOLTAR AO SUMÁRIO



Pinos de Entrada/Saída:

Entradas/ Saídas digitais



Entradas/ Saídas digitais

Pinos de Entrada/Saída:

Os pinos de entrada/saída são os nervos periféricos do Arduino. Vamos explorar como esses pinos permitem a comunicação entre o Arduino e o mundo exterior. Os pinos digitais são como interruptores, enquanto os analógicos oferecem uma gama mais ampla de valores.

Componentes Básicos:

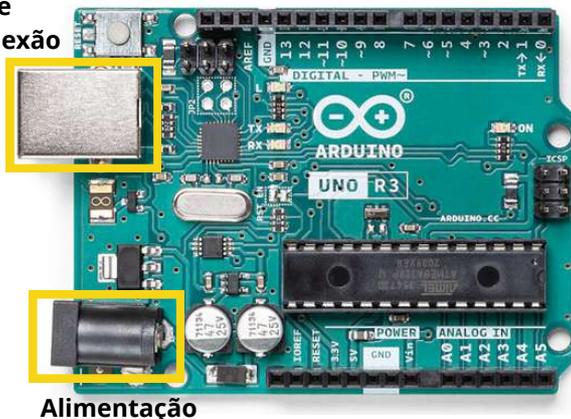
Ao examinarmos os componentes básicos, como resistores e LEDs, entenderemos como criar circuitos simples. Estes componentes são os blocos de construção para projetos mais complexos, e dominá-los é essencial para a criação eficaz. Em nosso caderno iremos trabalhar com vários desses componentes.

Componentes Básicos:



Alimentação e Conectividade:

Alimentação e Conexão



Alimentação

Alimentação e Conectividade:

Abordaremos os métodos de alimentação do Arduino e como conectá-lo ao computador. Esses passos são cruciais para a execução de projetos, pois garantem que o Arduino esteja energizado e pronto para receber programas.

Disponível em: <https://forum.arduino.cc/t/alimentar-arduino-com-fonte-chaveada-5v-5a/574872/6>



VOLTAR AD SUMÁRIO



O que são resistores?

Resistores



São componentes eletrônicos passivos que desempenham um papel fundamental na construção e no funcionamento de uma vasta gama de dispositivos eletrônicos. Sua principal função é controlar o fluxo de corrente elétrica em um circuito, oferecendo uma resistência específica medida em ohms (Ω). No entanto, sua utilidade vai muito além desse papel básico, e eles são amplamente empregados em uma variedade de aplicações dentro do mundo da eletrônica.

São frequentemente utilizados para limitar a corrente elétrica que flui por meio de um circuito. Esta função é crucial para proteger componentes mais sensíveis de correntes excessivas que poderiam danificá-los. Além disso, os resistores desempenham um papel vital como divisores de tensão, permitindo criar tensões proporcionais em diferentes partes de um circuito, o que é essencial em muitas aplicações de controle e medição.

Outra aplicação importante é o controle de corrente e tensão em um circuito. Eles são frequentemente utilizados para ajustar ou regular a corrente ou a tensão, permitindo que um circuito opere dentro de parâmetros específicos. Além disso, em circuitos de filtragem, os resistores são empregados juntamente com capacitores e indutores para filtrar sinais indesejados ou ajustar a resposta em frequência do circuito, o que é crucial em aplicações como áudio, comunicações e processamento de sinais.

Como fazer a medida dos resistores?

A medição dos valores de resistores por meio das cores das faixas é feita utilizando o código de cores dos resistores. Cada cor representa um dígito numérico, e a posição da cor determina o valor do dígito. O código de cores geralmente consiste em três ou quatro faixas coloridas, seguidas por uma faixa opcional que indica a tolerância do resistor. Aqui está um resumo básico de como interpretar o código de cores:

Identificação das faixas: Normalmente, as três primeiras faixas representam os dígitos significativos do valor da resistência, enquanto a última faixa, se presente, representa a tolerância do resistor.

Leitura dos dígitos: Cada cor corresponde a um número de 0 a 9, com uma cor específica associada a cada número. Por exemplo, preto = 0, marrom = 1, vermelho = 2, etc.

Posição dos dígitos: O primeiro e o segundo dígitos correspondem aos valores das duas primeiras faixas. O terceiro dígito indica o multiplicador, ou seja, o número de zeros que devem ser adicionados ao valor dos dois primeiros dígitos. Por exemplo, se a terceira faixa for vermelha, isso significa multiplicar o valor encontrado por 100.

Interpretação da tolerância: Se houver uma quarta faixa, ela indica a tolerância do resistor, ou seja, a variação permitida no valor nominal. Por exemplo, uma faixa dourada indica uma tolerância de $\pm 5\%$, enquanto uma prata indica uma tolerância de $\pm 10\%$.

Cálculo do valor da resistência: Com base nos valores associados a cada cor e na posição das faixas, é possível calcular o valor da resistência. Basta combinar os valores dos dígitos das três primeiras faixas e multiplicá-los pelo multiplicador indicado pela terceira faixa.

Por exemplo: se as cores das faixas de um resistor são marrom, preto, vermelho e ouro, isso representa um resistor com os valores de resistência de $1,0 \times 10$ ohms, ou 100 ohms, com uma tolerância de $\pm 5\%$.



VOLTAR AO
SUMÁRIO



Tabela de cores de resistores

4 - Faixas

Cor	1ª Faixa	2ª Faixa	3ª Faixa	Multiplicador	Decimal	Tolerância
Preto	0	0	0	1	1	
Marron	1	1	1	10	10	± 1%
Vermelho	2	2	2	100	100	± 2%
Laranja	3	3	3	1K	1.000	
Amarelo	4	4	4	10K	10.000	
Verde	5	5	5	100K	100.000	
Azul	6	6	6	1M	1.000.000	
Violeta	7	7	7	10M	10.000.000	
Cinza	8	8	8	100.000.000		
Branco	9	9	9	1.000.000.000		
Duro				0.1		± 5%
Prata				0.01		± 10%
Branco						± 20%

5 - Faixas

Exemplo de leitura

Cálculos

Pino do resistor: Esquerdo, Direito

1ª Faixa Vermelho.....2
 2ª Faixa Preto.....0
 Multiplicador Amarelo...x10000
 Tolerância Prata....10%

A faixa de tolerância é sempre à direita do resistor. O valor do resistor é lido da esquerda para a direita.

Leitura
 20 x 10000 = 200000 ± 10% de tolerância

Calculadora de resistência

Existem diversos aplicativos calculadora de resistência, consiste em uma ferramenta digital que permite aos usuários calcular o valor da resistência de um resistor a partir das cores de suas faixas. Geralmente, esses aplicativos possuem uma interface simples onde o usuário seleciona as cores das faixas do resistor e o aplicativo fornece o valor da resistência correspondente, seguindo o código de cores padrão. Essas calculadoras são úteis para técnicos, engenheiros e hobistas que precisam identificar rapidamente o valor de resistores em circuitos eletrônicos.

Tenha no seu celular!
 Baixe o aplicativo

Disponível para:



LED

Os LEDs (Light Emitting Diodes) são dispositivos semicondutores que convertem energia elétrica em luz. Compostos por materiais semicondutores específicos, os LEDs emitem luz quando elétrons se recombinam com lacunas na estrutura do material. Sua eficiência energética, longa vida útil e tamanho compacto os tornam extremamente versáteis em uma variedade de aplicações.



VOLTAR AO SUMÁRIO



O que é um LED?

LED

Esses dispositivos são amplamente utilizados na iluminação devido à sua capacidade de fornecer luz brilhante e de baixo consumo de energia. Além disso, são comuns em displays de eletrônicos, como telas de TV, monitores de computador e painéis de instrumentos. Nos setores automotivo e de sinalização, são empregados em faróis, lanternas e semáforos devido à sua rápida resposta e durabilidade. Também encontram aplicação em dispositivos de comunicação óptica, em aparelhos médicos para terapia de luz e até em dispositivos de cultivo indoor, onde podem fornecer o espectro de luz necessário para o crescimento das plantas.



Cuidado ao ligar o LED.

O LED, assim como alguns componentes, possui um lado positivo “+” e um lado negativo “-”, essa característica também é conhecida como polaridade. Para esse componente, se ligar de modo invertido, ele não irá funcionar e pode até acabar danificando. A haste (perna) maior do LED é o lado positivo e a menor é o lado negativo. Você pode ver também pelo lado mais achatado, que é o lado negativo enquanto o lado arredondado é o positivo.

Disponível em:

<https://www.makehero.com/blog/aprenda-a-piscar-um-led-com-arduino/>

Atenção!

Segundo o fabricante a maioria dos LEDs da plataforma arduino, funcionam com tensão de aproximadamente 2V, assim se você ligar ele direto a placa do Arduino, ele poderá queimar.

Geralmente ligamos em série com o LED, um resistor de aproximadamente 200 Ohms, com isso o resistor vai diminuir a tensão sobre o LED.

Observação: Se você não tiver um resistor de 200 Ohms para ligar com o LED, utilize um resistor entre 100 Ohms e 250 Ohms, já vai ajudar bastante o LED.



VOLTAR AO SUMÁRIO



Sensor de Temperatura



Sensor de Temperatura

O sensor de temperatura é um dispositivo que mede a temperatura ambiente e fornece um valor de saída que pode ser lido e processado pelo Arduino. Geralmente, os sensores de temperatura mais comuns, como o LM35 ou o DHT11/DHT22, convertem a temperatura em uma tensão ou um sinal digital, permitindo ao Arduino tomar decisões ou exibir informações baseadas na temperatura ambiente detectada.

Como ele funciona?

O sensor LM35 é um sensor de temperatura analógico. Funciona convertendo a temperatura ambiente em uma voltagem proporcional à temperatura. Cada grau Celsius de mudança na temperatura resulta em uma mudança de 10 mV na saída do sensor. Assim, o Arduino pode ler essa voltagem e convertê-la em uma leitura de temperatura precisa usando sua entrada analógica.



Aplicações do sensor de Temperatura

- Estufas agrícolas, nos quais o sensor de temperatura pode ativar ou desativar sistemas de ventilação ou aquecimento para manter as condições ideais de crescimento das plantas.
- Salas de servidores, nas quais o sensor pode controlar a operação de ventiladores de refrigeração para evitar o superaquecimento dos equipamentos.
- Residências, para automatizar sistemas de ar condicionado ou aquecimento, proporcionando um ambiente mais confortável e economizando energia elétrica.
- Essa solução oferece uma maneira eficaz e conveniente de controlar dispositivos com base nas condições de temperatura, melhorando a eficiência operacional e a comodidade do ambiente.
- Ao você rodar o programa, o sensor de temperatura LM35 mede a temperatura ambiente onde está localizado, e o monitor irá mostrar os valores de temperatura a cada 1 segundo. Note que ele irá medir a temperatura ambiente.
- Um outro teste que você pode fazer é aproximar objetos com temperaturas diferentes perto do sensor, note no monitor que os valores vão ser alterados.



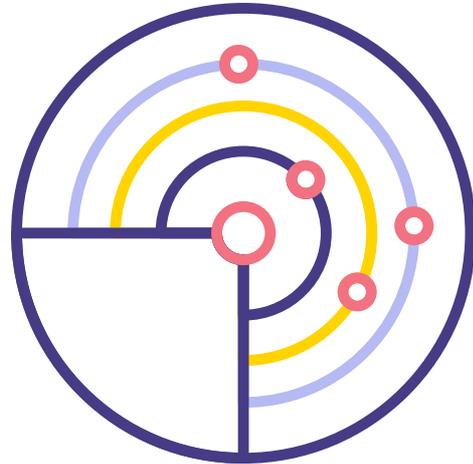
VOLTAR AO
SUMÁRIO



Sensor Ultrassônico

SENSOR ULTRASSÔNICO

O sensor ultrassônico HC-SR04 é um dispositivo popular utilizado com a plataforma Arduino para medir distâncias. Ele emite pulsos ultrassônicos que atingem um obstáculo e retorna ao receptor, com isso ele consegue detectar um obstáculo. Alguns animais utilizam a mesma técnica para se localizar no espaço, como morcegos e Golfinhos. Esse sensor pode realizar diversas aplicações, como controle de robôs, detecção de obstáculos e medição de níveis de líquidos, etc.

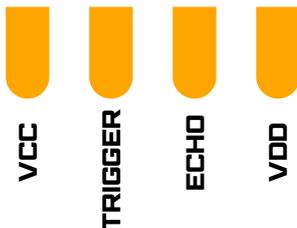


Como ele funciona?

O Sensor Ultrassônico HC-SR04 emite pulsos de ultrassom que são refletidos de volta por um objeto. Com base no tempo de ida e volta do pulso, o sensor calcula a distância até o objeto usando a velocidade do som como referência. Essa informação é então transmitida para o Arduino, que pode ser utilizado para diversas aplicações, como medição de distância ou detecção de obstáculos.

Emissor

Receptor



VCC

TRIGGER

ECHO

VDD

Algumas aplicações do Sensor Ultrassônico

- **Medição de distância:** O sensor pode ser usado para medir com precisão a distância entre ele e um objeto, útil em projetos de Robótica, controle de altura, ou qualquer aplicação que exija medição de distância.
- **Detecção de obstáculos:** Ele pode ser empregado em robôs, veículos autônomos ou sistemas de segurança para detectar a presença de obstáculos em seu caminho e evitar colisões.
- **Contagem de pessoas:** O sensor pode ser instalado em entradas ou saídas de espaços públicos para contar o número de pessoas que entram ou saem, sendo útil em controle de acesso ou gerenciamento de fluxo de pessoas.
- **Medição de nível de líquidos:** Quando posicionado acima de um líquido, o sensor pode determinar o nível do líquido com base na distância entre o sensor e a superfície do líquido, sendo útil em tanques de água, reservatórios ou sistemas de irrigação automatizados.
- **Monitoramento de vagas de estacionamento:** O sensor pode ser usado para detectar a presença de veículos em vagas de estacionamento, permitindo a criação de sistemas inteligentes de gerenciamento de estacionamento que indicam a disponibilidade de vagas em tempo real.



VOLTAR AO SUMÁRIO



Sensor LDR

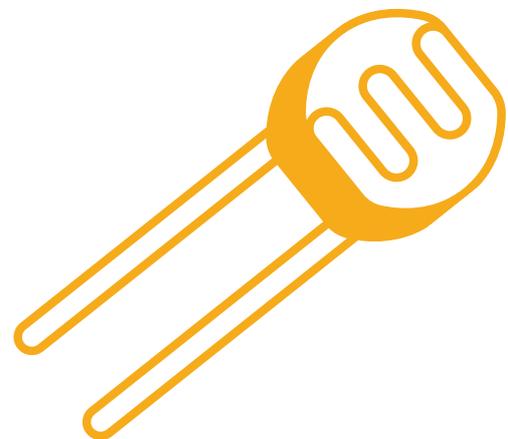


Sensor LDR

O sensor LDR (Light Dependent Resistor), ou resistor dependente de luz, é um componente usado para medir a intensidade de luz. Sua resistência varia conforme a quantidade de luz incidente sobre ele. Quando mais luz incide sobre o sensor, menor é sua resistência, e vice-versa. É comumente usado em projetos com Arduino para detecção de luz ambiente, controle de iluminação automático e outras aplicações relacionadas à luminosidade.

Como ele funciona?

O sensor LDR (Light Dependent Resistor) funciona com base em seu material semicondutor especial, que altera sua resistência elétrica em resposta à quantidade de luz incidente sobre ele. Quando exposto à luz, os elétrons no material do sensor ganham energia e migram para uma banda de condução, diminuindo a resistência do dispositivo. Quanto mais intensa a luz, menor será a resistência do LDR, e vice-versa. Isso permite que o sensor LDR seja usado para detectar variações na luminosidade e controlar dispositivos eletrônicos de acordo com as condições de iluminação ambiental.



Aplicações do sensor de Temperatura

- **Controle de iluminação automático:** O sensor LDR pode ser usado para ajustar automaticamente a intensidade da iluminação em ambientes internos ou externos com base na luz ambiente. Pode-se acender as luzes em um ambiente quando a luminosidade diminui e desligá-las quando a luz natural é suficiente.
- **Detecção de luz em sistemas de segurança:** O sensor LDR pode ser incorporado em sistemas de segurança para detectar variações na iluminação ambiente, acionando alarmes ou câmeras de vigilância em condições de baixa luminosidade, por exemplo.
- **Automação residencial:** Em sistemas de automação residencial, o sensor LDR pode ser utilizado para controlar persianas, cortinas ou janelas automáticas, permitindo ajustar a entrada de luz natural em uma residência de acordo com as preferências do usuário ou condições climáticas.
- **Monitoramento ambiental:** O sensor LDR pode ser usado em projetos de monitoramento ambiental para medir a luminosidade em diferentes áreas, auxiliando na análise de padrões de luz ao longo do dia ou identificando áreas com baixa iluminação.
- **Projetos de arte interativa:** O sensor LDR pode ser incorporado em instalações de arte interativa para detectar a presença de espectadores ou interagir com as condições de iluminação ao redor, criando experiências visuais dinâmicas baseadas na luz ambiente.



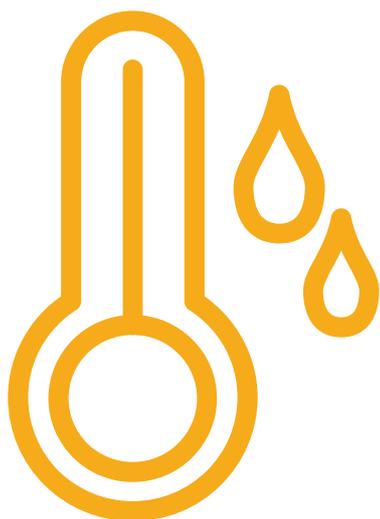
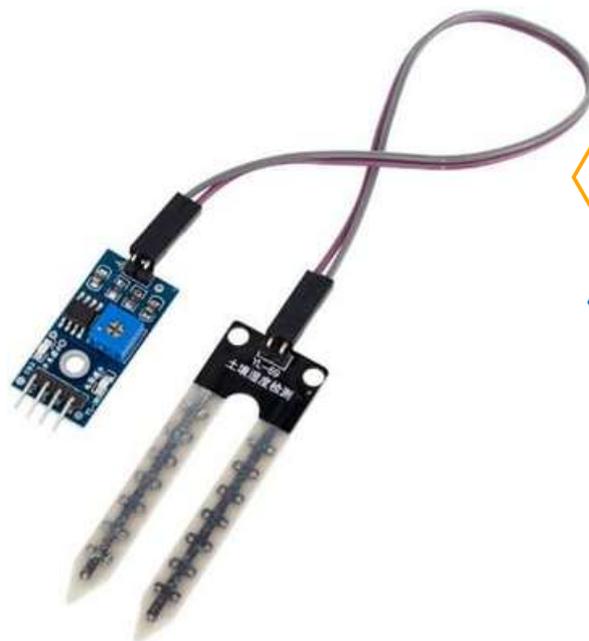
VOLTAR AO
SUMÁRIO



Sensor de Umidade

SENSOR DE UMIDADE

O sensor de umidade é um componente que detecta a quantidade de umidade presente em seu ambiente. Ele geralmente é usado para medir a umidade relativa do ar ou a umidade do solo. O sensor converte a umidade detectada em um sinal elétrico ou digital, que pode ser lido pelo Arduino. Isso permite que o Arduino monitore as condições de umidade e tome ações com base nessas informações, como controlar sistemas de irrigação, ativar dispositivos de ventilação ou alertar sobre condições de umidade indesejadas. Existem diferentes tipos de sensores de umidade disponíveis para o Arduino, incluindo sensores capacitivos e resistivos, cada um com suas próprias características e métodos de operação.



Como ele funciona?

Ele funciona por meio de princípios físicos, como capacitância ou resistência. Por exemplo, sensores capacitivos usam um material dielétrico entre placas condutoras, cuja capacitância é afetada pela umidade. Já os sensores resistivos alteram sua resistência de acordo com a umidade absorvida pelo material higroscópico. O Arduino interpreta a saída desses sensores e pode controlar sistemas, como irrigação ou umidificação, com base nas condições de umidade detectadas.

Algumas aplicações do Sensor Ultrassônico

- **Irrigação Inteligente:** Monitoramento e controle da umidade do solo para otimizar a irrigação agrícola, economizando água e garantindo o crescimento saudável das plantas.
- **Controle de Umidade em Ambientes Internos:** Regulação automática da umidade do ar em espaços fechados para evitar problemas de mofo, condensação e garantir o conforto dos ocupantes.
- **Gestão de Ambientes de Cultivo:** Manutenção das condições ideais de crescimento em estufas e viveiros através do monitoramento da umidade do solo e do ar.
- **Monitoramento Meteorológico:** Coleta de dados sobre a umidade do ar para previsão do tempo precisa e análise climática em estações meteorológicas e projetos de IoT.
- **Controle de Qualidade em Processos Industriais:** Garantia da qualidade e segurança de produtos em processos industriais sensíveis à umidade, como na fabricação de alimentos, produtos farmacêuticos e eletrônicos.



VOLTAR AO
SUMÁRIO



Dispositivo Buzzer



Dispositivo Buzzer

O buzzer é um dispositivo piezoelétrico que opera com base na conversão de energia elétrica em vibrações mecânicas, as quais produzem som. Embora tanto o buzzer quanto o alto-falante sejam dispositivos que produzem som, eles têm diferenças significativas. Ele produz um som mais simples e direto, muitas vezes na forma de um tom único ou de um padrão de alerta, como um bip ou um zumbido, emitindo uma nota de determinada Frequência.

Como ele funciona?

O buzzer opera com base na conversão de energia elétrica em vibrações mecânicas, as quais produzem som. Quando uma corrente elétrica é aplicada ao buzzer, um campo elétrico é gerado dentro do cristal piezoelétrico. Esse campo faz com que o cristal se expanda e contraia rapidamente, gerando vibrações mecânicas. Essas vibrações se propagam através do ar como ondas sonoras, criando o som audível que ouvimos. Dependendo da frequência e intensidade da corrente elétrica aplicada, é possível controlar o tom e o volume do som emitido pelo buzzer.



Aplicações do dispositivo Buzzer

- **Alarmes de Segurança:** Utilizado em sistemas de segurança residencial, comercial e automotiva para emitir alertas sonoros em caso de intrusão, detecção de fumaça, ou outras situações de emergência.
- **Sinalização em Dispositivos Eletrônicos:** Presente em equipamentos eletrônicos, como computadores, eletrodomésticos e dispositivos móveis, para indicar eventos como notificações de mensagens, alertas de bateria fraca ou finalização de processos.
- **Controle de Acessos:** Empregado em sistemas de controle de acesso para indicar autorização ou recusa de entrada, como em portas com fechaduras eletrônicas ou catracas.
- **Feedback Sonoro em Jogos e Brinquedos:** Incorporado em jogos eletrônicos, brinquedos e dispositivos de entretenimento para fornecer feedback sonoro durante interações do usuário, como pontos marcados, sucessos ou falhas.
- **Sinalização de Erros em Equipamentos:** Utilizado em equipamentos industriais, máquinas e dispositivos eletrônicos para alertar sobre falhas ou condições de erro, facilitando a identificação e correção de problemas.



VOLTAR AO
SUMÁRIO



Músicas na plataforma Arduino

MÚSICAS NA PLATAFORMA

Na plataforma Arduino, é possível produzir música utilizando um buzzer piezoelétrico ou um módulo de som como o DFPlayer Mini. Com o buzzer, podemos gerar diferentes frequências sonoras aplicando correntes elétricas em intervalos específicos, criando melodias simples através da programação. Já com o DFPlayer Mini, podemos reproduzir arquivos de áudio pré-gravados armazenados dentro da placa, controlando a reprodução por meio de comandos enviados ao módulo. Ambos os métodos permitem a criação de projetos musicais interessantes, desde reproduções de melodias simples até composições mais elaboradas.



Disponível em: <https://jrmcoaching.com.br/blog/20-musicas-superacao-vaio-inspirar/>



Como ele funciona?

No caso do buzzer piezoelétrico, o Arduino envia sinais elétricos para o buzzer em diferentes frequências, usando a função `define NOTE_xxxx`, para gerar notas musicais. A frequência determina o tom da nota, enquanto a duração do sinal define a sua duração. Isso permite criar sequências de notas que compõem uma melodia. Já com o DFPlayer Mini, o Arduino envia comandos serial para controlar a reprodução de arquivos de áudio armazenados em um cartão SD conectado ao módulo, possibilitando a reprodução de músicas pré-gravadas.

Algumas sugestões de programas completos de músicas

Super Mario bros



<https://github.com/robsoncouto/arduino-songs/blob/master/supermariobros/supermariobros.ino>



Game Of Thrones

<https://github.com/robsoncouto/arduino-songs/blob/master/supermariobros/supermariobros.ino>



VOLTAR AO SUMÁRIO



Software de Programação

Acender_um_Led | Arduino 1.8.19 (Windows Store 1.8.57.0)

Arquivo Editar Sketch Ferramentas Ajuda

```
Acender_um_Led$
1 .....
2 *
3 *   Projeto 02 - Ligar Led enquanto botão de pressão ficar press
4 *   SBCU-ES
5 *   GCCEB
6 *
7 .....
8
9 // Projeto 02 - Push button liga led quando pressionado (pull do
10 const int ledPin = 2; //led no pino 2
11 const int Botao = 12; //botao no pino 12
12 int estadoBotao; //Variavel para ler o status do pushbutton
13 void setup() {
14     pinMode(ledPin, OUTPUT); //Pino do led será saída
15     pinMode(Botao, INPUT); //Pino com botão será entrada
16 }
17 void loop() {
18     estadoBotao = digitalRead(Botao); //le o estado do botão - HI
19     if (estadoBotao == HIGH) { //se botão estiver pressionado (HI)
20         digitalWrite(ledPin, HIGH); // acende o led do pino 2.
21     }
22     else { //se não estiver pressionado (LOW)
23         digitalWrite(ledPin, LOW); //deixa o led do pino 2 apagar
24     }
25 }
26
```

Conheceremos o software de programação Arduino IDE, a interface que nos permite escrever e carregar programas no Arduino. Exploraremos a estrutura básica de um programa e como traduzi-lo para instruções compreensíveis pelo microcontrolador.

Para consolidar o aprendizado, realizaremos um exemplo prático conectando componentes ao Arduino. Isso não apenas reforçará os conceitos aprendidos, mas também dará vida ao potencial do Arduino.

A apresentação destacando projetos inspiradores desenvolvidos com Arduino em todo o mundo. Desde simples experimentos até inovações tecnológicas, esses exemplos ilustram a amplitude e a aplicabilidade do Arduino.

Ao desvendar os componentes e funcionalidades do Arduino, estamos dando o primeiro passo para explorar as vastas possibilidades da eletrônica criativa. Preparem-se para colocar as mãos na massa e transformar conceitos em realidade durante esta empolgante jornada com o Arduino!

Instalando o programa, IDE.



Arduino IDE 2.3.2

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits

Windows MSI installer

Windows ZIP file

Linux AppImage 64 bits (X86-64)

Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.15: "Catalina" or newer, 64 bits

macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

Atenção!

- Na hora de fazer o Download, escolha o programa adequado ao sistema operacional, correspondente ao seu PC.
- A IDE sempre sofre atualizações, é importante manter sua IDE atualizada.
- **IDE** é o ambiente de desenvolvimento utilizado para programar todas as placas da categoria Arduino. Vamos começar pela decomposição dessas palavras. IDE é uma abreviação do termo em inglês, **Integrated Development Environment**, que em português significa "Ambiente de Desenvolvimento Integrado".

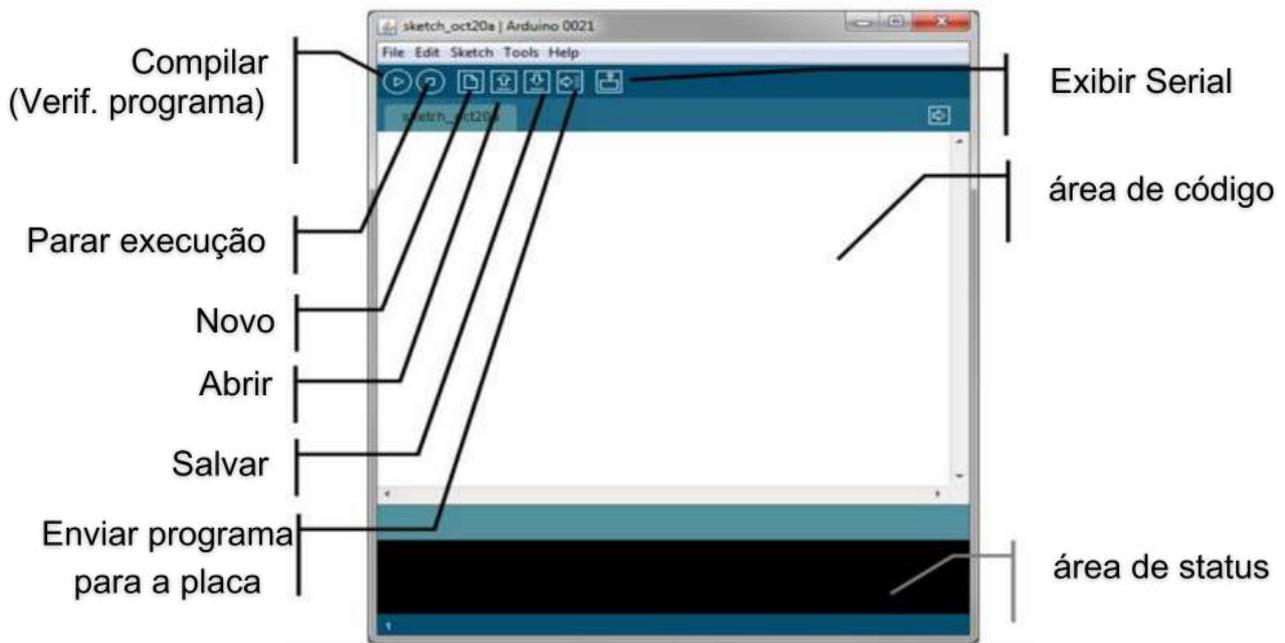


VOLTAR AO SUMÁRIO



Software de Programação:

O ambiente de desenvolvimento da plataforma Arduino oferece uma interface simplificada e acessível para programação de dispositivos eletrônicos. Com base em uma linguagem de programação C/C++ e suportado por uma ampla gama de bibliotecas de código aberto, o ambiente Arduino é projetado para facilitar o desenvolvimento de projetos, mesmo para iniciantes. Abaixo podemos observar, como é uma interface gráfica onde é exibida várias funções, com o tempo você irá se acostumar com o significado de cada parte da IDE, na figura abaixo trazemos algumas de suas principais funções, as figuras estão disponíveis na Revista Tecnológica da FATEC-PR, pg 200.



-  Verifica se o código tem erros.
-  Compila o código e carrega para a placa do Arduino.
-  Cria um novo sketch.
-  Apresenta um menu de todos os sketches existentes.
-  Salva o sketch.
-  Abre o monitor serial.

o que é sketch?

Um "sketch" no contexto do Arduino é simplesmente o código-fonte de um programa escrito em C/C++, que controla o funcionamento de um dispositivo eletrônico conectado à placa Arduino. É o conjunto de instruções que dizem ao microcontrolador o que fazer: ler dados de sensores, processar informações e controlar dispositivos de saída. O sketch é criado e editado usando o ambiente de desenvolvimento do Arduino, onde os usuários escrevem, compilam e carregam seus programas na placa Arduino. Ou seja, o sketch é a essência do código por trás de um projeto Arduino.



COMPETÊNCIAS GERAIS DA BNCC

A Base Nacional Comum Curricular (BNCC) é um documento fundamental para a educação brasileira, que estabelece diretrizes para a construção de currículos escolares em todo o país. Dentro desse contexto, as competências da BNCC ocupam um lugar central, delineando as habilidades essenciais que os estudantes devem desenvolver ao longo de sua trajetória educacional.

Destacamos a importância dessas competências e seu papel na formação de cidadãos preparados para os desafios do século XXI.

As competências da BNCC são agrupadas em cinco áreas: Competências Gerais, Competências Específicas da Educação Infantil, Competências Específicas do Ensino Fundamental I, Competências Específicas do Ensino Fundamental II, e Competências Específicas do Ensino Médio. Cada uma dessas áreas enfatiza aspectos diferentes do desenvolvimento integral dos estudantes, abrangendo desde habilidades cognitivas até socioemocionais.

São dez as Competências Gerais da BNCC, elas visam a formação de indivíduos autônomos, críticos e criativos, capazes de compreender e transformar a realidade à sua volta. Essas competências incluem a valorização da diversidade, o pensamento crítico, a comunicação eficaz, a resolução de problemas e a tomada de decisões éticas, entre outros aspectos. Ao priorizar tais habilidades, a BNCC reconhece a importância de uma educação que vai além da mera transmissão de conteúdos, buscando preparar os alunos para a vida em sociedade.

Competências Específicas de cada etapa de ensino complementam essa visão, adaptando-se às necessidades e características dos estudantes em diferentes fases do desenvolvimento. Na Educação Infantil, por exemplo, as competências estão voltadas para o estímulo à curiosidade, à experimentação e à expressão das crianças, enquanto no Ensino Médio o foco se volta para a formação cidadã, a preparação para o mundo do trabalho e o acesso ao conhecimento científico e tecnológico.

É importante ressaltar que as competências da BNCC não se limitam ao âmbito acadêmico, mas abrangem também aspectos socioemocionais fundamentais para o desenvolvimento humano. A promoção da empatia, da cooperação, da resiliência e da responsabilidade social são elementos essenciais presentes em diversas competências, refletindo a compreensão de que a educação deve formar não apenas indivíduos competentes, mas também éticos e solidários.

Portanto, as competências da BNCC representam um avanço significativo na concepção da educação brasileira, ao estabelecer diretrizes claras para a formação de cidadãos preparados para os desafios do século XXI. Ao promover uma visão ampla e integrada do processo educacional, que valoriza tanto o desenvolvimento cognitivo quanto o socioemocional, a BNCC contribui para a construção de uma sociedade mais justa, democrática e inclusiva. Portanto, é fundamental que gestores, educadores e toda a comunidade escolar se engajem na implementação efetiva dessas competências, garantindo que todos os alunos tenham acesso a uma educação de qualidade e relevo para suas vidas.

COMPETÊNCIAS GERAIS



BNCC PROPÓSITO

Contribuir para a construção de uma sociedade mais ética, democrática, responsável, sustentável e solidária, que respeite e promova a diversidade e os direitos humanos, sem preconceitos de qualquer natureza.



Crédito: Lauri Cericato
Fonte: BNCC
Arte: Sergio Cândido

Figura disponível em: <https://escoladeinventor.com.br/espaco-maker-como-trabalhar-a-bncc/>



VOLTAR AD
SUMÁRIO



Robótica, Programação e Pensamento computacional

Competência(s) Gerais da BNCC abordada(s) no caderno:

✓ Conhecimento

Valorizar e utilizar os conhecimentos historicamente construídos sobre o mundo físico, social, cultural e digital para entender e explicar a realidade, continuar aprendendo e colaborar para a construção de uma sociedade justa, democrática e inclusiva.

✓ Comunicação

Utilizar diferentes Linguagens, nesse contexto os estudantes podem, criar e simular algoritmos representados em linguagem oral, escrita ou pictográfica, que incluam sequências, repetições e seleções condicionais para resolver problemas de forma independente e em colaboração.

✓ Pensamento Científico, Crítico e Criativo

Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas) com base nos conhecimentos das diferentes áreas.



✔ Cultura digital

Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.

✔ Responsabilidade e cidadania

Agir pessoal e coletivamente com autonomia, responsabilidade, flexibilidade, resiliência e determinação, tomando decisões com base em princípios éticos, democráticos, inclusivos, sustentáveis e solidários.

✔ Empatia e Cooperação

Exercitar a empatia, o diálogo, a resolução de conflitos e a cooperação.

✔ Argumentação

Argumentar com base em fatos, dados e informações confiáveis, para formular, negociar e defender ideias, pontos de vista e decisões comuns que respeitem e promovam os direitos humanos, a consciência socioambiental e o consumo responsável em âmbito local, regional e global, com posicionamento ético em relação ao cuidado de si mesmo, dos outros e do planeta.



Competências em Computação da BNCC:

No contexto da sociedade contemporânea, marcada pela crescente digitalização e interconexão global, o Pensamento Computacional emerge como uma habilidade essencial para a compreensão e atuação eficaz no mundo digital. Associado à cultura digital, ele se torna uma ferramenta poderosa para a resolução de problemas complexos e a promoção da inovação. Sendo assim, a BNCC, disponibilizou um documento complementar, ligado a computação no qual, iremos explorar o papel do pensamento computacional no contexto do mundo digital e da cultura digital, destacando seus desafios e oportunidades.

O Pensamento Computacional pode ser definido como uma forma de raciocínio analítico e abstrato, que envolve a decomposição de problemas em partes menores, a identificação de padrões e a criação de algoritmos para sua resolução. Essa habilidade não se limita apenas aos especialistas em computação, mas é cada vez mais relevante em diversas áreas do conhecimento e da atuação profissional.

Além disso, o Pensamento Computacional está intrinsecamente ligado à cultura digital, que engloba as práticas, os valores e as representações sociais associadas ao universo digital. A cultura digital influencia como nos comunicamos, consumimos informação, nos relacionamos e construímos conhecimento. Nesse sentido, o pensamento computacional não apenas nos capacita a compreender melhor a cultura digital, mas também nos permite participar ativamente de sua construção, contribuindo para uma sociedade mais inclusiva e participativa.

Enquanto as máquinas estão cada vez mais comuns em nosso cotidiano, o Brasil está caminhando a passos curtos, para o desenvolvimento dessas competências como a argumentação, a cultura digital e o pensamento científico, crítico e criativo. É preciso fazer nossos discentes se apropriarem desses conhecimentos. É nesse cenário, que a Gerência de Currículo do Estado do Espírito Santo está elaborando esse material para compor o Currículo do nosso estado, concordamos com CAMPOS, quando ele diz que,

Uma proposta educacional adequada, o currículo e um ambiente de aprendizagem são alguns dos importantes elementos que podem direcionar a inovação no campo da robótica educacional. É necessário, portanto, mudar o foco da tecnologia para o desenvolvimento de currículo, pois acreditamos que o currículo seja elemento-chave na robótica educacional e é imprescindível incorporar os princípios da aprendizagem e determinar métricas qualitativas e quantitativas para os resultados esperados e para validação do currículo (Pg12).

Diante do exposto, o Pensamento Computacional representa um conjunto de habilidades fundamental para o enfrentamento dos desafios e a aproveitamento das oportunidades no mundo digital. Ao desenvolver essas habilidades, os indivíduos se tornam mais capacitados para compreender e atuar de forma criativa e ética em um ambiente cada vez mais permeado pela tecnologia. No entanto, é importante que o desenvolvimento do Pensamento Computacional seja acompanhado por uma reflexão crítica sobre seus impactos sociais, culturais e éticos, garantindo que a transformação digital ocorra de forma inclusiva e sustentável.



VOLTAR AO SUMÁRIO



De acordo com a RESOLUÇÃO Nº 1, DE 4 DE OUTUBRO DE 2022, foram definidas normas para a implementação de Computação na Educação Básica, em complemento à Base Nacional Comum Curricular (BNCC), o documento trás uma série de medidas que serão implementadas, dentre elas podemos destacar: O desenvolvimento e formulação dos currículos deve considerar as tabelas de competências e habilidades na área da computação, formação nacional para o desenvolvimento dos saberes docentes para o ensino de Computação na Educação Básica, dentre outras. A Secretaria Estadual do Espírito Santo, assumindo seu compromisso com uma educação de qualidade e atendida com as novas demandas que surgem na sociedade, vem por meio desse caderno dar um ponta pé inicial a implementação do ensino de róbótica, com atividades que podem ser aplicadas tanto no ensino fundamental, quanto no ensino médio.

No diagrama abaixo podemos observar os três eixos temáticos que serão explorados nesse caderno:



Disponível em: <https://liag.ft.unicamp.br/act/2022/10/18/aprovada-a-resolucao-com-normas-sobre-a-computacao-a-bncc/>



Para saber mais sobre o assunto, assista o vídeo no Youtube, a palestra foi realizada no dia 24 de abril de 2023.

Quer saber um pouco mais?

Palestra - BNCC Computação e Política Nacional de Educação Digital: caminhos para implantação

BR



YouTube



[VOLTAR AO SUMÁRIO](#)



Competências em Computação da BNCC:

No texto complementar a BNCC, chamado de **BNCC-Computação**, encontramos todas as competências gerais e específicas, assim como as habilidades desenvolvidas na área de Robótica. Os temas referentes ao uso dessas tecnologias empregadas no currículo, são diluídos desde a Educação Infantil até o Ensino Médio, que são: **Pensamento computacional, Cultura digital e Mundo digital. (BRASIL, 2018, p.61).**

Nosso caderno irá abordar as seguintes competências:

1

Compreender as possibilidades e os limites da Computação para resolver problemas, tanto em termos de viabilidade quanto de eficiência, propondo e analisando soluções computacionais para diversos domínios do conhecimento, considerando diferentes aspectos.

2

Analisar criticamente artefatos computacionais, sendo capaz de identificar as vulnerabilidades dos ambientes e das soluções computacionais buscando garantir a integridade, privacidade, sigilo e segurança das informações.

3

Analisar situações do mundo contemporâneo, selecionando técnicas computacionais apropriadas para a solução de problemas.

4

Construir conhecimento usando técnicas e tecnologias computacionais, produzindo conteúdos e artefatos de forma criativa, com respeito às questões éticas e legais, que proporcionem experiências para si e os demais.

5

Desenvolver projetos para investigar desafios do mundo contemporâneo, construir soluções e tomar decisões éticas, democráticas e socialmente responsáveis, articulando conceitos, procedimentos e linguagens próprias da Computação preferencialmente de maneira colaborativa.

6

Expressar e compartilhar informações, ideias, sentimentos e soluções computacionais utilizando diferentes plataformas, ferramentas, linguagens e tecnologias da Computação de forma fluente, criativa, crítica, significativa, reflexiva e ética.

7

Agir pessoal e coletivamente com respeito, autonomia, responsabilidade, flexibilidade, resiliência e determinação, identificando e reconhecendo seus direitos e deveres, recorrendo aos conhecimentos da Computação e suas tecnologias frente às questões de diferentes naturezas.



VOLTAR AO SUMÁRIO

Temas integradores e a Robótica:

Os temas integradores presentes na BNCC, são fundamentais para uma educação contextualizada e significativa. Esses temas conectam diferentes áreas do conhecimento, promovendo uma abordagem interdisciplinar que transcende as fronteiras tradicionais das disciplinas. Ao abordar questões complexas e relevantes para a sociedade contemporânea, como sustentabilidade e diversidade cultural, os temas integradores fornecem uma compreensão ampla do mundo e estimulam o pensamento crítico nos alunos. Além disso, promovem valores fundamentais, como ética e justiça, contribuindo para a formação de cidadãos conscientes e engajados. Assim, os temas integradores desempenham um papel crucial na preparação dos alunos, para os desafios exigidos por uma sociedade moderna, capacitando-os a compreender e enfrentar os problemas de forma reflexiva, íntegra e responsável. Destacamos três temas Integradores que estarão presentes nesse caderno, Educação Ambiental, Trabalho, Ciência e Tecnologia, e Educação para o trânsito.

(TI03) – Educação ambiental

A Educação Ambiental é um dos temas integradores essenciais na Base Nacional Comum Curricular (BNCC), abrangendo aspectos interdisciplinares que visam promover uma consciência ambiental crítica e sustentável nos estudantes. Por meio dela, os alunos são incentivados a compreender a complexidade dos problemas ambientais contemporâneos e a buscar soluções inovadoras e colaborativas para enfrentá-los. A Educação Ambiental na BNCC não se limita a disciplinas específicas, mas permeia todo o currículo, promovendo a interação entre diferentes áreas do conhecimento em prol da compreensão integral do ambiente. A compreensão da plataforma Arduino pode vir a contribuir com o meio ambiente, com alguns sensores é possível monitorar a quantidade de água, luminosidade e temperatura das plantas, plantações etc, facilitando o plantio e cultivo. Além disso, ela estimula a conexão dos estudantes com a natureza e a valorização da biodiversidade, contribuindo para a formação de cidadãos conscientes e responsáveis em relação ao seu papel na preservação do planeta.

(TI12) – Trabalho, Ciência e Tecnologia.



Trabalho, Ciência e Tecnologia, reflete a interconexão entre esses três pilares fundamentais para o desenvolvimento socioeconômico e cultural de uma nação. Através desse tema, os estudantes são incentivados a compreender a relação entre o avanço científico, a evolução tecnológica e as transformações no mercado de trabalho. Além disso, a BNCC busca promover uma visão crítica e reflexiva sobre o papel da ciência e da tecnologia na sociedade contemporânea, estimulando os alunos a se tornarem agentes ativos na construção de um futuro sustentável e inovador. O tema integrador também visa preparar os estudantes para os desafios e oportunidades do mundo do trabalho, desenvolvendo habilidades como pensamento crítico, criatividade, colaboração e adaptação às mudanças tecnológicas. Assim, Trabalho, Ciência e Tecnologia desempenham um papel crucial na formação de cidadãos capacitados e conscientes do seu papel na sociedade atual.

TI02. Educação para o Trânsito



Como cidadãos, os estudantes devem assumir diversos papéis, entre eles, o de pedestres, passageiros e até condutores de veículos. Assim, o Parecer CNE/CEB Nº 22/2004 (BRASIL, 2004) solicita a inclusão da Educação para o Trânsito no currículo das escolas e o apresentamos como tema transversal, em todos os níveis de ensino. A educação no trânsito não compreende apenas ensinar regras de circulação, mas, também, formar cidadãos participativos, responsáveis, autônomos e envolvidos com a preservação da vida.

Projeto 01:

Piscar um LED com Arduino

Etapa da Educação Básica:

Ensino Fundamental anos finais e Ensino médio

Público Alvo:

Alunos no 9º Ano, 1ª, 2ª ou 3ª Séries.

Modalidade:

() Sequência Didática. () Projeto () Iniciação Científica.

BNCC da Computação:

[\[EF69CO02\]](#) Elaborar algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.

[\[EF05CO07\]](#) Reconhecer a necessidade de um sistema operacional para a execução de programas e gerenciamento do hardware.

[\[EM13CO13\]](#) Analisar e utilizar as diferentes formas de representação e consulta a dados em formato digital para pesquisas científicas.

[\[EM13CO15\]](#) Analisar a interação entre usuários e artefatos computacionais, abordando aspectos da experiência do usuário e promovendo reflexão sobre a qualidade do uso dos artefatos nas esferas do trabalho, do lazer e do estudo.

[\[EM13CO16\]](#) Desenvolver projetos com robótica, utilizando artefatos físicos ou simuladores.

[\[EM13CO18\]](#) Planejar e gerenciar projetos integrados às áreas de conhecimento de forma colaborativa, solucionando problemas, usando diversos artefatos computacionais.

Objeto(s) de conhecimento:

- Conhecer a plataforma Arduino.
- Ligar um LED.



VOLTAR AD
SUMÁRIO



Vamos aos projetos!

Piscar um LED com Arduino



Objetivo da Aula:



Introduzir os alunos ao ambiente prático de programação e montagem de circuitos com Arduino, utilizando como exemplo a atividade básica de ligar um LED.

Contextualização



O projeto de piscar um LED com Arduino, é uma introdução prática e educativa ao mundo da programação e eletrônica. Ele oferece uma oportunidade simples e acessível para iniciantes aprenderem a ler ou a escrever o código, controlar dispositivos eletrônicos e entender os princípios básicos de automação. Além disso, ajuda a construir uma compreensão fundamental sobre como os microcontroladores podem ser usados para criar funcionalidades simples, mas eficazes.

Etapa



Orientações para o professor

Apresentando componentes da plataforma:

- ✓ Nessa primeira atividade você professor, deve apresentar aos estudantes a placa de prototipagem Arduino, seus pinos e componentes. Em especial, recomenda-se a leitura das páginas 13 a 15, deste material que tratam do Arduino Uno e seus componentes.
- ✓ Após a apresentação da placa de prototipagem e de seus componentes, o docente deve mostrar as possibilidades de entrada de alimentação para o Arduino:
- ✓ Você deve apresentar os elementos da plataforma, sensores, LED, protoboard, resistor.
- ✓ Ao apresentar o resistor, falar das aplicações, você professor deve ensinar como fazer a leitura através das listras, ver páginas 16 e 17.
- ✓ Ao ligar A IDE, para passar a programação do computador para a placa do Arduino, é recomendado ler as páginas 25, 26. Fique atento às informações na página 39.
- ✓ É importante que você esteja presente, sempre que solicitado pelo aluno, é o primeiro contato do estudante com a plataforma, muitas dúvidas irão surgir nesse momento.
- ✓ Separe o material da primeira atividade.
- ✓ E mãos a obra!



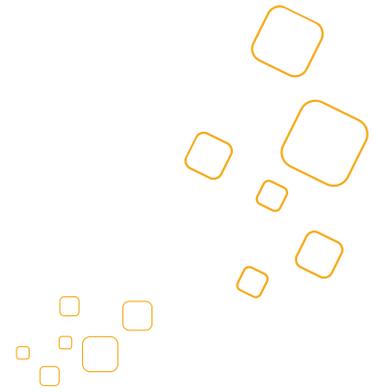
VOLTAR AO
SUMÁRIO



Etapa

2. Materiais Utilizados

- PC
- 1 x Cabo USB
- 1 x Placa Uno R3 com cabo USB
- 1 x Protoboard
- 1 x LED vermelho
- 1 x Resistor 220 Ohms
- 3x Fio Jumpers macho-macho
- 1 x Cabo USB para ligar a placa ao computador

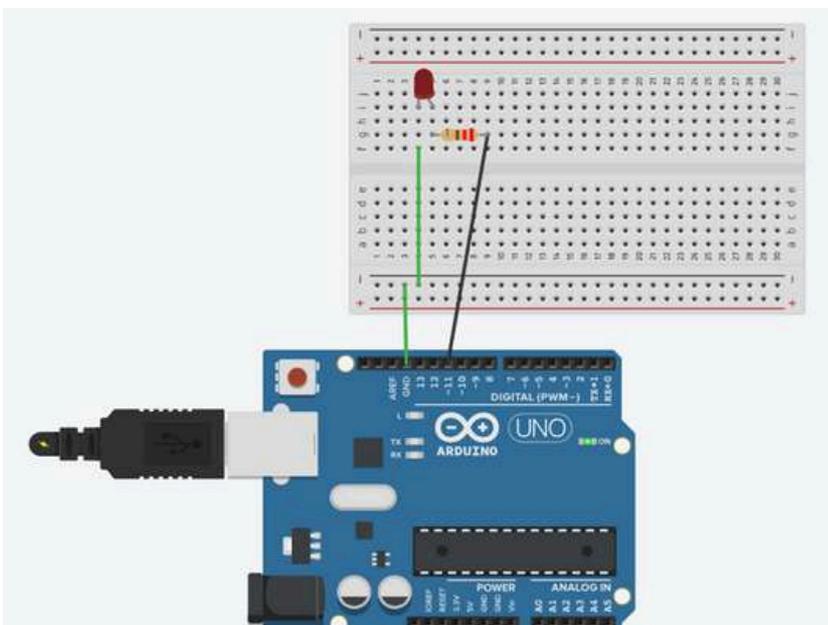


Etapa

3. Metodologia

3.1 Ligando Componentes:

- Separar os materiais a serem utilizados;
- Exibição dos componentes físicos: Arduino, protoboard, LED, resistência (se necessário) e fios jumpers;
- Explicar como se faz a leitura do resistor;
- Destaque para as conexões nos pinos de entrada/saída digitais do Arduino;
- Faça a ligação dos componentes conforme a figura a seguir;
- Tome cuidado ao ligar o LED, perna menor ligar no negativo = GND;



**AUTODESK
Tinkercad**

Quer montar esse projeto virtualmente, ele está disponível em:

<https://www.tinkercad.com/things/hRYNcp0VTjw-piscar-um-led-com-arduino>

Ou aponte o celular para o QR-code



[VOLTAR AO SUMÁRIO](#)



3.2 Programação:

- Agora é ligar o PC e instalar a IDE.
- Conecte a placa do Arduino ao PC, utilize o cabo USB.
- Esse momento é muito importante, escolha a porta adequada, na **aba Sketch**, clique em ferramentas, PORTA3, ou PORTA 4, assim por diante, **Figura 02**. Se você não estiver na porta correta a programação não será passada para a placa.
- Agora certifique-se que a placa do Arduino é a que você está utilizando, para isso clicando em ferramentas, vá até placa do arduino e escolha Arduino Uno, **Figura 01**.
- Próximo passo é passar a programação para a placa.

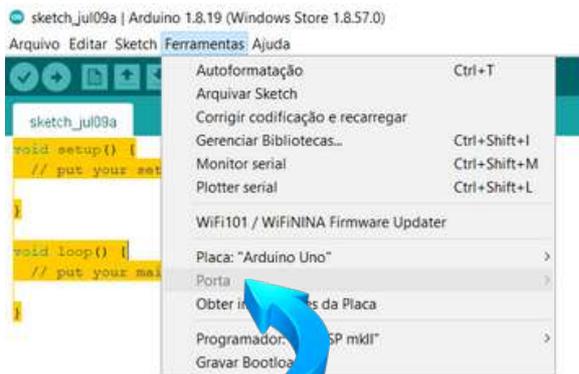


Figura 02

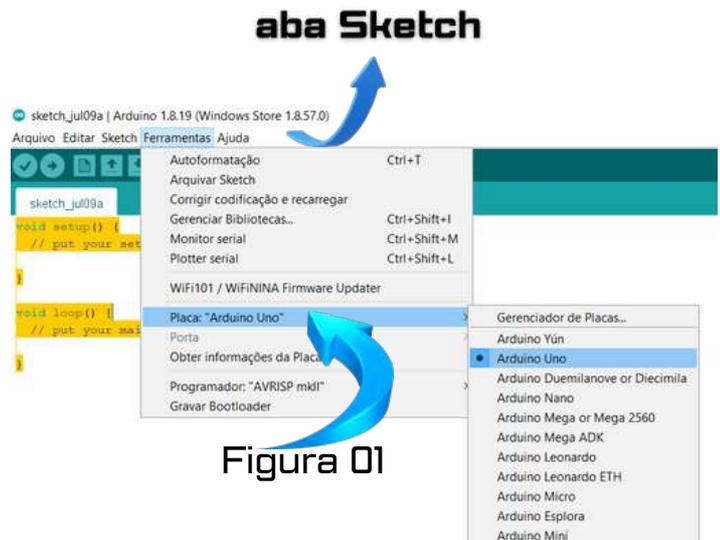


Figura 01

- Agora é passar o programa para a IDE, o nosso objetivo nesse primeiro momento é ensinar você a ler a programação e ver como ela funciona, por esse motivo iremos fornecer o código de programação.
- Copie o programa abaixo e cole dentro da aba Sketch.

```
/*  
*  
* Projeto 01 – Acender um Led.  
  SEDU-ES  
  GECEB  
*  
*****/  
  
void setup()  
{  
  pinMode(11, OUTPUT);  
}  
  
void loop()  
{  
  digitalWrite(11, HIGH);  
  
  delay(3000);  
  
  digitalWrite(11, LOW);  
  
  delay(1000);  
}
```



VOLTAR AO SUMÁRIO



- Atenção copie todo o código se esquecer qualquer caractere da programação, ele não irá funcionar.
- Na aba Sketch temos a opção de Verificar/Compilar que serve para ver se seu código está escrito corretamente, além disso tem um botão acima do espaço , que são respectivamente Verificar/Compilar e Carregar, onde o carregar serve para enviar o código para o Arduino.

-  Verificar
-  Compilar

- Se tudo cer certo, nesse momento o LED, já começa a piscar.

```
101010101
101011101
010101010
101010101
101011101
```

3.2 Lendo a Programação:

- Observe a linha de código número 3, `pinMode(11, OUTPUT)`; Você está dizendo que a placa vai ligar o LED na entrada digital 11.
- Observe a linha de código número 8, `digitalWrite(11, HIGH)`; Você está dizendo que a placa vai ligar o LED na entrada digital 11, e ele ficará ligado por `deley(3000)ms.` ou seja 3 segundos.
- Observe a linha de código número 12, `digitalWrite(11, LOW)`; Você está dizendo que a placa vai desligar o LED na entrada digital 11, e ele ficará desligado por `deley(1000)ms,` ou seja 1 segundos.
- A partir de agora, você poderá alterar a programação para que ela faça o que você quiser.

3.3 Alterando a Programação:



- Professor faça as sugestões abaixo para seus alunos:
- Peça para eles alterarem a programação, para o LED ficar 5 segundos ligado e 3 desligado.
- Peça para eles alterarem a programação, para o LED ficar 5 segundos ligado e 0,5 desligado.
- Peça para eles trocarem a entrada do led para entrada digital 13.
- Deixe que ele alterem a programação, trocando para `pinMode(13, OUTPUT)`, e faça as alterações necessárias para o LED funcionar.
- De a sugestão para os estudantes, liguem dois LEDs.
- Deixe eles pensarem, o pensamento crítico e inovador, surge a partir de situações problemas, tirando sempre os estudantes do equilíbrio para a busca da solução desejada.
- Peça para eles fazerem as ligações.



VOLTAR AO
SUMÁRIO

Etapa

Sistematização

- Número de aulas para essa atividade: 02 aulas.
- 1ª Aula: Apresentação da plataforma Arduino, Hardware e Software.
- 2ª Aula: Montagem do projeto e discussão.
- Professor, divida seus estudantes em grupos de 3 pessoas ou duplas.
- As aulas devem priorizar a discussão, o debate para tomada de decisões, afim de resolver a situação problema.

Etapa

Questionamentos

01. Funcionamento Básico: Como funciona o circuito para acender um LED? Quais são os componentes essenciais necessários e como eles interagem para fornecer energia ao LED?

02. Tipos de LEDs e Características: Existem diferentes tipos de LEDs disponíveis? Quais são suas características distintas, como cor, brilho e consumo de energia? Como selecionar o LED adequado para um projeto específico?

03. Resistor Limitador de Corrente: Por que é necessário usar um resistor limitador de corrente ao acender um LED? Como calcular o valor correto do resistor para garantir a corrente adequada que atravessa o LED e evitar danos?

04. Integração com Microcontroladores: Como um LED é controlado por um microcontrolador, como Arduino? Qual é a configuração típica dos pinos e como o estado do LED é controlado por meio do código do microcontrolador?

05. Aplicações e Utilidades: Quais são algumas das aplicações práticas de acender um LED? Além de indicadores visuais simples, onde mais os LEDs são comumente usados em projetos eletrônicos e sistemas embarcados?

06. Expansões e Melhorias: Como esse projeto básico pode ser expandido ou melhorado? Existem maneiras de adicionar funcionalidades adicionais, como piscar o LED em diferentes padrões, controlar a intensidade do brilho ou acender vários LEDs em sequência?

Respostas esperadas:



01. Funcionamento Básico: O circuito básico para acender um LED consiste em conectar o LED em série com um resistor limitador de corrente a uma fonte de energia adequada, como uma bateria ou uma saída de um microcontrolador. Quando a corrente flui pelo do circuito, o LED emite luz de acordo com sua cor e intensidade.

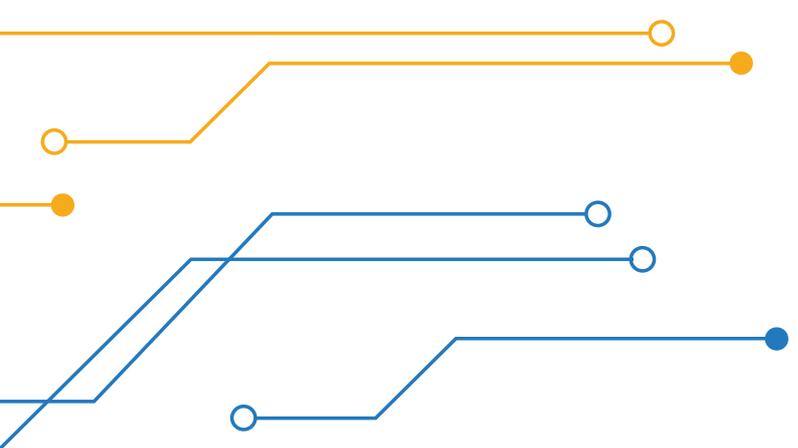
02. Tipos de LEDs e Características: Existem diversos tipos de LEDs, incluindo LEDs de diferentes cores (como vermelho, verde, azul e branco), LEDs de alto brilho, LEDs infravermelhos e LEDs ultravioleta. Cada tipo de LED tem suas próprias características elétricas e ópticas, e a seleção do LED depende das necessidades específicas do projeto.

03. Resistor Limitador de Corrente: O resistor limitador de corrente é necessário para proteger o LED contra correntes excessivas, que podem danificá-lo. Ele limita a corrente que passa pelo LED, garantindo que ele opere dentro de sua faixa de corrente segura. O valor do resistor é calculado com base na tensão de alimentação, na tensão de queda do LED e na corrente desejada.

04. Integração com Microcontroladores: Um LED é controlado por um microcontrolador através de um de seus pinos de saída digital. O microcontrolador configura o pino como HIGH para ligar o LED e como LOW para desligá-lo. O código do microcontrolador determina o comportamento do LED, como ligar, desligar, ou piscar em padrões específicos.

05. Aplicações e Utilidades: Acender um LED é útil em uma variedade de aplicações, como indicadores visuais, luzes de sinalização, iluminação de fundo para displays, efeito de iluminação em projetos de arte e decoração, e feedback visual em sistemas de controle e automação.

06. Expansões e Melhorias: Esse projeto básico pode ser expandido adicionando mais LEDs e controlando-os em conjunto para criar padrões de iluminação mais complexos. Também é possível adicionar sensores para acionar o LED em resposta a determinados eventos ou integrá-lo a outros sistemas para fornecer feedback visual.



Projeto 02

Ligar Led enquanto o Push-button ficar pressionado

Etapa da Educação Básica:

Ensino Fundamental anos finais e Ensino médio

Público Alvo:

Alunos no 9º Ano, 1ª, 2ª ou 3ª Séries.

Modalidade:

() Sequência Didática. () Projeto () Iniciação Científica.

BNCC da Computação:

[\[EF69CO02\]](#) Elaborar algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.

[\[EM13CO13\]](#) Analisar e utilizar as diferentes formas de representação e consulta a dados em formato digital para pesquisas científicas.

[\[EM13CO15\]](#) Analisar a interação entre usuários e artefatos computacionais, abordando aspectos da experiência do usuário e promovendo reflexão sobre a qualidade do uso dos artefatos nas esferas do trabalho, do lazer e do estudo.

[\[EM13CO16\]](#) Desenvolver projetos com robótica, utilizando artefatos físicos ou simuladores.

[\[EM13CO18\]](#) Planejar e gerenciar projetos integrados às áreas de conhecimento de forma colaborativa, solucionando problemas, usando diversos artefatos computacionais.

Objeto(s) de conhecimento:

- Conhecer o Push-Button
- Ligar um LED, mantendo o Push-Button acionado



VOLTAR AO
SUMÁRIO



Vamos ao projeto!

Ligar Led enquanto o Push-button ficar pressionado



Objetivo da Aula:



- Reconhecer o (push-button)
- Ligar um Led, quando o botão de pressão (push-button) estiver acionado.

Contextualização



Este projeto utiliza um botão push-button como interruptor para ligar e desligar um dispositivo ou circuito. A simplicidade e facilidade de uso desse tipo de interruptor tornando ideal para uma variedade de aplicações, desde o controle de luzes em uma casa até o acionamento de sistemas automatizados em ambientes industriais. Além disso, o baixo custo e a confiabilidade do botão push-button o tornam uma escolha popular para projetos de eletrônica.

Etapa



Orientações para o professor

Apresentando componentes da plataforma:

- ✓ Professor você, pode começar pedindo para os estudantes separarem os componentes a serem utilizados.
- ✓ Você deve comentar sobre o pushbutton, citar algumas situações de onde ele pode ser utilizado.
- ✓ Fique atento aos valores do resistor. O resistor ligado ao pushbutton, deve ser um resistor de 10 kΩ.
- ✓ Bom, agora que os estudantes já começaram se apropriar da plataforma Arduino, mãos a obra!



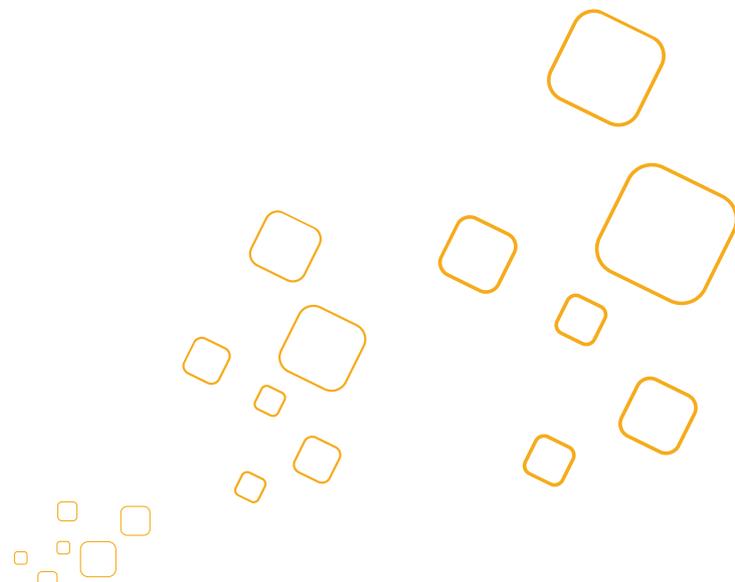
VOLTAR AO SUMÁRIO



Etapa

2. Materiais Utilizados

- 1 x PC
- 1 x Cabo USB
- 1 x Arduino Uno
- 1 x Protoboard
- 1 x Resistor 10 k Ω
- 1 x Resistores 220 Ω
- 1 x Chave Táctil (Push-button)
- 1 x LED
- 6 x Fio Jumpers



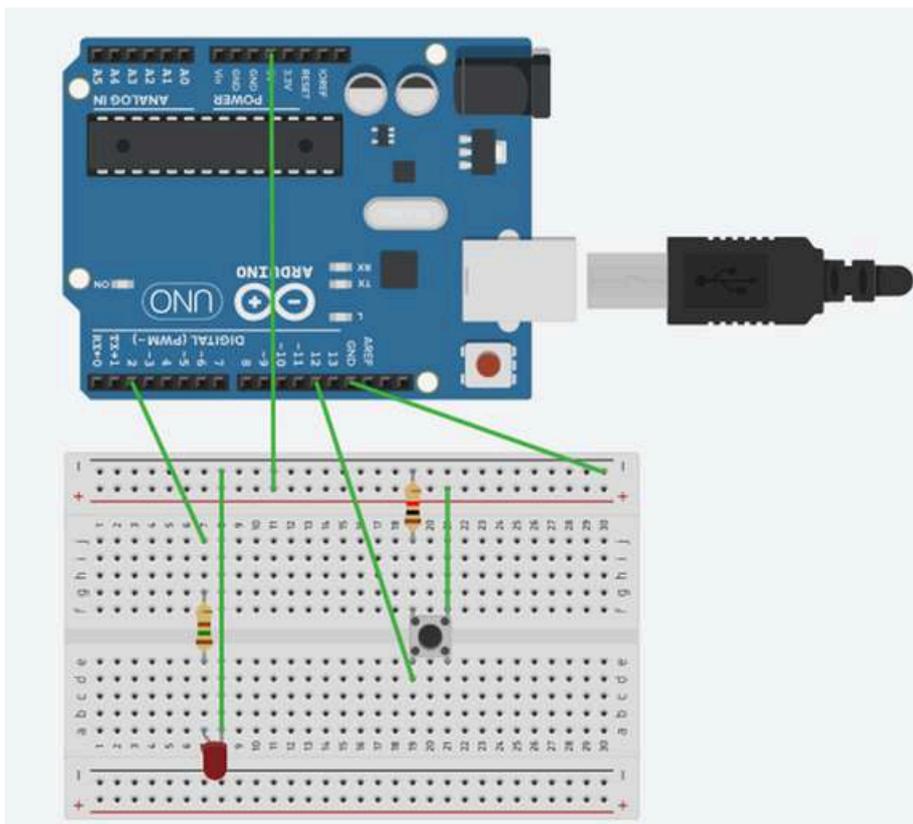
Etapa



3. Metodologia

3.1 Ligando Componentes:

- Separar os materiais a serem utilizados;
- Tomar cuidado ao ligar os dois resistores.
- Resistor de 220 Ω ligado no LED e resistor de 10 K Ω ligado push-button ;
- Destaque para as conexões nos pinos de entrada/saída digitais do Arduino;
- Faça a ligação dos componetes conforme a figura a seguir;
- Tome cuidado ao ligar o LED, perna menor ligar no negativo = GND;



**AUTODESK**
Tinkercad

Quer montar esse projeto virtualmente? Ele está disponível em:

<https://www.tinkercad.com/things/dYK9CU4GBIQ-acionamento-led-pressao>

Ou aponte o celular para o QR-code



VOLTAR AO SUMÁRIO 

3.2 Programação:

- Agora é ligar o PC, para passar a programação para a placa.
- Conecte a placa do Arduino ao PC, utilize o cabo USB.
- Esse momento professor, é muito importante, escolha a porta adequada, na **aba Sketch**, clique em ferramentas, certifique-se que a porta escolhida é a adequada.
- Agora é passar o programa para a IDE, o objetivo nesse primeiro momento professor, é ensinar os estudantes a lerem a programação e verem como ela funciona, por esse motivo iremos fornecer o código de programação.
- Peça aos alunos, que eles copiem o programa abaixo e cole dentro da aba Sketch.

```
/******  
*  
* Projeto 02 – Ligar Led enquanto botão de pressão ficar pressionado.
```



```
*****/  
  
// Projeto 02 – Push button liga led quando pressionado (pull down)
```

```
const int ledPin = 2; //led no pino 2
```

```
const int Botao = 12; //botao no pino 12
```

```
int estadoBotao; //Variavel para ler o status do pushbutton
```

```
void setup(){
```

```
    pinMode(ledPin, OUTPUT); //Pino do led será saída
```

```
    pinMode(Botao, INPUT); //Pino com botão será entrada
```

```
}
```

```
void loop(){
```

```
    estadoBotao = digitalRead(Botao); //le o estado do botão - HIGH OU LOW
```

```
    if (estadoBotao == HIGH){ //Se botão estiver pressionado (HIGH)
```

```
        digitalWrite(ledPin, HIGH); // acende o led do pino 2.
```

```
    }
```

```
    else{ //se não estiver pressionado (LOW)
```

```
        digitalWrite(ledPin, LOW); //deixa o led do pino 2 apagado
```

```
    }
```

```
}
```



 [VOLTAR AD SUMÁRIO](#)

- Atenção copie todo o código se esquecer qualquer caractere da programação, ele não irá funcionar.
- Na aba Sketch temos a opção de Verificar/Compilar que serve para ver se seu código está escrito corretamente, além disso tem um botão acima do espaço , que são respectivamente Verificar/Compilar e Carregar, onde o carregar serve para enviar o código para o Arduino.
 -  Verificar
 -  Compilar
- Agora é fazer o teste, pressione o botão push-button, se der certo o LED irá acender.
- Quando você soltar o push-button o LED irá apagar.

3.2 Lendo a Programação:

```
101010101
101011101
010101010
101010101
101011101
```

- `const int ledPin = 2; //led no pino 2 // Definido pino 2, para alimentar o LED`
- `const int Botao = 12; //botao no pino 12 // Definido pino 12 energizando o push-button.`
- `pinMode(ledPin, OUTPUT); //Pino do led será saída`
- `pinMode(Botao, INPUT); //Pino com botão será entrada`
- `if (estadoBotao == HIGH){ //Se botão estiver pressionado//O LED ficará aceso.`
- `else{ //se não estiver pressionado (LOW) // O LED ficará apagado.`

3.3 Alterando a Programação:



- Faça uma sugestão para seus estudantes:
- Peça para eles trocarem a entrada do LED, para entrada digital 5.
- Peça para eles trocarem a saída do Push-button para entrada digital 4.
- Deixe que eles alterem a programação, trocando para pinMode e faça as alterações necessárias para o LED funcionar.
- De a sugestão para seus estudantes, liguem dois LEDs, utilizando dois push-button. Deixe eles pensarem, o pensamento crítico e inovador, surge a partir de situações problemas, tirando sempre os estudantes do equilíbrio para que eles busquem a solução desejada.
- Peça para eles fazerem as ligações, e alterarem a programação, para resolverem a situação problema.



VOLTAR AO
SUMÁRIO

Etapa

Sistematização

- Número de aulas para essa atividade: 01 aula.
- Montagem do projeto e discussão.
- Professor, divida seus estudantes em grupos de 3 pessoas ou duplas.
- As aulas devem priorizar a discussão, o debate para tomada de decisões, afim de resolver a situação problema.

Etapa

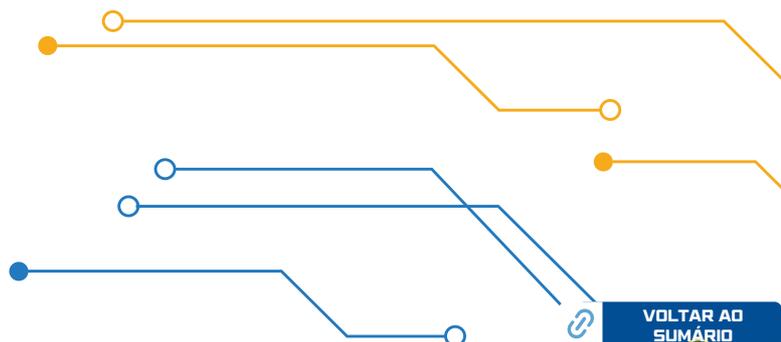
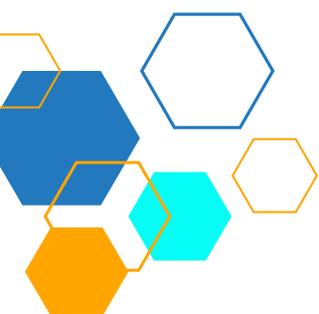
Questionamentos

01. Temporização e Controle: Como é controlada a ligação e desligamento do LED quando o push button é pressionado e liberado? Existe algum mecanismo de temporização envolvido para garantir que o LED permaneça aceso apenas enquanto o botão está pressionado?

02. Debounce e Controle de Ruído: Como lidar com o problema de bouncing do push button nesse projeto? Existem técnicas específicas de debounce que devem ser aplicadas para garantir um funcionamento estável do circuito?

03. Aplicações e Utilidades: Quais são algumas das aplicações práticas desse projeto? Ele pode ser útil em sistemas de controle onde uma ação precisa ser realizada apenas enquanto um botão está sendo pressionado? Existem cenários específicos onde esse tipo de funcionalidade é especialmente útil?

04. Expansões e Melhorias: Existem maneiras de expandir ou melhorar esse projeto básico? Por exemplo, é possível adicionar mais LEDs ou botões e criar funcionalidades adicionais, como alternar entre diferentes modos de operação?



[VOLTAR AO SUMÁRIO](#)



Respostas esperadas:

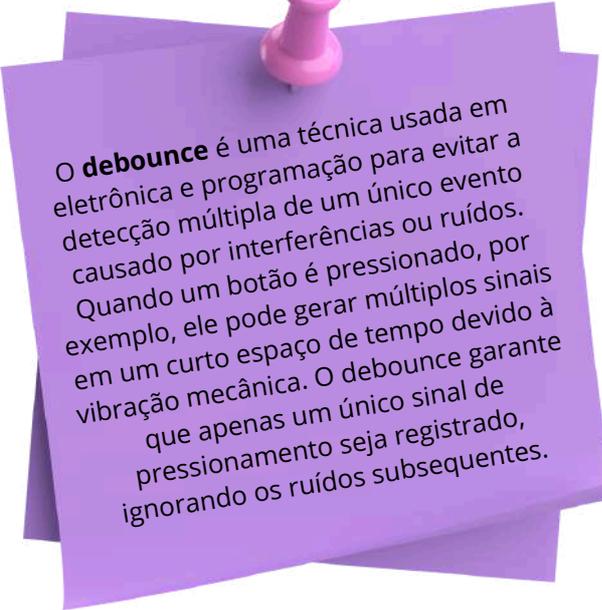


01. Temporização e Controle: A ligação e desligamento do LED são controlados pelo programa no microcontrolador. O microcontrolador verifica continuamente o estado do push button. Se estiver pressionado, ele configura o pino de saída conectado ao LED como HIGH para ligá-lo. Quando o push button é liberado, o pino de saída é configurado como LOW para desligar o LED.

02. Debounce e Controle de Ruído: Para lidar com o problema de bouncing, o código do microcontrolador implementa uma técnica de debounce. Isso pode ser feito, por exemplo, ignorando leituras de botão por um curto período de tempo após uma mudança de estado, garantindo que apenas uma transição válida seja considerada.

03. Aplicações e Utilidades: Esse projeto pode ser útil em diversas aplicações, como em dispositivos de controle onde uma ação precisa ser realizada apenas enquanto um botão está sendo pressionado, como abrir uma porta, acionar um dispositivo de segurança ou ativar temporariamente uma função em um sistema.

04. Expansões e Melhorias: O projeto básico pode ser expandido adicionando mais LEDs e push buttons, criando, por exemplo, um sistema de controle com múltiplas funções. Outras melhorias podem incluir a implementação de diferentes padrões de piscar para o LED, dependendo do tempo que o botão é pressionado.



O **debounce** é uma técnica usada em eletrônica e programação para evitar a detecção múltipla de um único evento causado por interferências ou ruídos. Quando um botão é pressionado, por exemplo, ele pode gerar múltiplos sinais em um curto espaço de tempo devido à vibração mecânica. O debounce garante que apenas um único sinal de pressionamento seja registrado, ignorando os ruídos subsequentes.



VOLTAR AO
SUMÁRIO



Projeto 03

Push-button como interruptor

Etapa da Educação Básica:

Ensino Fundamental anos finais e Ensino médio

Público Alvo:

Alunos no 9º Ano, 1ª, 2ª ou 3ª Séries.

Modalidade:

Sequência Didática. Projeto Iniciação Científica.

BNCC da Computação:

[\(EF69CO02\)](#) Elaborar algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.

[\(EM13CO13\)](#) Analisar e utilizar as diferentes formas de representação e consulta a dados em formato digital para pesquisas científicas.

[\(EM13CO15\)](#) Analisar a interação entre usuários e artefatos computacionais, abordando aspectos da experiência do usuário e promovendo reflexão sobre a qualidade do uso dos artefatos nas esferas do trabalho, do lazer e do estudo.

[\(EM13CO16\)](#) Desenvolver projetos com robótica, utilizando artefatos físicos ou simuladores.

[\(EM13CO18\)](#) Planejar e gerenciar projetos integrados às áreas de conhecimento de forma colaborativa, solucionando problemas, usando diversos artefatos computacionais.

Objeto(s) de conhecimento:

- Conhecer o Push-Button
- Ligar um LED, mantendo o Push-Buton acionado



VOLTAR AO
SUMÁRIO



Vamos ao projeto!

Push-button como interruptor



Objetivo da Aula:



- Compreender o funcionamento do push-button atuando como interruptor.
- Ligar e desligar um Led, quando o botão de pressão (push-button) for acionado.
- Diferenciar os tipos de acionamento do push-button.

Contextualização



Este projeto utiliza um botão push-button como interruptor para ligar e desligar um dispositivo ou circuito. A simplicidade e facilidade de uso desse tipo de interruptor tornam-no ideal para uma variedade de aplicações, desde o controle de luzes em uma casa até o acionamento de sistemas automatizados em ambientes industriais. Além disso, o baixo custo e a confiabilidade do botão push-button o tornam uma escolha popular para projetos de eletrônica.

Etapa



Orientações para o professor

Apresentando componentes da plataforma:

- ✓ Caro professor, você pode começar pedindo para os alunos separarem os componentes a serem utilizados.
- ✓ Comente sobre o pushbutton, resalte a diferença entre o interruptor acionado uma única vez, ou ficar pressionado continuamente.
- ✓ Fique atento aos valores do resistor. O resistor ligado ao pushbutton, deve ser um resistor de 10 KΩ.
- ✓ Bom, agora que os estudantes já começaram se apropriar da plataforma Arduino, mãos a obra!
- ✓ Professor, note que a diferença do projeto 2 para o 3 é apenas a forma de programar o funcionamento do botão.



VOLTAR AO SUMÁRIO



Etapa

2. Materiais Utilizados

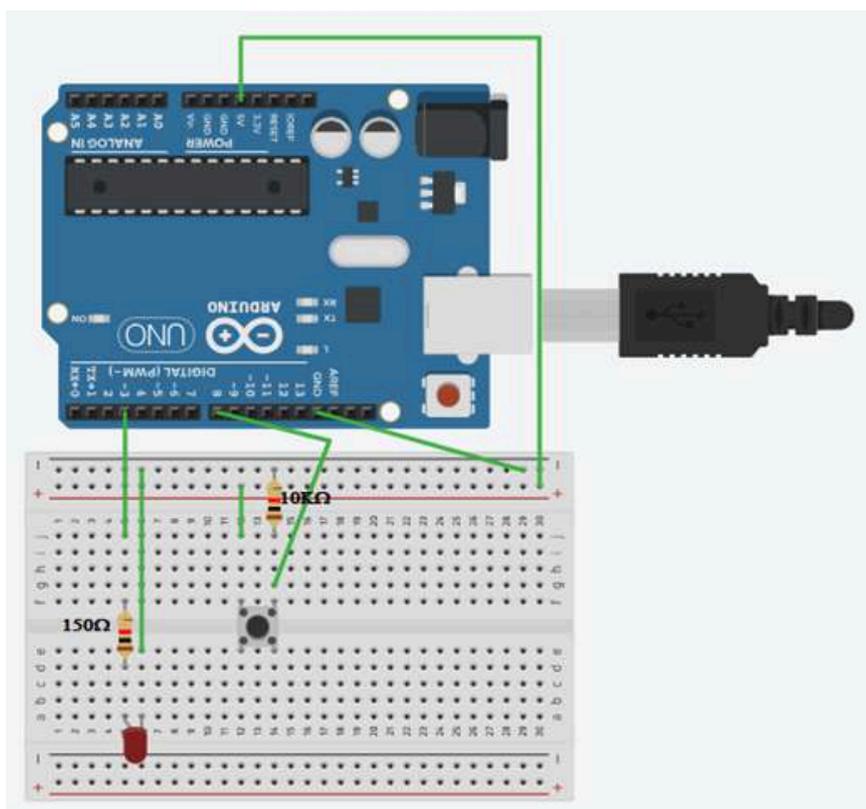
- 1 x PC
- 1 x Cabo USB
- 1 x Arduino Uno
- 1 x Protoboard
- 1 x Resistor 10 k Ω
- 1 x Resistores 150 Ω
- 1 x Chave Táctil (Push-button)
- 1 x LED
- 6 x Fio Jumpers

Etapa

3. Metodologia

3.1 Ligando Componentes:

- Separar os materiais a serem utilizados;
- Tomar cuidado ao ligar os dois resistores.
- Resistor de 150 Ω ligado no LED e resistor de 10 K Ω ligado push-button ;
- Destaque para as conexões nos pinos de entrada/saída digitais do Arduino;
- Faça a ligação dos componetes conforme a figura a seguir;
- Tome cuidado ao ligar o LED, perna menor ligar no negativo = GND;



TINKERCAD AUTODESK
Tinkercad

Quer montar esse projeto virtualmente?
Ele está disponível em:

<https://www.tinkercad.com/things/3knePIOH7hj-push-button-como-interruptor>

Ou aponte o celular para o QR-code



VOLTAR AO SUMÁRIO

3.2 Programação:

- Agora é ligar o PC, e passar a programação para a placa do Arduino.
- Conecte a placa do Arduino ao PC, utilize o cabo USB.
- Esse momento é muito importante, escolha a porta adequada, na **aba Sketch**, clique em ferramentas, certifique-se que a porta escolhida é a adequada.
- Passe o programa para a IDE, o nosso objetivo nesse primeiro momento é ensinar o aluno a ler a programação e ver como ela funciona, por esse motivo iremos fornecer o código de programação.
- Copie o programa abaixo e cole dentro da aba Sketch.

```
/******  
* Projeto 03 – O botão Push-button como interruptor  
  SEDU-ES  
  GECEB  
*  
*****/  
  
// set pinos  
const int buttonPin = 8; // número do pino pushbutton  
const int ledPin = 3; // número do pino LED  
  
// set variables  
int estado = 0; // variável para leitura do pushbutton  
int guarda_estado = LOW; // variável para armazenar valores do pushbutton  
void setup() {  
  // define o pino do Led como saída do Arduino  
  pinMode(ledPin, OUTPUT);  
  // define pino do pushbutton como entrada do Arduino:  
  pinMode(buttonPin, INPUT);  
}  
void loop(){  
  // le o estado pushbutton: ligado (HIGH) ou desligado (LOW)  
  estado = digitalRead(buttonPin);  
  
  // verifica se o botão (pushbutton) está pressionado  
  if (estado == HIGH) {  
    // inverte valor da variável variable_buttonEstado  
    guarda_estado = !guarda_estado;  
    //espera o tempo de 500ms para evitar que haja várias vezes alterações  
    delay(500);  
  }  
  if (guarda_estado == HIGH) {  
    // liga o led  
    digitalWrite(ledPin, HIGH);  
  }  
  else {  
    // desliga o led  
    digitalWrite(ledPin, LOW);  
  }  
}
```



VOLTAR AD
SUMÁRIO



- Atenção copie todo o código se esquecer qualquer caractere da programação, ele não irá funcionar.
- Na aba Sketch temos a opção de Verificar/Compilar que serve para ver se seu código está escrito corretamente, além disso tem um botão acima do espaço , que são respectivamente Verificar/Compilar e Carregar, onde o carregar serve para enviar o código para o Arduino.
-  Verificar
-  Compilar
- Agora é fazer o teste, pressione o botão push-button uma única vez, se der certo o LED irá acender.
- Quando você soltar o push-button o LED irá ficar aceso.
- Para o LED desligar, devemos pressionar o push-button.

3.2 Lendo a Programação:

```
101010101
101011101
010101010
101010101
101011101
```

- `const int buttonPin = 8; // Definição de alimentação do pushbutton`
- `const int ledPin = 3; // Definição de alimentação do LED`
- `// verifica se o botão (pushbutton) está pressionado`
- `if (estado == HIGH) {`
- `// inverte valor da variável variable_buttonEstado`
- `guarda_estado = !guarda_estado;`
- `//espera o tempo de 500ms para evitar que haja várias vezes alterações`
- `delay(500);`
- Note que o LED não desliga automaticamente, pedimos para ele ligar ou desligar após 0,5 segundos.

3.3 Alterando a Programação:



- Professor faça uma sugestão para seus estudantes:
- Peça para eles trocarem a entrada do LED, para entrada digital 6.
- Solicite que eles troquem a saída do Push-button para entrada digital 7.
- Deixe que eles alterem a programação, trocando para pinMode e faça as alterações necessárias para o LED funcionar.
- Dê a sugestão para os estudantes, ligar dois LEDs, utilizando dois push-button. Deixe eles pensarem, o pensamento crítico e inovador, surge a partir de situações problemas, tirando sempre os estudantes do equilíbrio para que eles busquem a solução desejada.
- Solicite que eles façam as ligações, e alterarem a programação, para resolverem a situação problema.



VOLTAR AO
SUMÁRIO

Etapa

Sistematização

- Número de aulas para essa atividade: 01 aula.
- Montagem do projeto e discussão.
- Professor, divida os estudantes em grupos de 3 pessoas ou duplas.
- As aulas devem priorizar a discussão, o debate para tomada de decisões, afim de resolver a situação problema.

Etapa

Questionamentos

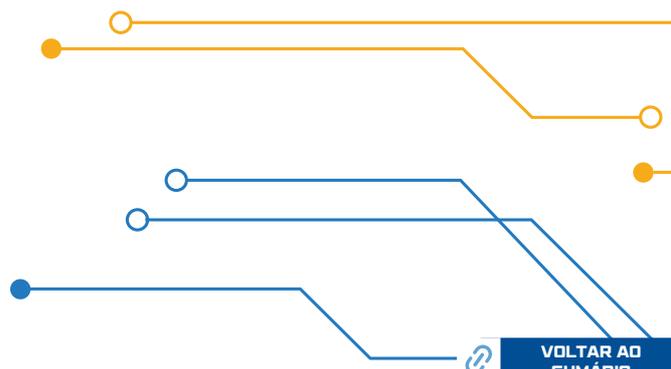
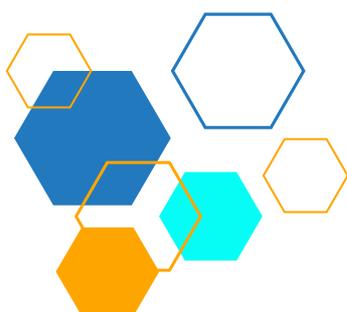
01. Funcionamento Básico: Como funciona um push button? Quais são os princípios básicos por trás de seu funcionamento e como ele é ativado?

02. Aplicações e Usos Comuns: Quais são as aplicações típicas de um push button em projetos eletrônicos? Em quais dispositivos ou sistemas ele pode ser encontrado comumente?

03. Tipos e Variedades: Existem diferentes tipos de push buttons disponíveis? Quais são suas características distintas e como eles se comparam em termos de funcionalidade e aplicação?

04. Debounce e Controle de Ruído: Por que é importante lidar com o problema de bouncing (rebote) ao usar um push button? Quais são as técnicas comuns para lidar com esse problema e garantir uma leitura precisa do estado do botão?

05. Integração com Microcontroladores: Como um push button é integrado a um microcontrolador como Arduino? Qual é a configuração típica dos pinos e como o estado do botão é lido e interpretado no código?



[VOLTAR AO SUMÁRIO](#)



Respostas esperadas:



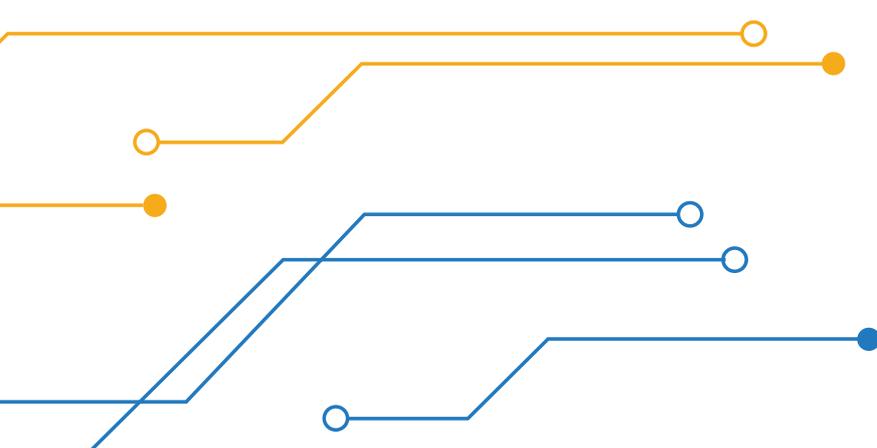
01. Funcionamento Básico: Um push button é um componente simples que consiste em uma parte móvel que é pressionada para fazer contato com uma parte fixa, completando assim um circuito elétrico. Quando pressionado, o botão fecha o circuito, permitindo que a corrente flua, e quando liberado, o circuito é aberto.

02. Aplicações e Usos Comuns: Os push buttons são amplamente utilizados em uma variedade de dispositivos eletrônicos, incluindo controles remotos, dispositivos de jogos, painéis de controle de máquinas, sistemas de segurança e interfaces de usuário em geral. Eles são comumente usados para iniciar ou parar processos, ativar funções específicas ou fazer seleções em um menu.

03. Tipos e Variedades: Existem diferentes tipos de push buttons, incluindo push buttons momentâneos, que retornam à posição original quando liberados, e push buttons de trava, que permanecem pressionados até serem pressionados novamente. Eles também podem variar em termos de tamanho, forma e cor.

04. Debounce e Controle de Ruído: O problema de bouncing ocorre quando um push button é pressionado e libera rapidamente, resultando em múltiplos contatos elétricos breves. Isso pode causar leituras inconsistentes. Técnicas de debounce, como adicionar capacitores ou software de debounce no código do microcontrolador, são comumente utilizadas para lidar com esse problema.

05. Integração com Microcontroladores: Um push button é conectado a um microcontrolador através de um dos pinos digitais. Um resistor pull-up ou pull-down é geralmente usado para garantir um estado definido quando o botão não está pressionado. No código do microcontrolador, a leitura do estado do botão é feita através da função `digitalRead()`, que retorna HIGH quando o botão é pressionado e LOW quando não é pressionado.



VOLTAR AO
SUMÁRIO



Projeto 04

Criando um Semáforo

Etapa da Educação Básica:

Ensino Fundamental anos finais e Ensino médio

Público Alvo:

Alunos no 9º Ano, 1ª, 2ª ou 3ª Séries.

Modalidade:

() Sequência Didática. () Projeto () Iniciação Científica.

BNCC da Computação:

(EF69CO02) Elaborar algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.

(EF15CO05) Codificar a informação de diferentes formas, entendendo a importância desta codificação para o armazenamento, manipulação e transmissão em dispositivos computacionais.

(EF15CO07) Conhecer o conceito de Sistema Operacional e sua importância na integração entre software e hardware.

(EM13CO13) Analisar e utilizar as diferentes formas de representação e consulta a dados em formato digital para pesquisas científicas.

(EM13CO15) Analisar a interação entre usuários e artefatos computacionais, abordando aspectos da experiência do usuário e promovendo reflexão sobre a qualidade do uso dos artefatos nas esferas do trabalho, do lazer e do estudo.

(EM13CO16) Desenvolver projetos com robótica, utilizando artefatos físicos ou simuladores.

(EM13CO18) Planejar e gerenciar projetos integrados às áreas de conhecimento de forma colaborativa, solucionando problemas, usando diversos artefatos computacionais.

Objeto(s) de conhecimento:

- Conhecer o Push-Button
- Ligar um LED, mantendo o Push-Button acionado



VOLTAR AO
SUMÁRIO



Vamos ao projeto!

Criando um Semáforo



Objetivo da Aula:



- Construir um semáforo simples utilizando uma placa Arduino, LEDs e resistores. Este projeto é ideal para iniciantes, proporcionando uma introdução prática à programação e controle de saídas digitais.

Contextualização



O projeto de um semáforo eletrônico oferece uma oportunidade prática e educativa para explorar conceitos de eletrônica e programação. Além de ser um projeto divertido, ele permite entender o funcionamento de um sistema de controle de tráfego e sua importância na segurança viária. A aplicação desse projeto pode ser útil em cenários de aprendizado em escolas, para demonstrações educativas em eventos ou até mesmo em projetos de automação residencial para simulação de controle de tráfego.

Etapa



Orientações para o professor

Apresentando componentes da plataforma:

- ✓ Prezado professor, você deve começar pedindo para os estudantes separarem os componentes a serem utilizados.
- ✓ Comente sobre o funcionamento de um semáforo. Cite exemplos do cotidiano, onde em cruzamentos temos vários semáforos que trabalham de forma sincronizada.
- ✓ Fique atento aos valores do resistor de 220Ω , ligados ao LED.
- ✓ Este projeto, seus alunos irão tirar de letra, bora trabalhar!



VOLTAR AO
SUMÁRIO



Etapa

2. Materiais Utilizados

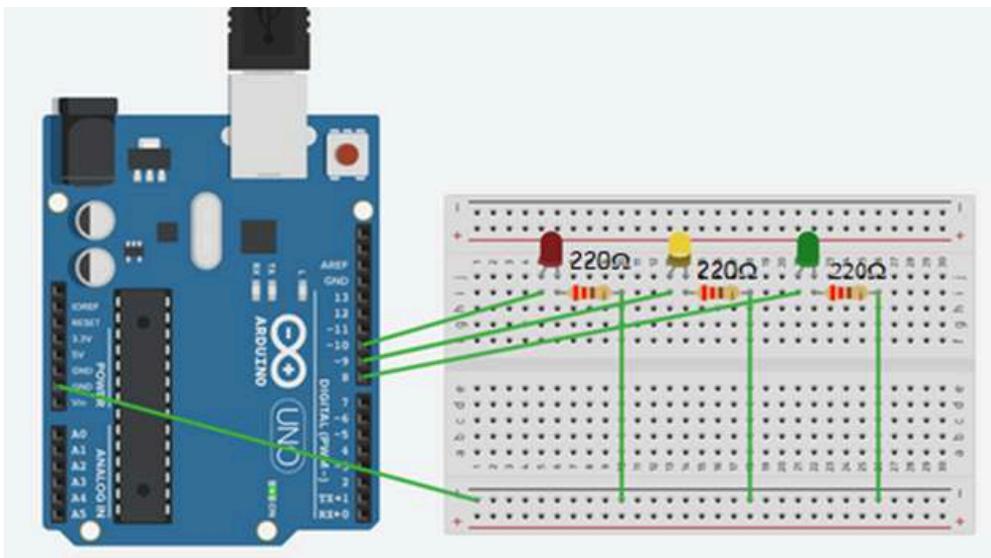
- PC
- 1 x Cabo USB
- 1 x Arduino Uno
- 1 x Protoboard
- 3 x Resistores 220 Ω
- 1 x LED Vermelho
- 1 x LED Amarelo
- 1 x LED Verde
- 7 x Fios Jumpers

Etapa

3. Metodologia

3.1 Ligando Componentes:

- Separe os materiais a serem utilizados;
- Tome cuidado ao ligar os dois resistores.
- Resistor de 220 Ω ligado nos LEDs;
- Destaque para as conexões nos pinos de entrada/saída digitais do Arduino;
- Faça a ligação dos componentes conforme a figura a seguir;
- Tome cuidado ao ligar o LED, perna menor ligar no negativo = GND;



TINKERCAD
AUTODESK
Tinkercad

Quer montar esse projeto virtualmente? Ele está disponível em:

<https://www.tinkercad.com/things/0Jm1XFBcZ61-semaforo>

Ou aponte o celular para o QR-code



VOLTAR AO SUMÁRIO



3.2 Programação:

- Agora é ligar o PC, para passar a programação para a placa do Arduino.
- Conecte a placa do Arduino ao PC, utilize o cabo USB.
- Esse momento é muito importante, escolha a porta adequada, na **aba Sketch**, clique em ferramentas, certifique-se que a porta escolhida é a adequada.
- Agora é passar o programa para a IDE, o nosso objetivo nesse primeiro caderno é ensinar o estudante a ler a programação e ver como ela funciona, por esse motivo iremos fornecer o código de programação.
- Copie o programa abaixo e cole dentro da aba Sketch.

```
/******
```

```
*
```

```
* Projeto 04 – Projeto Semáforo
```

SEDU-ES
GECEB 

```
*
```

```
*****/
```

```
// Projeto Semáforo
void setup()
{
// initialize digital pin LED_BUILTIN as an output.
pinMode(8, OUTPUT);//VERDE//
pinMode(9, OUTPUT);//AMARELA//
pinMode(10, OUTPUT);//VERMELHA//
}
void loop()
{
digitalWrite(8, HIGH);
delay(5000);
digitalWrite(8, LOW);
digitalWrite(9, HIGH);
delay(1000);
digitalWrite(9, LOW);
digitalWrite(10, HIGH);
delay(5000);
digitalWrite(10, LOW);
}
}
```



VOLTAR AO
SUMÁRIO



- Atenção copie todo o código se esquecer qualquer caractere da programação, ele não irá funcionar.
- Na aba Sketch temos a opção de Verificar/Compilar que serve para ver se seu código está escrito corretamente, além disso tem um botão acima do espaço , que são respectivamente Verificar/Compilar e Carregar, onde o carregar serve para enviar o código para o Arduino.
 -  Verificar
 -  Compilar
- Agora é fazer o teste, faça a compilação e veja o que vai acontecer.
- Se tudo deu certo, os LEDs, verde, amarelo e vermelho, irão acender e apagar conforme o tempo programado.

3.2 Lendo a Programação:

```
101010101
101011101
010101010
101010101
101011101
```

- `pinMode(8, OUTPUT);//VERDE//` alimentar LED verde na porta 8.
- `pinMode(9, OUTPUT);//AMARELA//` alimentar LED amarelo na porta 9.
- `pinMode(10, OUTPUT);//VERMELHA//` alimentar LED vermelho na porta 10.
- `delay(5000);` // LED verde, irá desligar após 5 segundos.
- `delay(1000);` // LED amarelo, irá desligar após 1 segundo.
- `delay(5000);` // LED vermelho, irá desligar após 5 segundos.



3.3 Alterando a Programação:

- Professor faça as sugestões abaixo para seus alunos:
- Peça para eles trocarem a entrada dos LEDs, para as entradas digital 6,7,8.
- Deixe que eles alterem a programação, trocando para `pinMode` e façam as alterações necessárias para os LEDs funcionarem.
- Dê a sugestão para os estudantes, liguem os dois semáforos, em um cruzamento de forma sincronizada.
- Peça para eles fazerem as ligações, e alterarem a programação, para resolverem a situação problema.
- Caso eles não consigam realizar a atividade, peça para eles fazerem o projeto 11 desse caderno, para conseguir a programação, basta usar a Inteligência Artificial, ChatGPT.



VOLTAR AO
SUMÁRIO



Etapa

Sistematização

- Número de aulas para essa atividade: 01 aula.
- Montagem do projeto e discussão.
- Professor, divida seus estudantes em grupos de 3 pessoas ou duplas.
- Suas aulas devem priorizar a discussão, o debate para tomada de decisões, afim de resolver a situação problema.

Etapa

Questionamentos

01. Princípio de Funcionamento: Como funciona um semáforo? Quais são os diferentes estados em que um semáforo pode estar e como ele controla o fluxo do tráfego?

02. Componentes e Construção: Quais são os componentes essenciais necessários para construir um semáforo simples? Como esses componentes interagem para produzir os diferentes estados de sinalização?

03. Controle e Temporização: Como é feito o controle do semáforo para alternar entre os diferentes estados? É utilizado um temporizador para regular a duração de cada estado? Como isso é implementado na prática?

04. Aplicações e Utilidades: Além de regular o tráfego de veículos em vias públicas, quais são outras aplicações possíveis para semáforos? Eles são usados em sistemas de controle de tráfego ferroviário, controle de acesso em estacionamentos, ou existem outras aplicações menos conhecidas?

05. Desafios e Considerações de Segurança: Quais são os desafios envolvidos na criação e implantação de semáforos? Como garantir a segurança e eficácia do sistema em diferentes condições ambientais e de tráfego?

Respostas esperadas:



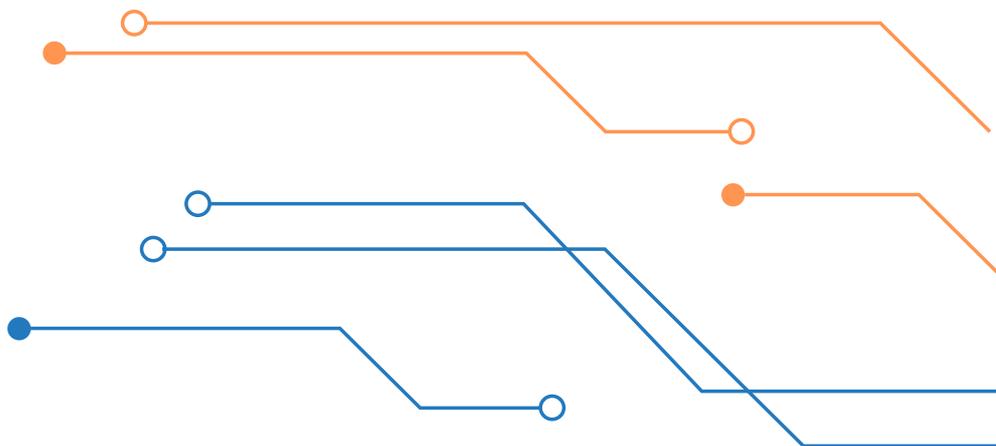
01. Princípio de Funcionamento: Um semáforo funciona alternando entre diferentes estados de sinalização para controlar o fluxo do tráfego. Os estados típicos são verde, amarelo e vermelho. O verde indica que os veículos podem prosseguir, o amarelo indica uma transição entre verde e vermelho, e o vermelho indica que os veículos devem parar. Esses estados são controlados por um sistema eletrônico que usa temporizadores para determinar a duração de cada estado.

02. Componentes e Construção: Os componentes essenciais para construir um semáforo simples incluem LEDs ou lâmpadas incandescentes para cada cor (vermelho, amarelo e verde), resistores para limitar a corrente, um microcontrolador ou circuito de temporização para controlar os LEDs e talvez um sensor de presença de veículos para ajustar os tempos de sinalização.

03. Controle e Temporização: O controle do semáforo é geralmente feito por um microcontrolador que executa um programa para alternar entre os diferentes estados de sinalização. Os temporizadores são usados para determinar quanto tempo cada estado deve durar. Por exemplo, o verde pode durar mais tempo em uma avenida principal do que em uma rua lateral menos movimentada.

04. Aplicações e Utilidades: Além de regular o tráfego de veículos em vias públicas, os semáforos são utilizados em diversos contextos, como controle de tráfego em cruzamentos, em entradas de estacionamentos, em corredores de metrô, em aeroportos e em zonas de pedestres. Eles são essenciais para garantir a segurança e a eficiência do fluxo de tráfego.

05. Desafios e Considerações de Segurança: Alguns dos desafios na criação de semáforos incluem a garantia da sincronização adequada entre os semáforos em um sistema de tráfego interligado, a resistência a condições climáticas adversas, como chuva ou neve, e a consideração de necessidades específicas, como a passagem segura de pedestres. A segurança é uma prioridade, e os semáforos devem ser projetados e instalados de forma a minimizar o risco de acidentes.



Projeto 05

Sensor de temperatura

Etapa da Educação Básica:

Ensino Fundamental anos finais e Ensino médio

Público Alvo:

Alunos no 9º Ano, 1ª, 2ª ou 3ª Séries.

Modalidade:

() Sequência Didática. () Projeto () Iniciação Científica.

BNCC da Computação:

(EF69CO02) Elaborar algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.

(EM13CO13) Analisar e utilizar as diferentes formas de representação e consulta a dados em formato digital para pesquisas científicas.

(EM13CO15) Analisar a interação entre usuários e artefatos computacionais, abordando aspectos da experiência do usuário e promovendo reflexão sobre a qualidade do uso dos artefatos nas esferas do trabalho, do lazer e do estudo.

(EM13CO16) Desenvolver projetos com robótica, utilizando artefatos físicos ou simuladores.

(EM13CO18) Planejar e gerenciar projetos integrados às áreas de conhecimento de forma colaborativa, solucionando problemas, usando diversos artefatos computacionais.

Objeto(s) de conhecimento:

- Conhecer o Sensor de temperatura.
- Ligar o Sensor de temperatura, para medir a temperatura de objetos variados.



VOLTAR AO
SUMÁRIO



Vamos aos projetos!

Sensor de Temperatura



Objetivo da Aula:



- Entender o funcionamento do sensor de temperatura.
- Reconhecer o sensor de temperatura.
- Compreender quais são as aplicações do sensor de temperatura.

Contextualização



O uso de um sensor de temperatura como interruptor permite automatizar o controle de dispositivos com base nas variações de temperatura ambiente. Isso é útil em situações em que é necessário ativar ou desativar equipamentos, como ventiladores ou aquecedores, dependendo da temperatura ambiente. Além disso, essa automação contribui para a eficiência energética e o conforto do ambiente.

Etapa



Orientações para o professor

Apresentando componentes da plataforma:

- ✓ Professor, você pode começar pedindo para os estudantes separarem os componentes a serem utilizados.
- ✓ Comente sobre o sensor de temperatura, fale como ele funciona, diga que não existe somente esse sensor, existem outros tipos de sensores, recomenda-se a leitura dos questionamentos no final desse projeto.
- ✓ É importante destacar as aplicações desse sensor. Professor, recomendamos que você faça a leitura da a página 19.
- ✓ Mãos a obra!



VOLTAR AO
SUMÁRIO



Etapa

2. Materiais Utilizados

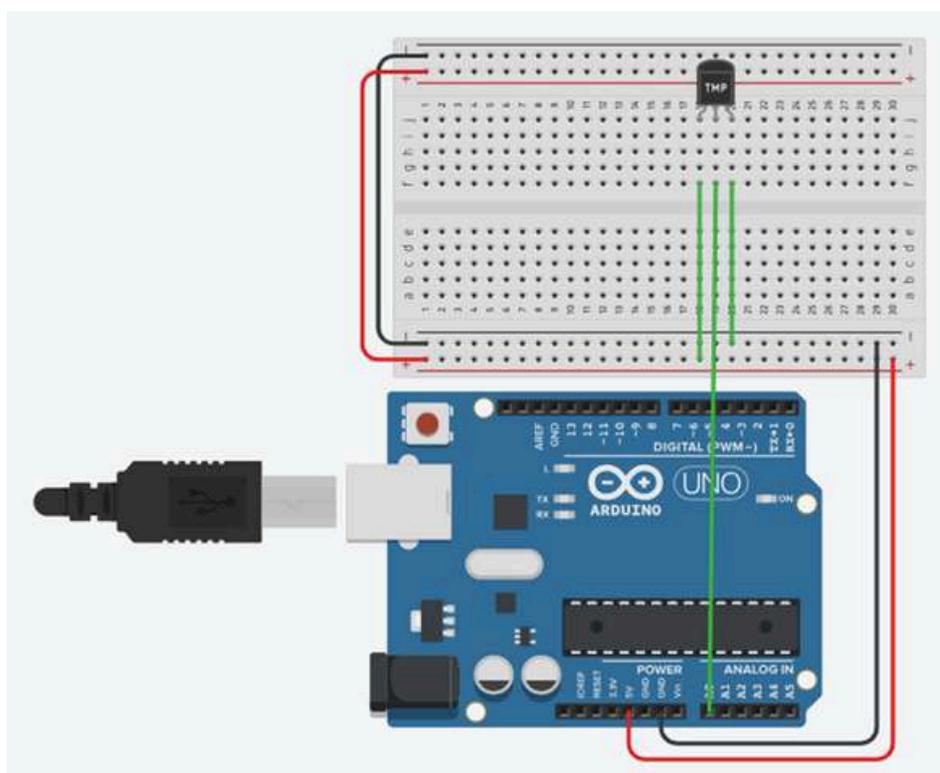
- 1 x PC
- 1 x Cabo USB
- 1 x Arduino Uno
- 1 x Protoboard
- 1 x Sensor de temperatura LM35
- 7 x Fios Jumpers

Etapa

3. Metodologia

3.1 Ligando Componentes:

- Separe os materiais a serem utilizados;
- Tome cuidado ao ligar o sensor de temperatura;
- Note que a parte arredondada do sensor fica para trás;
- Tome cuidado ao ligar os três pinos do Sensor de Temperatura;
- Atenção se você inverter o fio, o sensor poderá queimar;
- Destaque para as conexões nos pinos de entrada/saída digitais do Arduino;
- Faça a ligação dos componentes conforme a figura a seguir.



TINKERCAD
AUTODESK
Tinkercad

Quer montar esse projeto virtualmente?
Ele está disponível em:

<https://www.tinkercad.com/things/1ZCgXbjjAb2-sensor-de-temperatura-lm35>

Du aponte o celular para o QR-code



VOLTAR AO SUMÁRIO

3.2 Programação:

- Agora é ligar o PC, para passar a programação para a placa .
- Conecte a placa do Arduino ao PC, utilize o cabo USB.
- Esse momento é muito importante, escolha a porta adequada, na **aba Sketch**, clique em ferramentas, certifique-se que a porta escolhida é a adequada.
- Agora é passar o programa para a IDE, iremos fornecer o código de programação.
- Copie o programa abaixo e cole dentro da aba Sketch.
- Observe as constantes novas que surgiram na programação
- Aprenda a fazer a leitura e as alterações da programação, que é importante!

```
/******  
*  
* PROJETO 05 - SENSOR TEMPERATURA LM35  
* SEDU – ES   
* GECEB  
*  
*****/  
  
const int sensorLM35 = 0;  
float temperatura = 0;  
int ADClido = 0;  
  
void setup(){  
  Serial.begin(9600);  
  analogReference(INTERNAL);  
}  
  
void loop(){  
  ADClido = analogRead(sensorLM35);  
  temperatura = ADClido * 0.1075268817;  
  
  Serial.print("Temperatura = ");  
  Serial.print(temperatura);  
  Serial.println(" *C");  
  delay(1000);  
}
```



- Atenção copie todo o código se esquecer qualquer caractere da programação, ele não irá funcionar.
- Na aba Sketch temos a opção de Verificar/Compilar que serve para ver se seu código está escrito corretamente, além disso tem um botão acima do espaço , que são respectivamente Verificar/Compilar e Carregar, onde o carregar serve para enviar o código para o Arduino.
 -  Verificar
 -  Compilar
- Agora é fazer o teste, é só ligar.
- Se tudo der certo, o sensor rá começar a registrar a temperatura do ambiente, será exibido na tela.

3.2 Lendo a Programação:

```
101010101
101011101
010101010
101010101
101011101
```

- `const int sensorLM35 = 0; // Entrada sensor no Arduino: A0 - Analógica`
- `float temperatura = 0; // variável tipo float - inicia no 0`
- `Serial.begin(9600); // taxa comunicação da placa com o computador`
- `ADClido = analogRead(sensorLM35); //faz a leitura analógica A0`
- `temperatura = ADClido * 0.1075268817; //no Leonardo use 0.4887585532 (constante para cada 1°C)`
- `Serial.print("Temperatura = "); // mostra valor na tela`
- `Serial.println(" *C"); // mostra valor na tela em graus Celsius`
- `delay(1000); // mostra o valor na tela a cada 1 segundo`



3.3 Alterando a Programação:

- Professor, faça as seguintes sugestões para seus estudantes:
- Peça para eles trocarem a entrada Sensor de Temperatura, para entrada Analógica 3.
- Deixe que eles alterem a programação, `const int sensorLM35` e faça as alterações necessárias para o sensor funcionar.
- Peça para eles fazerem as ligações, e alterarem a programação, de forma que o sensor passe a medir a temperatura a cada 5 segundos.
- Deixe os estudantes medirem a temperatura de outros objetos.
- Sugestão, situação problema para medir a temperatura corporal.
- Peça para eles colocarem o dedo no sensor, para medir a temperatura do “corpo”.



VOLTAR AO
SUMÁRIO



Etapa



Sistematização

- Número de aulas para essa atividade: 01 aula.
- Montagem do projeto e discussão.
- Professor, divida seus estudantes em grupos de 3 pessoas ou duplas.
- Suas aulas devem priorizar a discussão, o debate para tomada de decisões, afim de resolver a situação problema.

Etapa



Questionamentos



- 1. Princípio de Funcionamento:** Como funciona um sensor de temperatura? Quais são os diferentes tipos de sensores de temperatura e como eles medem a temperatura ambiente?
- 2. Precisão e Calibração:** Quão preciso é um sensor de temperatura? Eles requerem calibração periódica para garantir precisão nas medições? Como a calibração é realizada e qual é a importância desse processo?
- 3. Aplicações e Casos de Uso:** Quais são algumas das aplicações mais comuns para sensores de temperatura? Eles são amplamente utilizados em sistemas de aquecimento, ventilação e ar condicionado (HVAC), monitoramento ambiental, controle de processos industriais, ou existem outras aplicações menos conhecidas?
- 4. Configuração e Interface:** Como configurar um sensor de temperatura para integrá-lo a um sistema eletrônico? Eles são compatíveis com interfaces digitais, como I2C ou SPI? Existe uma ampla variedade de sensores de temperatura disponíveis no mercado?
- 5. Comparação com Outros Sensores:** Como os sensores de temperatura se comparam a outros tipos de sensores, como termopares ou termistores? Quais são as vantagens e desvantagens de cada tipo de sensor em diferentes cenários de uso?

{I2C (Inter-Integrated Circuit): O I2C é um protocolo serial de dois fios que permite a comunicação entre vários dispositivos em um barramento comum. Ele usa um par de fios para transmitir dados: um para enviar sinais de clock (SCL) e outro para enviar dados (SDA). O I2C é usado para conectar dispositivos de baixa velocidade, como sensores, displays, memórias EEPROM e outros dispositivos periféricos em um sistema.

SPI (Serial Peripheral Interface): O SPI é um protocolo serial síncrono que permite a comunicação de alta velocidade entre um mestre (geralmente um microcontrolador) e vários dispositivos escravos. Ele usa quatro linhas de comunicação: MOSI (Master Out Slave In), MISO (Master In Slave Out), SCK (Serial Clock) e SS (Slave Select). O SPI é comumente usado para comunicação de alta velocidade entre dispositivos como microcontroladores, memórias flash, conversores analógico-digitais (ADCs) e displays TFT.}



VOLTAR AD
SUMÁRIO



Respostas esperadas:



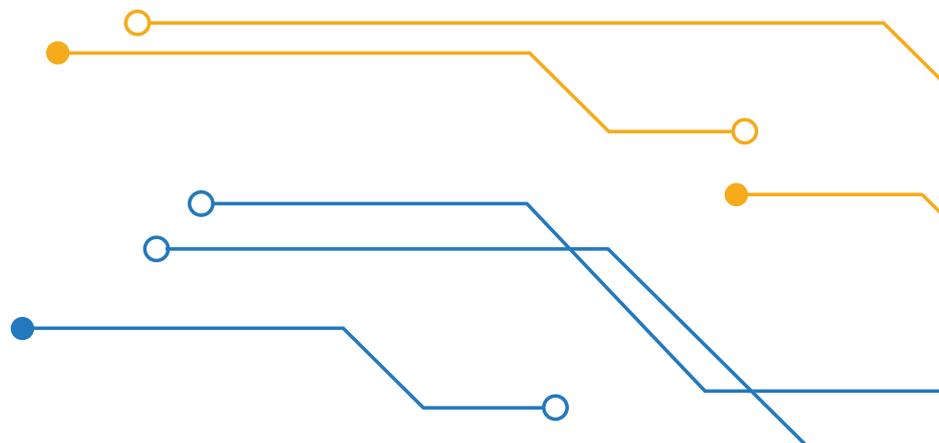
1. Princípio de Funcionamento: Os sensores de temperatura operam com base em diferentes princípios, incluindo variação de resistência (termistores), variação de tensão (termopares), variação de voltagem (diodos de junção PN) ou variação de corrente (transistores de junção bipolar). Eles medem a temperatura ambiente convertendo a mudança física resultante em um sinal elétrico que pode ser interpretado e exibido.

2. Precisão e Calibração: A precisão de um sensor de temperatura pode variar dependendo do tipo e da qualidade do sensor. Sensores mais sofisticados geralmente possuem maior precisão. A calibração é importante para garantir que as medições sejam precisas ao longo do tempo. Ela pode ser feita comparando as leituras do sensor com medições de temperatura conhecidas e ajustando os valores conforme necessário.

3. Aplicações e Casos de Uso: Os sensores de temperatura são amplamente utilizados em uma variedade de aplicações, incluindo controle de temperatura em sistemas HVAC, monitoramento de processos industriais, detecção de falhas em equipamentos eletrônicos, monitoramento ambiental em laboratórios e pesquisa científica, entre outros. Eles também são encontrados em dispositivos de consumo, como termômetros digitais, termômetros infravermelhos e termostatos.

4. Configuração e Interface: A configuração de um sensor de temperatura pode variar dependendo do tipo e do modelo do sensor. Alguns sensores se comunicam através de interfaces digitais como I2C ou SPI, enquanto outros podem fornecer uma saída analógica que requer conversão para um formato digital. A seleção do sensor e a interface com o microcontrolador ou sistema eletrônico dependem dos requisitos específicos do projeto.

5. Comparação com Outros Sensores: Comparado a outros sensores de temperatura, como termopares ou termistores, os sensores baseados em diodos de junção PN oferecem uma resposta mais linear e uma faixa de temperatura mais ampla. No entanto, cada tipo de sensor tem suas próprias vantagens e desvantagens em termos de precisão, faixa de temperatura, custo e compatibilidade com diferentes ambientes e aplicações. A escolha do sensor ideal depende das necessidades específicas do projeto.



VOLTAR AO
SUMÁRIO



Projeto 06

Sensor Ultrassônico HC-SR04

Etapa da Educação Básica:

Ensino Fundamental anos finais e Ensino médio

Público Alvo:

Alunos no 9º Ano, 1ª, 2ª ou 3ª Séries.

Modalidade:

() Sequência Didática. () Projeto () Iniciação Científica.

BNCC da Computação:

(EF69CO02) Elaborar algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.

(EM13CO13) Analisar e utilizar as diferentes formas de representação e consulta a dados em formato digital para pesquisas científicas.

(EM13CO15) Analisar a interação entre usuários e artefatos computacionais, abordando aspectos da experiência do usuário e promovendo reflexão sobre a qualidade do uso dos artefatos nas esferas do trabalho, do lazer e do estudo.

(EM13CO16) Desenvolver projetos com robótica, utilizando artefatos físicos ou simuladores.

(EM13CO18) Planejar e gerenciar projetos integrados às áreas de conhecimento de forma colaborativa, solucionando problemas, usando diversos artefatos computacionais.

Objeto(s) de conhecimento:

- Conhecer o Sensor Ultrassônico.
- Ligar o Sensor Ultrassônico, para medir a distância do sensor a variados objetos.



VOLTAR AO
SUMÁRIO



Vamos aos projetos!

Sensor Ultrassônico HC-SR04



Objetivo da Aula:



- Reconhecer o Sensor Ultrassônico
- Reconhecer as aplicações do Sensor Ultrassônico
- Entender como o Sensor Ultrassônico funciona

Contextualizaç



ão
O sensor ultrassônico HC-SR04 é amplamente utilizado devido à sua precisão e facilidade de uso na detecção de objetos a curtas distâncias. Este projeto aproveita essas características para criar um sistema de detecção de obstáculos, que pode ser aplicado em diversos contextos, desde sistemas de segurança veicular até robôs móveis e automação industrial.

Etapa



Orientações para o professor

Apresentando componentes da plataforma:

- ✓ Professor você pode começar pedindo para os estudantes separarem os componentes a serem utilizados.
- ✓ Comente sobre o Sensor Ultrassônico, citar algumas situações de onde ele pode ser utilizado.
- ✓ É importante fazer a leitura da página 25, lá destacamos a importância e o funcionamento do sensor ultrassônico.
- ✓ Leia os questionamentos no final desse projeto, para discutir com seus estudantes.
- ✓ Mãos a obra!



VOLTAR AO
SUMÁRIO



Etapa

2. Materiais Utilizados

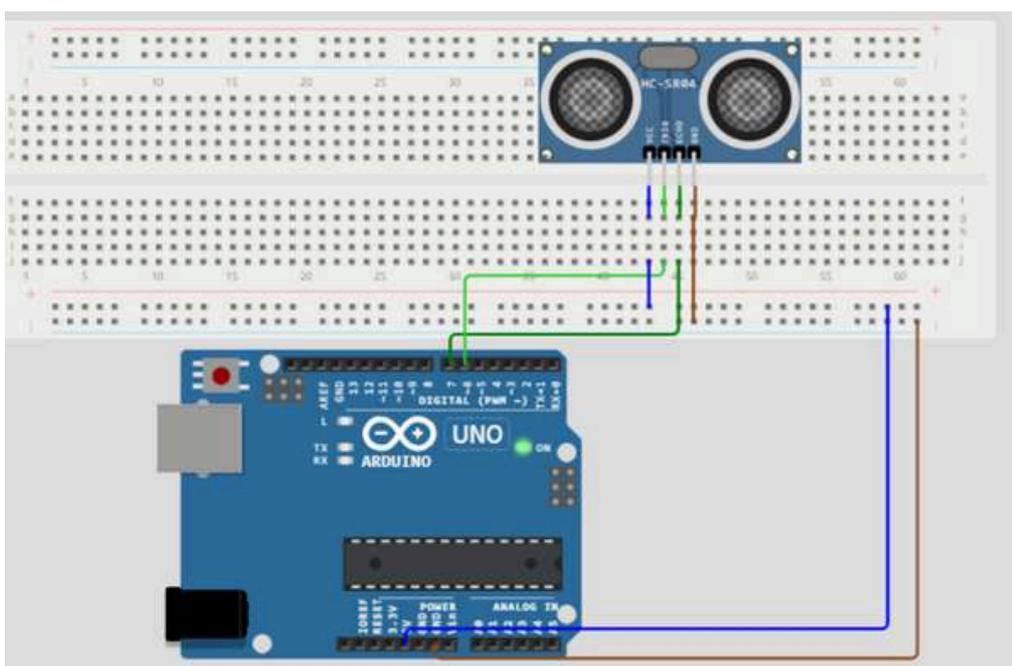
- PC
- 1 x Cabo USB
- 1 x Arduino Uno
- 1 x Protoboard
- 1 x Sensor ultrassônico
- 6 x Fios Jumpers

Etapa

3. Metodologia

3.1 Ligando Componentes:

- Separe os materiais a serem utilizados;
- Tome cuidado ao ligar o sensor ultrassônico;
- Atenção ao ligar os quatro pinos do Sensor ultrassônico;
- Cuidado, se você inverter o fio, o sensor poderá queimar;
- Destaque para as conexões nos pinos de entrada/saída digitais do Arduino;
- Faça a ligação dos componentes conforme a figura a seguir.



WOKWI

Quer montar esse projeto virtualmente?
Ele está disponível em:
<https://wokwi.com/projects/390460086883049473>

Ou aponte o celular para o QR-code



VOLTAR AO SUMÁRIO



3.2 Programação:

- Agora é ligar o PC, e passar a programação para a placa do Arduino.
- Conecte a placa do Arduino ao PC, utilize o cabo USB.
- Esse momento é muito importante, escolha a porta adequada, na **aba Sketch**, clique em ferramentas, certifique-se que a porta escolhida é a adequada.
- Agora é passar o programa para a IDE, iremos fornecer o código de programação.
- Copie o programa abaixo e cole dentro da aba Sketch.
- Observe as constantes novas que surgiram na programação
- Aprenda a fazer a leitura e as alterações da programação, que é importante!

```
//*****  
* PROJETO 06 - SENSOR ULTRASSÔNICO HC-SR04  
* SEDU - ES   
* GECEB  
*****/
```

ATENÇÃO! COPIE DESSE PONTO PARA BAIXO.

```
#define TRIGGER_PIN 7  
#define ECHO_PIN 6  
  
void setup() {  
  Serial.begin(9600);  
  pinMode(TRIGGER_PIN, OUTPUT);  
  pinMode(ECHO_PIN, INPUT);  
}  
  
void loop() {  
  long duration, distance;  
  
  digitalWrite(TRIGGER_PIN, LOW);  
  delayMicroseconds(2);  
  digitalWrite(TRIGGER_PIN, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(TRIGGER_PIN, LOW);  
  
  duration = pulseIn(ECHO_PIN, HIGH);  
  
  distance = (duration * 0.0343) / 2;  
  
  Serial.print("Distancia: ");  
  Serial.print(distance);  
  Serial.println(" cm");  
  
  delay(1000);  
}
```



VOLTAR AO
SUMÁRIO



- Atenção copie todo o código, se esquecer qualquer caractere da programação, ele não irá funcionar.
- Na aba Sketch temos a opção de Verificar/Compilar que serve para ver se seu código está escrito corretamente, além disso tem um botão acima do espaço , que são respectivamente Verificar/Compilar e Carregar, onde o carregar serve para enviar o código para o Arduino.
 -  Verificar
 -  Compilar
- Agora é fazer o teste.
- Se tudo der certo, ele irá começar a registrar a distância do sensor ao obstáculo.
- Aproxime a mão na frente do sensor, veja se ele irá registrar a distância correta.

3.2 Lendo a Programação:

```
101010101
101011101
010101010
101010101
101011101
```

- `#define TRIGGER_PIN 7 - // Define os pinos para o sensor ultrassônico HC-SR04`
- `#define ECHO_PIN 6 - // Define os pinos para o sensor ultrassônico HC-SR04`
- `Serial.begin(9600); // Inicia a comunicação serial`
- `pinMode(TRIGGER_PIN, OUTPUT); // Configura o pino de trigger como saída`
- `pinMode(ECHO_PIN, INPUT); // Configura o pino de echo como entrada`
- `long duration, distance; // Variáveis para armazenar a duração do pulso e a distância`
- `duration = pulseIn(ECHO_PIN, HIGH); // Mede a duração do pulso no pino de echo`
- `distance = (duration * 0.0343) / 2; // Calcula a distância em centímetros`
- `Serial.print("Distancia: "); // Exibe a distância medida no monitor serial`
- `delay(1000); // Aguarda 1 segundo antes de realizar a próxima leitura`



3.3 Alterando a Programação:

- Professor, faça as seguintes sugestões para seus estudantes:
- Peça para eles levarem o sensor para lugares diferentes.
- Deixe eles tentarem medir a distância entre do sensor e outros obstáculos como, uma parede a perna de uma mesa, ou de uma cadeira.
- Eles irão perceber que a o sensor de distância, apresenta certas limitações.
- É muito importante que os estudantes explorem os sensores. Conhecer o funcionamento e as limitações dos sensores vai fazer seus projetos futuros mais elaborados como a construção de robos, funcionarem de forma correta.



VOLTAR AO
SUMÁRIO



Etapa

Sistematização

- Número de aulas para essa atividade: 01 aula.
- Montagem do projeto e discussão.
- Professor, divida seus estudantes em grupos de 3 pessoas ou duplas.
- Suas aulas devem priorizar a discussão, o debate para tomada de decisões, afim de resolver a situação problema.

Etapa

Questionamentos

1. Princípio de Funcionamento: Como funciona um sensor ultrassônico? Quais são os componentes essenciais desse tipo de sensor e como eles interagem para medir distâncias?

2. Precisão e Limitações: Quão preciso é um sensor ultrassônico na medição de distâncias? Existem limitações em relação ao tipo de superfície ou condições ambientais que podem afetar sua precisão?

3. Aplicações e Casos de Uso: Quais são algumas das aplicações mais comuns para sensores ultrassônicos? Eles são amplamente utilizados em robótica, automação industrial, sistemas de estacionamento de veículos, ou existem outras aplicações menos conhecidas?

4. Configuração e Programação: Como configurar e programar um sensor ultrassônico para medir distâncias? Existem bibliotecas ou recursos específicos que facilitam a integração desse sensor em projetos eletrônicos?

5. Comparação com Outros Sensores: Como os sensores ultrassônicos se comparam a outros tipos de sensores de distância, como sensores de infravermelho ou sensores a laser? Quais são as vantagens e desvantagens de cada tipo de sensor em diferentes cenários de uso?



Respostas esperadas:



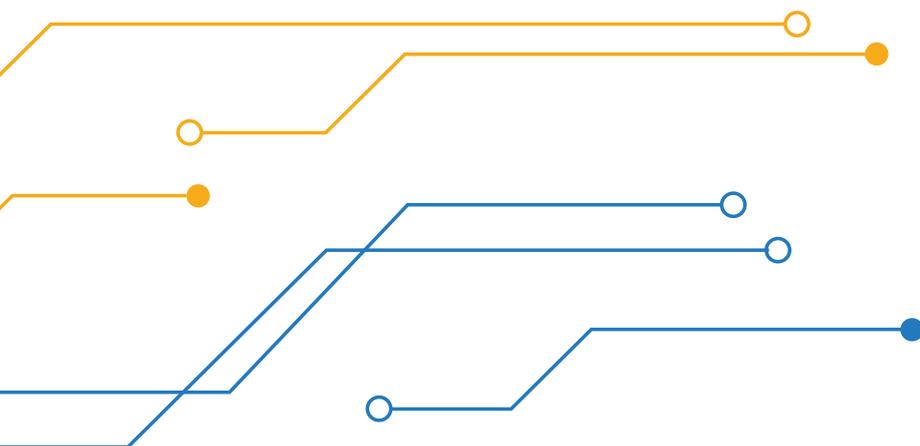
1. Princípio de Funcionamento: Um sensor ultrassônico emite pulsos de ondas sonoras ultrassônicas e mede o tempo que leva para os pulsos serem refletidos de volta por um objeto. Com base nesse tempo e na velocidade do som no ar, o sensor calcula a distância até o objeto. Os componentes essenciais incluem um transmissor ultrassônico e um receptor, além de eletrônica para controlar a emissão e recepção dos pulsos.

2. Precisão e Limitações: A precisão de um sensor ultrassônico pode variar, geralmente dependendo da qualidade do sensor e das condições ambientais. Superfícies irregulares, materiais absorventes de som e mudanças na temperatura e umidade do ar podem afetar a precisão das medições. Além disso, sensores ultrassônicos podem não funcionar bem em ambientes com muitos obstáculos ou com alta incidência de ruído acústico.

3. Aplicações e Casos de Uso: Sensores ultrassônicos têm uma variedade de aplicações, desde sistemas de estacionamento de veículos até robôs autônomos, sistemas de navegação em ambientes internos, detecção de obstáculos em drones, entre outros. Eles são amplamente utilizados em automação industrial para detecção de presença e medição de distância em linhas de produção.

4. Configuração e Programação: Para configurar um sensor ultrassônico, é necessário alimentá-lo com energia elétrica e conectar os pinos de transmissão e recepção ao microcontrolador ou outro dispositivo de processamento. A programação geralmente envolve enviar pulsos ultrassônicos e medir o tempo de resposta para calcular a distância. Existem bibliotecas disponíveis para plataformas como Arduino que simplificam esse processo.

5. Comparação com Outros Sensores: Comparado a sensores de infravermelho, os sensores ultrassônicos têm uma faixa de detecção mais ampla e são menos afetados por condições de iluminação. Em comparação com sensores a laser, eles tendem a ser mais acessíveis e podem funcionar bem em ambientes internos. No entanto, os sensores ultrassônicos podem ter uma precisão menor em distâncias muito curtas e são mais suscetíveis a interferências sonoras.



Projeto 07

Sensor LDR

Etapa da Educação Básica:

Ensino Fundamental anos finais e Ensino médio

Público Alvo:

Alunos no 9º Ano, 1ª, 2ª ou 3ª Séries.

Modalidade:

() Sequência Didática. () Projeto () Iniciação Científica.

BNCC da Computação:

(EF69CO02) Elaborar algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.

(EM13CO13) Analisar e utilizar as diferentes formas de representação e consulta a dados em formato digital para pesquisas científicas.

(EM13CO15) Analisar a interação entre usuários e artefatos computacionais, abordando aspectos da experiência do usuário e promovendo reflexão sobre a qualidade do uso dos artefatos nas esferas do trabalho, do lazer e do estudo.

(EM13CO16) Desenvolver projetos com robótica, utilizando artefatos físicos ou simuladores.

(EM13CO18) Planejar e gerenciar projetos integrados às áreas de conhecimento de forma colaborativa, solucionando problemas, usando diversos artefatos computacionais.

Objeto(s) de conhecimento:

- Conhecer o Sensor LDR.
- Ligar o Sensor LDR para medir a luminosidade de diversos ambientes.

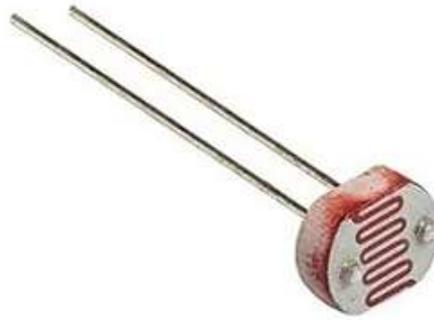


VOLTAR AO
SUMÁRIO



Vamos aos projetos!

Sensor LDR



Objetivo da Aula:



- Reconhecer o Sensor LDR
- Reconhecer as aplicações do LDR
- Entender como o Sensor LDR funciona

Contextualização



O uso de um sensor LDR (Light Dependent Resistor) em um projeto de controle de iluminação oferece uma maneira eficaz de automatizar o ajuste da luminosidade com base na quantidade de luz ambiente. Isso é particularmente útil em ambientes internos onde a iluminação natural varia ao longo do dia, permitindo economia de energia ao ajustar a iluminação artificial de acordo com a necessidade.

Etapa



Orientações para o professor

Apresentando componentes da plataforma:

- ✓ Professor, você pode começar pedindo para os estudantes separarem os componentes a serem utilizados.
- ✓ Comente sobre o Sensor LDR, cite algumas situações de onde ele pode ser utilizado.
- ✓ É importante destacar as aplicações desse sensor. Professor, recomendamos que você faça a leitura da a página 21.
- ✓ Leia os questionamentos no final desse projeto, para discutir com seus alunos.
- ✓ Mãos a obra!



VOLTAR AO
SUMÁRIO



Etapa

2. Materiais Utilizados

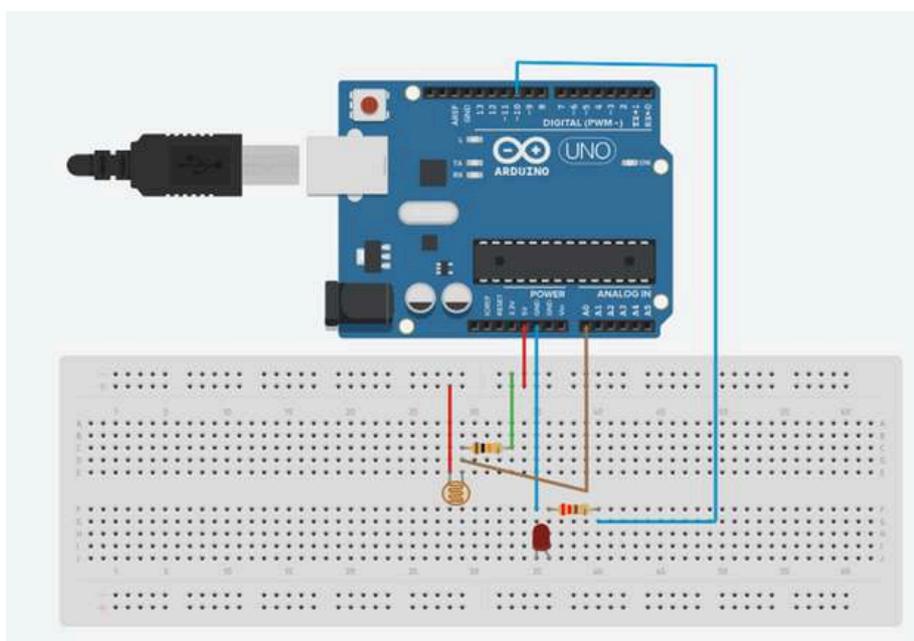
- 1 x PC
- 1 x Cabo USB
- 1 x Arduino Uno
- 1 x Protoboard
- 1 x LED
- 1 x Resistores 220 Ω para o LED
- 1 x Resistores 10 K Ω para o LDR
- 6 x Fios Jumpers
- 1 x Sensor LDR

Etapa

3. Metodologia

3.1 Ligando Componentes:

- Separe os materiais a serem utilizados;
- Tome cuidado ao ligar o sensor LDR;
- Atenção se você inverter o fio, o sensor poderá queimar;
- Destaque para as conexões nos pinos de entrada/saída digitais do Arduino;
- Faça a ligação dos componentes conforme a figura a seguir.



TINKERCAD
AUTODESK
Tinkercad

Quer montar esse projeto virtualmente?
Ele está disponível em:
<https://www.tinkercad.com/things/exxxFXD2c0L-sensor-ldr>

Ou aponte o celular para o QR-code



[VOLTAR AO SUMÁRIO](#)

3.2 Programação:

- Agora é ligar o PC, para passar a programação para o Arduino .
- Conecte a placa do Arduino ao PC, utilize o cabo USB.
- Esse momento é muito importante, escolha a porta adequada, na **aba Sketch**, clique em ferramentas, certifique-se que a porta escolhida é a adequada.
- Agora é passar o programa para a IDE, iremos fornecer o código de programação.
- Copie o programa abaixo e cole dentro da aba Sketch.
- Observe as constantes novas que surgiram na programação
- Aprenda a fazer a leitura e as alterações da programação, que é importante!

```
/******
```

```
*
```

```
* PROJETO 07 - SENSOR LDR
```

```
* SEDU - ES 
  GECEB
```

```
*
```

```
*****/
```

```
int led = 10; //Atribui a porta digital 10 a variável led
```

```
int ldr = A0; //Atribui A0 a variável ldr
```

```
int valorldr = 0; //Declara a variável valorldr como inteiro
```

```
void setup() {
```

```
  pinMode(led, OUTPUT); //Define led (pino digital 10) como saída
```

```
  pinMode(ldr, INPUT); //Define ldr (pino analógico A0) como saída
```

```
  Serial.begin(9600); //Inicialização da comunicação serial, com velocidade de
  comunicação de 9600
```

```
}
```

```
void loop() {
```

```
  valorldr = analogRead(ldr); //Lê o valor do sensor ldr e armazena na variável valorldr
```

```
  Serial.println(valorldr); //Imprime na serial os dados de valorldr
```

```
  if ((valorldr) < 500) { //Se o valor de valorldr for menor que 500:
```

```
    digitalWrite(led, HIGH); //Coloca led em alto para acioná-lo
```

```
  }
```

```
  else { //Senão:
```

```
    digitalWrite(led, LOW); //Coloca led em baixo para que o mesmo desligue ou permaneça
    desligado
```

```
  }
```

```
}
```



VOLTAR AO
SUMÁRIO



- Atenção copie todo o código se esquecer qualquer caractere da programação, ele não irá funcionar.
- Na aba Sketch temos a opção de Verificar/Compilar que serve para ver se seu código está escrito corretamente, além disso tem um botão acima do espaço , que são respectivamente Verificar/Compilar e Carregar, onde o carregar serve para enviar o código para o Arduino.
-  Verificar
 -  Compilar
- Agora é fazer o teste. Inicie o programa, se tudo der certo a lâmpada ira estar apagada, mas ela pode acender conforme a iluminação fique mais fraca.
- Coloque um obstáculo em cima do sensor e veja se a luz vai acender.

3.2 Lendo a Programação:

```
101010101
101011101
010101010
101010101
101011101
```

- `int led = 10; // Atribui a porta digital 10 à variável led`
- `int ldr = A0; // Atribui A0 à variável ldr`
- `int valorldr = 0; // Declara a variável valorldr como inteiro`
- `pinMode(led, OUTPUT); // Define led (pino digital 10) como saída`
- `pinMode(ldr, INPUT); // Define ldr (pino analógico A0) como entrada`
- `Serial.begin(9600); // Inicialização da comunicação serial, com velocidade de comunicação de 9600`
- `valorldr = analogRead(ldr); // Lê o valor do sensor ldr e armazena na variável valorldr`
- `Serial.println(valorldr); // Imprime na serial os dados de valorldr`
- `if (valorldr < 500) // Se o valor de valorldr for menor que 500:`
- `digitalWrite(led, HIGH); // Coloca led em alto para acioná-lo`
- `else // Senão:`
- `digitalWrite(led, LOW); // Coloca led em baixo para que o mesmo desligue ou permaneça desligado`

3.3 Alterando a Programação:



- Professor, faça as seguintes sugestões para seus estudantes:
- Peça para eles fazerem as ligações, e alterarem a programação, de forma que o sensor LDR seja acionado quando faltar 80% da luminosidade no local.
- Deixe os estudantes medirem a luminosidade dos locais com o sensor LDR e ver o que irá acontecer.
- Sugestão, leve seus estudantes para fora da escola e veja como o sensor irá se comportar na presença da luz solar.
- O sensor LDR é muito útil para trabalhar com controle de Iluminosidade.
- Tente desenvolver um projeto que utilize esse sensor.
- Leia as questões e respostas esperadas, isso vai te dar ideias para realização de vários projetos, envolvendo o LDR.



VOLTAR AO
SUMÁRIO



Etapa



Sistematização

- Número de aulas para essa atividade: 01 aula.
- Montagem do projeto e discussão.
- Professor divida seus estudantes em grupos de 3 pessoas ou duplas.
- As aulas devem priorizar a discussão, o debate para tomada de decisões, afim de resolver a situação problema.

Etapa



Questionamentos

1. Princípio de funcionamento: Como o sensor LDR detecta a luz? Ele varia sua resistência elétrica em resposta à intensidade luminosa? Como essa variação de resistência é utilizada para medir a luz ambiente?

2. Sensibilidade e faixa de resposta: O quão sensível é o sensor LDR à luz? Ele pode detectar variações sutis na luminosidade ou é mais adequado para detectar mudanças significativas? Qual é a faixa de resposta do sensor em termos de intensidade luminosa?

3. Aplicações e ambientes de uso: Onde os sensores LDR são mais comumente utilizados? Eles são adequados para ambientes externos ou internos? Quais são as considerações de uso em ambientes com diferentes níveis de luminosidade?

4. Interferências e fontes de erro: Quais fatores podem interferir nas medições do sensor LDR? Ele é afetado por fontes de luz não desejadas, como luzes artificiais ou reflexos? Como mitigar esses efeitos para obter medições mais precisas?

5. Integração e projetos práticos: Como os sensores LDR podem ser integrados a projetos eletrônicos? Quais são algumas aplicações práticas dos sensores LDR, como controle de iluminação automático, detecção de presença, sistemas de segurança, entre outros?



[VOLTAR AO SUMÁRIO](#)



Respostas esperadas:



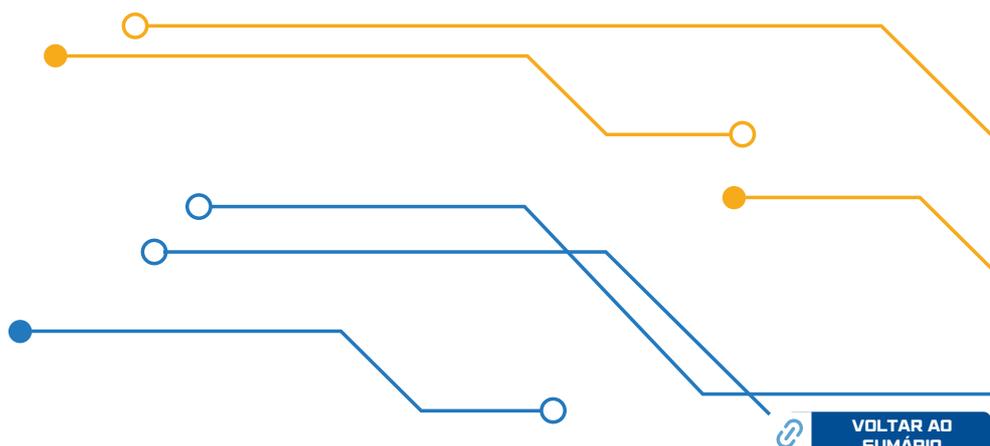
01. Princípio de funcionamento: O sensor LDR opera com base na variação da resistência elétrica em resposta à intensidade luminosa. Quando exposto à luz, a resistência do LDR diminui; quando em escuridão, sua resistência aumenta. Essa mudança na resistência pode ser medida e usada para determinar a quantidade de luz ambiente.

02. Sensibilidade e faixa de resposta: A sensibilidade do sensor LDR à luz pode variar dependendo do modelo e da qualidade do sensor. Em geral, eles podem detectar uma ampla faixa de intensidades luminosas, desde luz ambiente fraca até luz solar direta. No entanto, sua sensibilidade pode ser limitada em condições extremas de luminosidade.

03. Aplicações e ambientes de uso: Os sensores LDR são comumente utilizados em projetos de automação residencial, sistemas de iluminação automática, detectores de luminosidade para câmeras, sistemas de segurança com detecção de luz, entre outros. Eles são adequados para uso em ambientes internos e externos, embora possam ser afetados por condições climáticas adversas, como chuva ou neblina.

04. Interferências e fontes de erro: Fontes de luz não desejadas, como luzes artificiais ou reflexos, podem interferir nas medições do sensor LDR, causando leituras imprecisas. Além disso, a exposição direta à luz solar pode saturar o sensor, levando a leituras errôneas. Para minimizar esses efeitos, é importante posicionar o sensor adequadamente e usar filtros ou invólucros para reduzir a interferência externa.

05. Integração e projetos práticos: Os sensores LDR podem ser integrados a projetos eletrônicos de diversas maneiras, geralmente como parte de um circuito de controle automático de iluminação. Eles são usados em sistemas que ativam luzes automaticamente em ambientes escuros e as desligam quando há luz suficiente. Além disso, são utilizados em projetos de arte interativa, controle de exposição em fotografia, e em experimentos científicos para medir a intensidade luminosa em diferentes ambientes.



VOLTAR AO
SUMÁRIO



Projeto 08

Sensor Umidade do solo com LED

Etapa da Educação Básica:

Ensino Fundamental anos finais e Ensino médio

Público Alvo:

Alunos no 9º Ano, 1ª, 2ª ou 3ª Séries.

Modalidade:

() Sequência Didática. () Projeto () Iniciação Científica.

BNCC da Computação:

[\(EF69CO02\)](#) Elaborar algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.

[\(EM13CO13\)](#) Analisar e utilizar as diferentes formas de representação e consulta a dados em formato digital para pesquisas científicas.

[\(EM13CO15\)](#) Analisar a interação entre usuários e artefatos computacionais, abordando aspectos da experiência do usuário e promovendo reflexão sobre a qualidade do uso dos artefatos nas esferas do trabalho, do lazer e do estudo.

[\(EM13CO16\)](#) Desenvolver projetos com robótica, utilizando artefatos físicos ou simuladores.

[\(EM13CO18\)](#) Planejar e gerenciar projetos integrados às áreas de conhecimento de forma colaborativa, solucionando problemas, usando diversos artefatos computacionais.

Objeto(s) de conhecimento:

- Conhecer o Sensor Umidade.
- Ligar o Sensor de umidade para medir a umidade do solo em alguns ambientes.

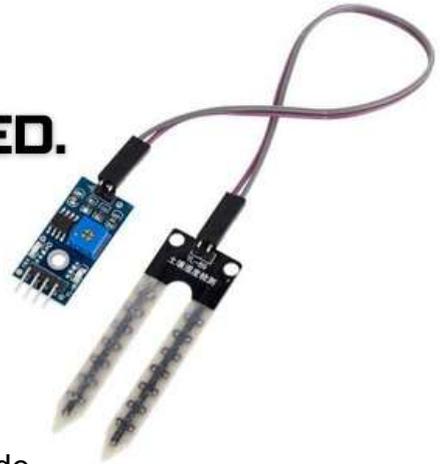


VOLTAR AO
SUMÁRIO



Vamos aos projetos!

Sensor Umidade do solo com LED.



Objetivo da Aula:



- Reconhecer o Sensor de Umidade
- Reconhecer as aplicações do Sensor de Umidade
- Entender como o Sensor de Umidade funciona

Contextualização



O projeto de um monitor de umidade do solo com Arduino é uma forma eficaz de garantir o crescimento saudável de plantas em ambientes internos ou externos. Ao monitorar a umidade do solo, os agricultores e jardineiros podem determinar com precisão quando as plantas precisam de água, evitando tanto a irrigação excessiva quanto a insuficiente. Isso resulta em economia de água, melhor saúde das plantas e maior eficiência no cultivo.

Etapa



Orientações para o professor

Apresentando componentes da plataforma:

- ✓ Caro professor, você pode começar pedindo para os estudantes separarem os componentes a serem utilizados.
- ✓ Comente sobre o Sensor de umidade, cite algumas situações de onde ele pode ser utilizado.
- ✓ É importante destacar as aplicações desse sensor. Professor, recomendamos que você faça a leitura da a página 22.
- ✓ Leia os questionamentos no final desse projeto, para discutir com seus alunos.
- ✓ Mão a obra!



VOLTAR AO
SUMÁRIO



Etapa

2. Materiais Utilizados

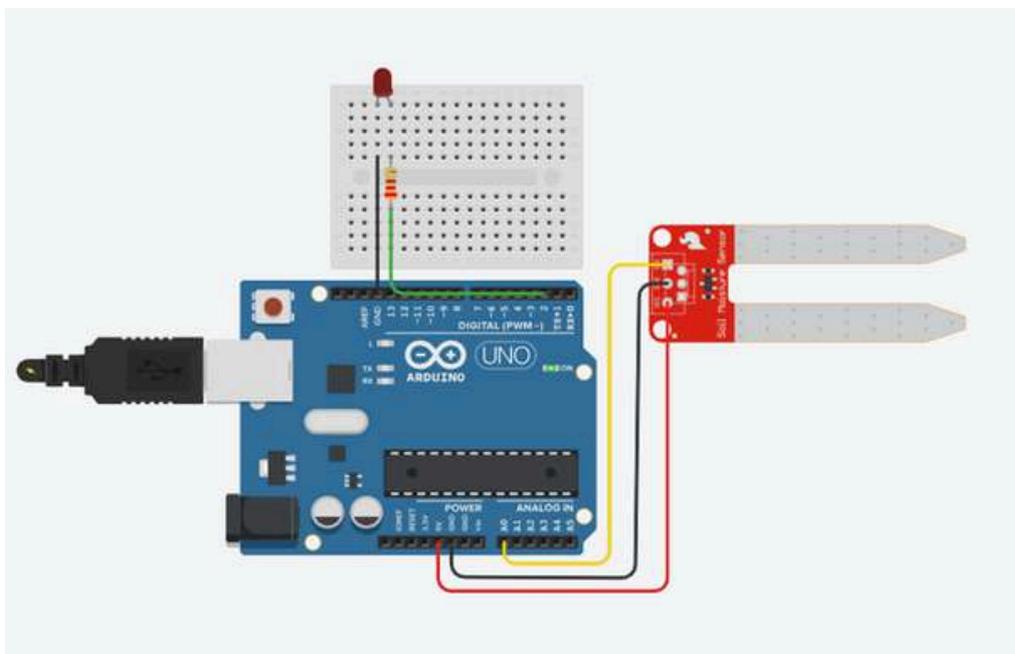
- 1 x PC
- 1 x Cabo USB
- 1 x Arduino Uno
- 1 x Protoboard
- 1 x LED
- 1 x Resistores 220 Ω para o LED
- 5 x Fios Jumpers
- 1 x Sensor de Umidade

Etapa

3. Metodologia

3.1 Ligando Componentes:

- Separe os materiais a serem utilizados;
- Tome cuidado ao ligar o sensor de umidade;
- Tome cuidado ao ligar os três pinos do Sensor de umidade;
- Atenção se você inverter o fio, o sensor poderá queimar, observe a posição dos fios, na figura abaixo;
- Destaque para as conexões nos pinos de entrada/saída digitais do Arduino;
- Peça aos estudantes que façam a ligação dos componentes conforme a figura a seguir.



TINKERCAD
AUTODESK
Tinkercad

Quer montar esse projeto virtualmente? Ele está disponível em:

<https://www.tinkercad.com/things/gWa05ST7vUa-sensor-de-umidade-com-led>

Ou aponte o celular para o QR-code



VOLTAR AO SUMÁRIO

3.2 Programação:

- Agora é ligar o PC, para .
- Conecte a placa do Arduino ao PC, utilize o cabo USB.
- Esse momento é muito importante, escolha a porta adequada, na **aba Sketch**, clique em ferramentas, certifique-se que a porta escolhida é a adequada.
- Agora é passar o programa para a IDE, iremos fornecer o código de programação.
- Copie o programa abaixo e cole dentro da aba Sketch.
- Observe as constantes novas que surgiram na programação
- Aprenda a fazer a leitura e as alterações da programação, que é importante!

```
/******
```

```
* Projeto 08 – Sensor Umidade do solo com LED.
```

```
*  
```

```
*****/
```

```
const int limiteUmidade = 50;
int umidadeSolo = 0;
const int pinoSensor = A0;
const int LED = 2;
```

```
void setup()
{
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
}
```

```
void loop()
{
  umidadeSolo = analogRead(pinoSensor);
```

```
// Converte a variação do sensor de 0 a 1023 para 0 a 100
umidadeSolo = map(umidadeSolo, 1023, 0, 0, 100);
```

```
// Porcentagem de umidade do solo
Serial.println(umidadeSolo);
```

```
// Verifica se umidade está no limite estipulado
```

```
if( umidadeSolo > limiteUmidade ) {
  digitalWrite(LED, HIGH);
}
else
{
  digitalWrite(LED, LOW);
}
```

```
delay(1000);
```

```
}
```



[VOLTAR AO SUMÁRIO](#)

- Atenção copie todo o código se esquecer qualquer caractere da programação, ele não irá funcionar.
- Na aba Sketch temos a opção de Verificar/Compilar que serve para ver se seu código está escrito corretamente, além disso tem um botão acima do espaço , que são respectivamente Verificar/Compilar e Carregar, onde o carregar serve para enviar o código para o Arduino.
 -  Verificar
 -  Compilar
- Agora é fazer o teste, coloque a ponta metálica do sensor em contato com a água, se tudo der certo, o LED irá apagar.

3.2 Lendo a Programação:

```
101010101
101011101
010101010
101010101
101011101
```

- `const int limiteUmidade = 50;` - Define o limite de umidade do solo em 50%. Se a umidade medida estiver acima desse limite, o LED será ligado.
- `int umidadeSolo = 0;` - Variável para armazenar o valor lido do sensor de umidade.
- `const int pinoSensor = A0;` - Define o pino analógico ao qual o sensor de umidade está conectado.
- `const int LED = 2;` - Define o pino digital ao qual o LED está conectado.
- `void setup()` - Função de configuração que é executada uma vez quando o Arduino é inicializado. Define o pino do LED como saída e inicia a comunicação serial.
- `void loop()` - Função principal que é executada repetidamente.
- `umidadeSolo = analogRead(pinoSensor);` - Lê o valor analógico do sensor de umidade e armazena na variável `umidadeSolo`.
- `umidadeSolo = map(umidadeSolo, 1023, 0, 0, 100);` - Converte o valor lido do sensor de 0 a 1023 para uma escala de 0 a 100 (porcentagem de umidade).
- `Serial.println(umidadeSolo);` - Imprime a umidade do solo no monitor serial.
- `if (umidadeSolo > limiteUmidade) { digitalWrite(LED, HIGH); }` - Se a umidade do solo for maior que o limite definido, o LED é ligado; caso contrário, ele é desligado.
- `delay(1000);` - Aguarda 1 segundo antes de ler novamente a umidade do solo. Isso evita leituras muito frequentes e reduz o processamento desnecessário.

3.3 Alterando a Programação:



- Professor, faça as seguintes sugestões para seus estudantes:
- Peça para eles Trocarem a porcentagem de umidade de 50 para 70%;
- Deixe que eles alterem a programação, `const int limiteUmidade = 50;` e faça as alterações necessárias para o sensor funcionar;
- Peça para eles fazerem as ligações, e alterarem a programação, de forma que o sensor passe a medida para, 5 segundos;
- Deixe os estudantes medirem a umidade do solo, de alguma planta;
- Sugestão, situação problema medir a taxa de umidade de três vasos de planta;
- Arrume três vasos de planta, regue ele de formas diferentes e veja o que acontece.

Etapa

Sistematização

- Número de aulas para essa atividade: 01 aula.
- Montagem do projeto e discussão.
- Professor, divida seus estudantes em grupos de 3 pessoas ou duplas.
- As aulas devem priorizar a discussão, o debate para tomada de decisões, afim de resolver a situação problema.

Etapa

Questionamentos

1. Princípio de funcionamento: Como os sensores de umidade no solo detectam a umidade? Eles medem diretamente a quantidade de água presente no solo ou utilizam alguma outra propriedade física para inferir a umidade?

2. Precisão e calibração: Quão precisos são os sensores de umidade no solo? Eles requerem calibração ou ajustes para garantir medições precisas? Como a calibração é realizada e qual é a frequência recomendada para calibrar esses sensores?

3. Instalação e posicionamento: Qual é a melhor maneira de instalar e posicionar os sensores de umidade no solo para obter resultados precisos? Eles devem ser enterrados a uma profundidade específica ou posicionados em determinada distância das raízes das plantas?

4. Interferências e falsos positivos: Quais fatores podem interferir nas medições dos sensores de umidade no solo? Existem fontes de erro comuns que podem levar a falsas leituras de umidade? Como mitigar esses problemas para obter resultados mais confiáveis?

5. Aplicações e benefícios: Além da agricultura e jardinagem, quais são outras aplicações potenciais dos sensores de umidade no solo? Como esses sensores contribuem para a eficiência hídrica, a saúde das plantas e a sustentabilidade ambiental em diferentes contextos?



VOLTAR AO
SUMÁRIO



Respostas esperadas:



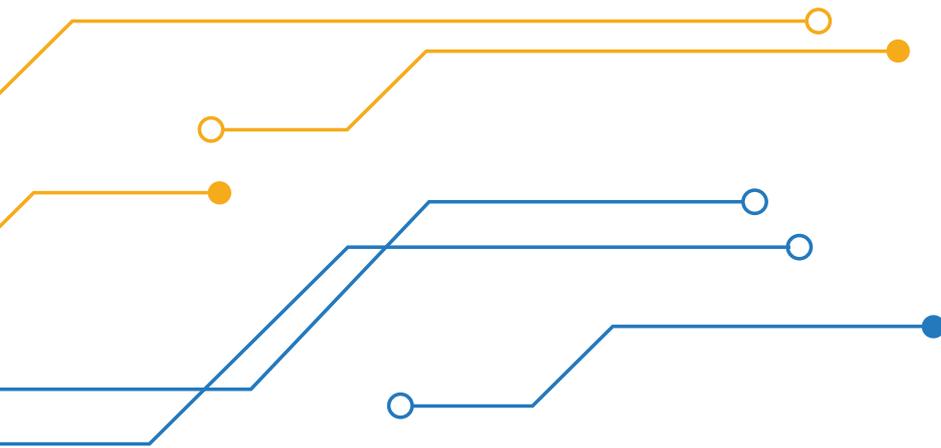
1. Princípio de funcionamento: Os sensores de umidade no solo geralmente utilizam um princípio resistivo, onde a umidade do solo afeta a resistência elétrica do sensor. Quanto mais úmido o solo, menor será a resistência medida pelo sensor. Alguns sensores também podem usar princípios capacitivos, ópticos ou térmicos para medir a umidade do solo.

2. Precisão e calibração: A precisão dos sensores de umidade no solo pode variar entre diferentes modelos e marcas. Alguns sensores podem exigir calibração inicial para compensar variações individuais. A calibração geralmente envolve comparar as leituras do sensor com medições de umidade conhecidas e ajustar os valores conforme necessário.

3. Instalação e posicionamento: Os sensores de umidade devem ser instalados em uma profundidade onde possam fornecer uma leitura representativa da umidade do solo nas raízes das plantas. Isso geralmente significa enterrá-los a uma profundidade que corresponda à zona de enraizamento das plantas em questão. O posicionamento também deve garantir que o sensor não fique sujeito a interferências externas, como água corrente ou detritos.

4. Interferências e falsos positivos: Interferências podem incluir fatores como salinidade do solo, compactação do solo, presença de materiais não condutores próximos ao sensor, entre outros. Para minimizar interferências e falsas leituras, é importante realizar uma instalação adequada e calibrar o sensor conforme necessário. Além disso, a manutenção regular, como a limpeza dos sensores, pode ajudar a garantir a precisão das medições.

5. Aplicações e benefícios: Além da agricultura e jardinagem, os sensores de umidade no solo também são usados em áreas como monitoramento ambiental, pesquisa científica, controle de erosão, controle de pragas e conservação da água. Seus benefícios incluem a economia de água, a promoção do crescimento saudável das plantas, a prevenção de inundações e a minimização do desperdício de recursos hídricos em várias aplicações.



VOLTAR AO
SUMÁRIO



Projeto 09

Piano com buzzer

Etapa da Educação Básica:

Ensino Fundamental anos finais e Ensino médio

Público Alvo:

Alunos no 9º Ano, 1ª, 2ª ou 3ª Séries.

Modalidade:

() Sequência Didática. () Projeto () Iniciação Científica.

BNCC da Computação:

[\[EF69CO02\]](#) Elaborar algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.

[\[EM13CO13\]](#) Analisar e utilizar as diferentes formas de representação e consulta a dados em formato digital para pesquisas científicas.

[\[EM13CO15\]](#) Analisar a interação entre usuários e artefatos computacionais, abordando aspectos da experiência do usuário e promovendo reflexão sobre a qualidade do uso dos artefatos nas esferas do trabalho, do lazer e do estudo.

[\[EM13CO16\]](#) Desenvolver projetos com robótica, utilizando artefatos físicos ou simuladores.

[\[EM13CO18\]](#) Planejar e gerenciar projetos integrados às áreas de conhecimento de forma colaborativa, solucionando problemas, usando diversos artefatos computacionais.

Objeto(s) de conhecimento:

- Conhecer o dispositivo Buzzer
- Ligar o Buzzer e programar algumas notas musicais.



VOLTAR AO
SUMÁRIO



Vamos aos projetos!

Piano com buzzer



Objetivo da Aula:



- Reconhecer o dispositivo Buzzer.
- Reconhecer as aplicações do Buzzer .
- Entender como o dispositivo Buzzer funciona.

Contextualização



O projeto de um piano com buzzer oferece uma introdução prática à eletrônica e programação para iniciantes, permitindo que explorem conceitos fundamentais enquanto criam música. Além disso, proporciona uma maneira divertida e acessível de aprender sobre frequências sonoras, sequenciamento de tons e controle de dispositivos de saída, promovendo o aprendizado interativo e criativo.

Etapa



Orientações para o professor

Apresentando componentes da plataforma:

- ✓ Professor, você pode começar pedindo para os alunos separarem os componentes a serem utilizados.
- ✓ Comente sobre o dispositivo buzzer, cite algumas situações de onde ele pode ser utilizado.
- ✓ É importante destacar as aplicações desse sensor. Professor, recomendamos que você faça a leitura da a página 23.
- ✓ Leia os questionamentos no final desse projeto, para discutir com seus alunos.
- ✓ Mãos a obra!



VOLTAR AO SUMÁRIO



Etapa

2. Materiais Utilizados

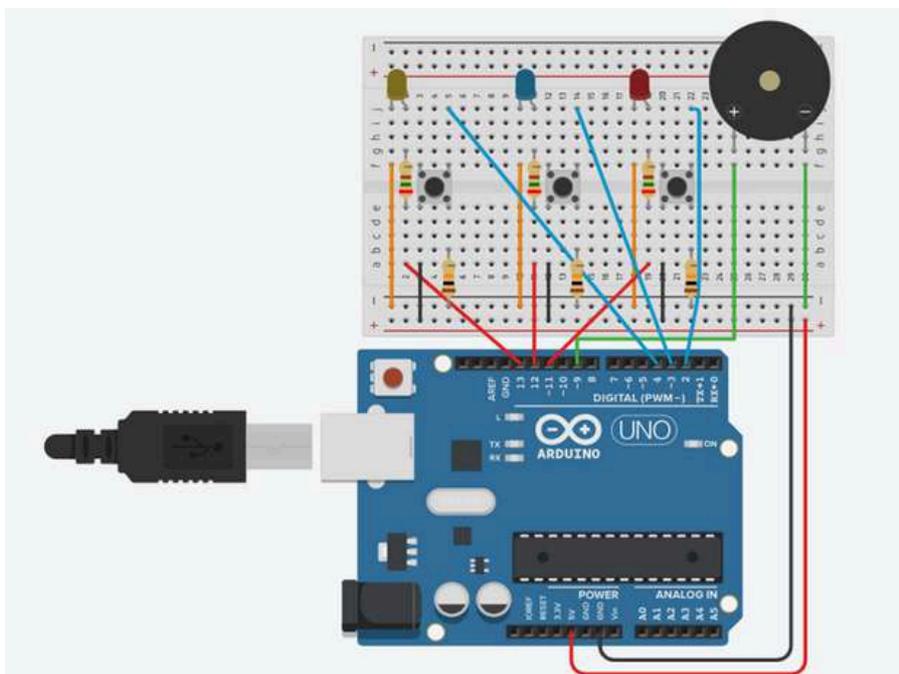
- 1 x PC
- 1 x Cabo USB
- 1 x Arduino Uno
- 1 x Protoboard
- 3 x LED
- 3 x Resistores 220 Ω para o LED
- 3 x Resistores 10k Ω para o Push buton
- 3 x Push buton
- 17 x Fios Jumpers
- 1 x Sensor de Umidade

Etapa

3. Metodologia

3.1 Ligando Componentes:

- Separe os materiais a serem utilizados;
- Tome cuidado ao ligar o dispositivo buzzer;
- Note que ele tem dois polos, positivo (+) e negativo (-);
- Tome cuidado ao ligar o dispositivo, o positivo nesse projeto será ligado na porta 9 digital do Arduino, e o negativo deve ser ligado no GND, da placa Arduino;
- Atenção se você inverter o fio, o dispositivo buzzer poderá queimar;
- Destaque para as conexões nos pinos de entrada/saída digitais do Arduino;
- Faça a ligação dos componetes conforme a figura a seguir.



**AUTODESK**
Tinkercad

Quer montar esse projeto virtualmente? Ele está disponível em:

<https://www.tinkercad.com/things/8f9NGDXEB6W-projeto-piano-com-buzzer?sharecode=I58YBrZLMIGCVW08XVMMW-8milfz6czx3WrUSvXOQOE>

Ou aponte o celular para o QR-code



VOLTAR AD SUMÁRIO

3.2 Programação:

- Agora é ligar o PC, para passar a programação para a placa do Arduino.
- Conecte a placa do Arduino ao PC, utilize o cabo USB.
- Esse momento é muito importante, escolha a porta adequada, na **aba Sketch**, clique em ferramentas, certifique-se que a porta escolhida é a adequada.
- Agora é passar o programa para a IDE, iremos fornecer o código de programação.
- Copie o programa abaixo e cole dentro da aba Sketch.
- Observe as constantes novas que surgiram na programação
- Aprenda a fazer a leitura e as alterações da programação, que é importante!

*

* **Projeto 09 – Projeto Piano com Buzzer**

*



*

*****/

```
const int ledpino1 = 13;
const int ledpino2 = 12;
const int ledpino3 = 11;
const int botaoA = 4;
const int botaoB = 3;
const int botaoC = 2;
const int som = 9;

int estado_botaoA = 0;
int estado_botaoB = 0;
int estado_botaoC = 0;
int tom = 0;

void setup() {
  // coloque seu código de setup aqui, para rodar uma vez: define saída e entrada do arduino
  pinMode(ledpino1, OUTPUT); //pino saida
  pinMode(ledpino2, OUTPUT); //pino saida
  pinMode(ledpino3, OUTPUT); //pino saida
  pinMode(som, OUTPUT); // pino de saída do arduino
  pinMode(botaoA, INPUT); //pino do botão é entrada
  pinMode(botaoB, INPUT); //pino do botão é entrada
  pinMode(botaoC, INPUT); //pino do botão é entrada
}

void loop() {
  // coloque seu código principal aqui, para rodar repetidamente:
```



VOLTAR AO
SUMÁRIO



3.2 Continuação da Programação:

```
estado_botaoA = digitalRead(botaoA); //define estado bt A
estado_botaoB = digitalRead(botaoB);
estado_botaoC = digitalRead(botaoC);

if(estado_botaoA && !estado_botaoB && !estado_botaoC) {
    tom = 264;
    digitalWrite(ledpino1, HIGH);
}
if(estado_botaoB && !estado_botaoA && !estado_botaoC) {
    tom = 330;
digitalWrite(ledpino2, HIGH);
}
if (estado_botaoC && !estado_botaoA && !estado_botaoB) {
    tom = 440;
    digitalWrite(ledpino3, HIGH);
}

if(tom > 0) { //enquanto Tom for maior que zero faça o que esta descrit o baixo:
    digitalWrite(som, HIGH); // Liga buzzer
    delayMicroseconds(tom); // Espera o tempo proporcional ao comprimento de onda da
nota musical em milisegundos
    digitalWrite(som, LOW); // Desliga buzzer
    delayMicroseconds(tom); // Espera o tempo proporcional ao comprimento de onda da
nota musical em milisegundos
    tom = 0; // Reseta o Tom para zero, para sair do loop while e nao tocar o som
constantemente
    digitalWrite(ledpino1, LOW);
    digitalWrite(ledpino2, LOW);
    digitalWrite(ledpino3, LOW);
}

} // SEDU-ES
```

Atenção!

□ código de programação tem que ser **copiado todo**.
Caso, falte alguma linha de código o projeto **não irá funcionar**.



VOLTAR AO
SUMÁRIO



- Atenção copie todo o código se esquecer qualquer caractere da programação, ele não irá funcionar.
- Na aba Sketch temos a opção de Verificar/Compilar que serve para ver se seu código está escrito corretamente, além disso tem um botão acima do espaço , que são respectivamente Verificar/Compilar e Carregar, onde o carregar serve para enviar o código para o Arduino.
 -  Verificar
 -  Compilar
- Agora é fazer o teste, pressione o botão push-button uma única vez, se der certo o LED irá acender, e o buzzer irá emitir uma nota musical

3.2 Lendo a Programação:

```
101010101
101011101
010101010
101010101
101011101
```

- **É importante destacar essa parte da programação:**
 - tom 264 Hz corresponde ao Dó.
 - tom 330 Hz corresponde ao Mi.
 - tom 440 Hz corresponde ao Lá.
 - Veja a tabela de notas musicais ao lado.
- ```

• if(estado_botaoA && !estado_botaoB && !estado_botaoC) {
• tom = 264;
• digitalWrite(ledpino1, HIGH);
• }
• if(estado_botaoB && !estado_botaoA && !estado_botaoC) {
• tom = 330;
• digitalWrite(ledpino2, HIGH);
• }
• if (estado_botaoC && !estado_botaoA && !estado_botaoB) {
• tom = 440;
• digitalWrite(ledpino3, HIGH);

```

| Notas | Frequência (Hz) |
|-------|-----------------|
| Dó    | 264             |
| Ré    | 300             |
| Mi    | 330             |
| Fá    | 352             |
| Sol   | 396             |
| Lá    | 440             |
| Si    | 495             |

Disponível em:  
<https://vestibulares.estrategia.com/public/questoes/Analise-tabela-que386f58f35/>

### 3.3 Alterando a Programação:



- Professor, faça as seguintes sugestões para seus estudantes:
- Peça para eles Trocarem as notas musicais mudando a frequência;
- Deixe que eles alterem a programação;
- Peça para eles fazerem as ligações, e alterarem a programação, de forma que o piano tenha as 7 notas musicais acima;
- Sugestão, peça que a inteligência artificial, chat-GPT faça a programação para o projeto;
- Caso você queira montar esse projeto, vai precisar de mais uma aula.



VOLTAR AO SUMÁRIO



## Etapa



### Sistematização

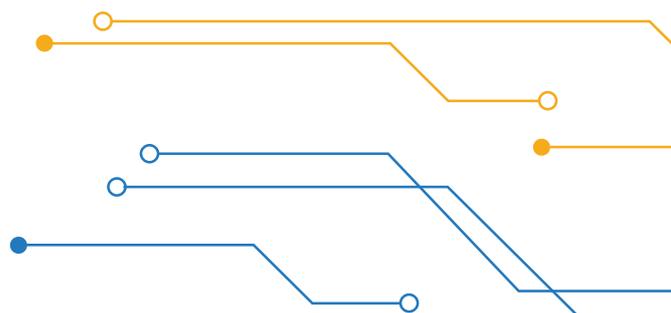
- Número de aulas para essa atividade: 01 aula.
- Montagem do projeto e discussão.
- Professor, divida seus estudantes em grupos de 3 pessoas ou duplas.
- As aulas devem priorizar a discussão, o debate para tomada de decisões, afim de resolver a situação problema.

## Etapa



### Questionamentos

- 1. Design do Piano com Buzzer:** Como o piano com buzzer é projetado e montado? Quais são os componentes principais necessários, além do buzzer? Como os diferentes tons musicais são produzidos utilizando-se apenas um buzzer?
- 2. Programação e Sequenciamento de Tons:** Como a programação é realizada para criar diferentes notas musicais no piano com buzzer? É utilizado algum algoritmo específico para mapear as notas para frequências sonoras? Como são organizados os sequenciamentos de tons para reproduzir músicas ou melodias?
- 3. Variação de Intensidade e Duração das Notas:** O piano com buzzer é capaz de reproduzir diferentes intensidades de som ou apenas notas de volume constante? É possível controlar a duração das notas, como notas curtas ou longas? Como essas variações são implementadas na programação?
- 4. Possíveis Aplicações e Utilidades:** Além de ser uma ferramenta educativa e de entretenimento, quais são as possíveis aplicações práticas de um piano com buzzer? Ele pode ser usado em sistemas de alerta sonoro, brinquedos interativos, demonstrações educacionais, entre outros contextos?
- 5. Desafios e Oportunidades de Aprendizado:** Quais são os desafios comuns enfrentados ao projetar e programar um piano com buzzer? Como esses desafios podem ser superados? Que oportunidades de aprendizado esse projeto oferece em termos de eletrônica, programação e música?



VOLTAR AO SUMÁRIO



## Respostas esperadas:



**1. Design do Piano com Buzzer:** O piano com buzzer pode ser projetado utilizando um Arduino ou microcontrolador semelhante, um buzzer piezoelétrico e botões ou teclas para cada nota musical. O buzzer é capaz de produzir diferentes tons musicais variando a frequência do sinal enviado a ele. As diferentes notas musicais são produzidas enviando-se sinais de diferentes frequências para o buzzer em resposta ao pressionar das teclas.

**2. Programação e Sequenciamento de Tons:** A programação para o piano com buzzer envolve mapear cada tecla do piano para uma frequência específica de nota musical. Isso pode ser feito atribuindo valores de frequência predefinidos a cada tecla ou usando uma fórmula matemática para calcular as frequências com base na posição da tecla. O sequenciamento de tons para reproduzir músicas ou melodias pode ser realizado armazenando sequências de frequências correspondentes às notas em uma matriz ou em uma estrutura de dados semelhante.

**3. Variação de Intensidade e Duração das Notas:** Geralmente, um buzzer piezoelétrico produzirá notas de volume constante. No entanto, é possível simular uma variação na intensidade do som pulsando rapidamente o sinal enviado ao buzzer, criando assim a ilusão de variação de volume. Quanto à duração das notas, ela pode ser controlada programaticamente, especificando o tempo que o sinal de cada nota é mantido ativo.

**4. Possíveis Aplicações e Utilidades:** Além de ser uma ferramenta educativa e de entretenimento, um piano com buzzer pode ser utilizado em projetos de automação residencial para alertas sonoros ou como parte de um sistema de música automatizado. Ele também pode ser incorporado em brinquedos educativos para ensinar conceitos musicais básicos para crianças.

**5. Desafios e Oportunidades de Aprendizado:** Os desafios ao projetar um piano com buzzer podem incluir a calibração das notas para garantir uma reprodução precisa, a programação de sequências musicais complexas e a gestão do tempo de execução para produzir notas de duração variável. No entanto, esses desafios oferecem oportunidades valiosas de aprendizado em eletrônica, programação e música, permitindo aos criadores desenvolver habilidades nessas áreas enquanto se divertem construindo e tocando o instrumento.



# Projeto 10

## Músicas no Arduino

### Etapa da Educação Básica:

Ensino Fundamental anos finais e Ensino médio

### Público Alvo:

Alunos no 9º Ano, 1ª, 2ª ou 3ª Séries.

### Modalidade:

(  ) Sequência Didática.      (  ) Projeto      (  ) Iniciação Científica.

### BNCC da Computação:

[\[EF69CO02\]](#) Elaborar algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.

[\[EM13CO13\]](#) Analisar e utilizar as diferentes formas de representação e consulta a dados em formato digital para pesquisas científicas.

[\[EM13CO15\]](#) Analisar a interação entre usuários e artefatos computacionais, abordando aspectos da experiência do usuário e promovendo reflexão sobre a qualidade do uso dos artefatos nas esferas do trabalho, do lazer e do estudo.

[\[EM13CO16\]](#) Desenvolver projetos com robótica, utilizando artefatos físicos ou simuladores.

[\[EM13CO18\]](#) Planejar e gerenciar projetos integrados às áreas de conhecimento de forma colaborativa, solucionando problemas, usando diversos artefatos computacionais.

### Objeto(s) de conhecimento:

- Reconhecer que podemos colocar músicas para o Buzzer tocar.
- Conhecer bibliotecas de músicas para a plataforma Arduino.



VOLTAR AO  
SUMÁRIO



# Vamos aos projetos!

## Músicas no Arduino



Disponível em:  
<https://jrmcoaching.com.br/blog/20-musicas-superacao-vaio-inspirar/>

### Objetivo da Aula:



- Reconhecer que podemos produzir músicas no Arduino.
- Reconhecer diversas programações para tocar músicas.
- Aplicar o componente curricular Arte, na plataforma Arduino.

### Contextualização



O projeto de músicas com Arduino é uma maneira divertida e educativa de explorar as capacidades sonoras do Arduino, ensinando conceitos de programação, eletrônica e áudio de uma forma prática e envolvente. Além disso, permite criar e personalizar melodias e efeitos sonoros para uma variedade de aplicações, desde brinquedos musicais até sistemas de sinalização sonora em projetos de automação residencial ou industrial, com músicas personalizadas.

### Etapa



### Orientações para o professor

#### Apresentando componentes da plataforma:

- ✓ Professor, você pode começar pedindo para os estudantes separarem os componentes a serem utilizados.
- ✓ Comente sobre o dispositivo buzzer, cite algumas situações de onde ele pode ser utilizado.
- ✓ É importante destacar como gerar músicas na plataforma Arduino. Professor, recomendamos que você faça a leitura da a página 24.
- ✓ Leia os questionamentos no final desse projeto, para discutir com seus estudantes.
- ✓ Agora é, mãos a obra!



VOLTAR AD  
SUMÁRIO



## Etapa

### 2. Materiais Utilizados

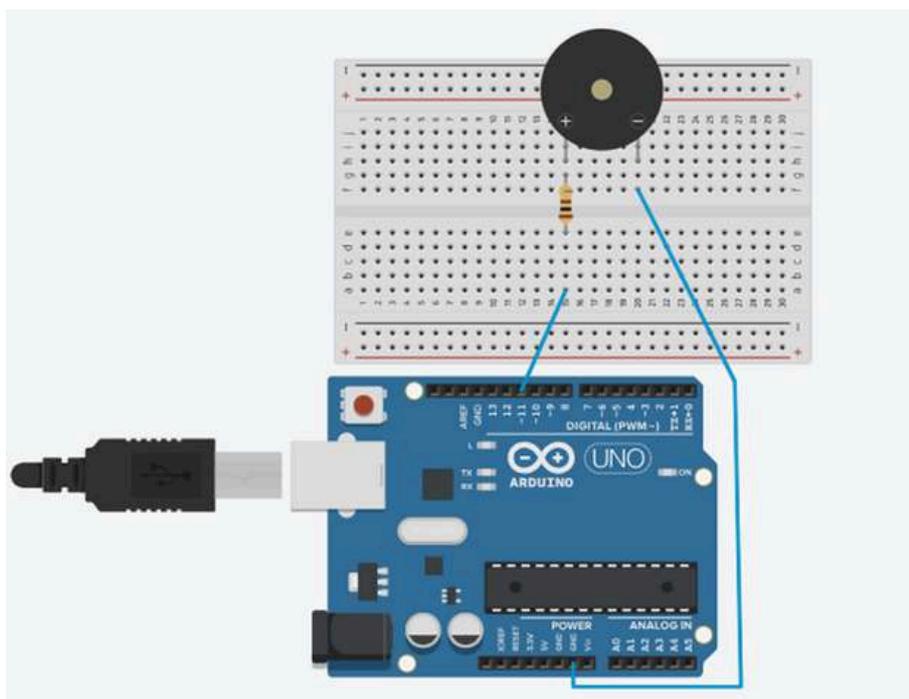
- 1 x PC
- 1 x Cabo USB
- 1 x Arduino Uno
- 1 x Protoboard
- 1 x Resistores 220  $\Omega$  para o buzzer
- 1 x Dispositivo buzzer

## Etapa

### 3. Metodologia

#### 3.1 Ligando Componentes:

- Separe os materiais a serem utilizados;
- Tome cuidado ao ligar o dispositivo buzzer;
- Note que ele tem dois polos, positivo (+) e negativo (-);
- Atenção se você inverter o fio, o dispositivo buzzer poderá queimar;
- Destaque para as conexões nos pinos de entrada/saída digitais do Arduino;
- Faça a ligação dos componentes conforme a figura a seguir.



**TINKER CAD** AUTODESK  
Tinkercad

Quer montar esse projeto virtualmente?  
Ele está disponível em:

[https://www.tinkercad.com/things/5u2n8W555Gz-musicas-no-arduino?sharecode=nRJWSGdQa\\_Nj8T5Gc5GfrLrYUyde3ReuyGm5MxZILLE](https://www.tinkercad.com/things/5u2n8W555Gz-musicas-no-arduino?sharecode=nRJWSGdQa_Nj8T5Gc5GfrLrYUyde3ReuyGm5MxZILLE)

Du aponte o celular para o QR-code



VOLTAR AO SUMÁRIO



## 3.2 Programação:

- Agora é ligar o PC, para passar a programação para a placa Arduino .
- Conecte a placa do Arduino ao PC, utilize o cabo USB.
- Esse momento é muito importante, escolha a porta adequada, na **aba Sketch**, clique em ferramentas, certifique-se que a porta escolhida é a adequada.
- Agora é passar o programa para a IDE, iremos fornecer o código de programação.
- Copie o programa abaixo e cole dentro da aba Sketch.
- Observe as constantes novas que surgiram na programação
- Aprenda a fazer a leitura e as alterações da programação, que é importante!

```
/*
*
* Projeto 10 – MÚSICAS NO ARDUINO
* 
*
*
*****/

/*
Super Mario Bros - Overworld theme
Connect a piezo buzzer or speaker to pin 11 or select a new pin.
*/

#define NOTE_B0 31
#define NOTE_C1 33
#define NOTE_CS1 35
#define NOTE_D1 37
#define NOTE_DS1 39
#define NOTE_E1 41
#define NOTE_F1 44
#define NOTE_FS1 46
#define NOTE_G1 49
#define NOTE_GS1 52
#define NOTE_A1 55
#define NOTE_AS1 58
#define NOTE_B1 62
#define NOTE_C2 65
#define NOTE_CS2 69
#define NOTE_D2 73
#define NOTE_DS2 78
#define NOTE_E2 82
```



VOLTAR AO  
SUMÁRIO



### 3.21 Continuação da Programação:

```
.#define NOTE_F2 87
#define NOTE_FS2 93
#define NOTE_G2 98
#define NOTE_GS2 104
#define NOTE_A2 110
#define NOTE_AS2 117
#define NOTE_B2 123
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
```

## 3.22 Continuação da Programação

```
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE_B7 3951
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE_D8 4699
#define NOTE_DS8 4978
#define REST 0
```



### 3.23 Continuação da Programação:

```
// change this to make the song slower or faster
int tempo = 200;
```

```
// change this to whichever pin you want to use
int buzzer = 11;
```

```
// notes of the melody followed by the duration.
// a 4 means a quarter note, 8 an eighteenth , 16 sixteenth, so on
// !!negative numbers are used to represent dotted notes,
// so -4 means a dotted quarter note, that is, a quarter plus an eighteenth!!
int melody[] = {
```

```
 // Super Mario Bros theme
 // Score available at https://musescore.com/user/2123/scores/2145
 // Theme by Koji Kondo
```

```
NOTE_E5,8, NOTE_E5,8, REST,8, NOTE_E5,8, REST,8, NOTE_C5,8, NOTE_E5,8, //1
NOTE_G5,4, REST,4, NOTE_G4,8, REST,4,
NOTE_C5,-4, NOTE_G4,8, REST,4, NOTE_E4,-4, // 3
NOTE_A4,4, NOTE_B4,4, NOTE_AS4,8, NOTE_A4,4,
NOTE_G4,-8, NOTE_E5,-8, NOTE_G5,-8, NOTE_A5,4, NOTE_F5,8, NOTE_G5,8,
REST,8, NOTE_E5,4,NOTE_C5,8, NOTE_D5,8, NOTE_B4,-4,
NOTE_C5,-4, NOTE_G4,8, REST,4, NOTE_E4,-4, // repeats from 3
NOTE_A4,4, NOTE_B4,4, NOTE_AS4,8, NOTE_A4,4,
NOTE_G4,-8, NOTE_E5,-8, NOTE_G5,-8, NOTE_A5,4, NOTE_F5,8, NOTE_G5,8,
REST,8, NOTE_E5,4,NOTE_C5,8, NOTE_D5,8, NOTE_B4,-4,
REST,4, NOTE_G5,8, NOTE_FS5,8, NOTE_F5,8, NOTE_DS5,4, NOTE_E5,8, //7
REST,8, NOTE_GS4,8, NOTE_A4,8, NOTE_C4,8, REST,8, NOTE_A4,8, NOTE_C5,8,
NOTE_D5,8,
REST,4, NOTE_DS5,4, REST,8, NOTE_D5,-4,
NOTE_C5,2, REST,2,
```

```
REST,4, NOTE_G5,8, NOTE_FS5,8, NOTE_F5,8, NOTE_DS5,4, NOTE_E5,8, //repeats
from 7
REST,8, NOTE_GS4,8, NOTE_A4,8, NOTE_C4,8, REST,8, NOTE_A4,8, NOTE_C5,8,
NOTE_D5,8,
REST,4, NOTE_DS5,4, REST,8, NOTE_D5,-4,
NOTE_C5,2, REST,2,
```



VOLTAR AO  
SUMÁRIO



### 3.24 Continuação da Programação

NOTE\_C5,8, NOTE\_C5,4, NOTE\_C5,8, REST,8, NOTE\_C5,8, NOTE\_D5,4, //11  
NOTE\_E5,8, NOTE\_C5,4, NOTE\_A4,8, NOTE\_G4,2,

NOTE\_C5,8, NOTE\_C5,4, NOTE\_C5,8, REST,8, NOTE\_C5,8, NOTE\_D5,8,  
NOTE\_E5,8, //13

REST,1,

NOTE\_C5,8, NOTE\_C5,4, NOTE\_C5,8, REST,8, NOTE\_C5,8, NOTE\_D5,4,

NOTE\_E5,8, NOTE\_C5,4, NOTE\_A4,8, NOTE\_G4,2,

NOTE\_E5,8, NOTE\_E5,8, REST,8, NOTE\_E5,8, REST,8, NOTE\_C5,8, NOTE\_E5,4,

NOTE\_G5,4, REST,4, NOTE\_G4,4, REST,4,

NOTE\_C5,-4, NOTE\_G4,8, REST,4, NOTE\_E4,-4, // 19

NOTE\_A4,4, NOTE\_B4,4, NOTE\_AS4,8, NOTE\_A4,4,

NOTE\_G4,-8, NOTE\_E5,-8, NOTE\_G5,-8, NOTE\_A5,4, NOTE\_F5,8, NOTE\_G5,8,

REST,8, NOTE\_E5,4, NOTE\_C5,8, NOTE\_D5,8, NOTE\_B4,-4,

NOTE\_C5,-4, NOTE\_G4,8, REST,4, NOTE\_E4,-4, // repeats from 19

NOTE\_A4,4, NOTE\_B4,4, NOTE\_AS4,8, NOTE\_A4,4,

NOTE\_G4,-8, NOTE\_E5,-8, NOTE\_G5,-8, NOTE\_A5,4, NOTE\_F5,8, NOTE\_G5,8,

REST,8, NOTE\_E5,4, NOTE\_C5,8, NOTE\_D5,8, NOTE\_B4,-4,

NOTE\_E5,8, NOTE\_C5,4, NOTE\_G4,8, REST,4, NOTE\_GS4,4, //23

NOTE\_A4,8, NOTE\_F5,4, NOTE\_F5,8, NOTE\_A4,2,

NOTE\_D5,-8, NOTE\_A5,-8, NOTE\_A5,-8, NOTE\_A5,-8, NOTE\_G5,-8, NOTE\_F5,-8,

NOTE\_E5,8, NOTE\_C5,4, NOTE\_A4,8, NOTE\_G4,2, //26

NOTE\_E5,8, NOTE\_C5,4, NOTE\_G4,8, REST,4, NOTE\_GS4,4,

NOTE\_A4,8, NOTE\_F5,4, NOTE\_F5,8, NOTE\_A4,2,

NOTE\_B4,8, NOTE\_F5,4, NOTE\_F5,8, NOTE\_F5,-8, NOTE\_E5,-8, NOTE\_D5,-8,

NOTE\_C5,8, NOTE\_E4,4, NOTE\_E4,8, NOTE\_C4,2,

NOTE\_E5,8, NOTE\_C5,4, NOTE\_G4,8, REST,4, NOTE\_GS4,4, //repeats from 23

NOTE\_A4,8, NOTE\_F5,4, NOTE\_F5,8, NOTE\_A4,2,

NOTE\_D5,-8, NOTE\_A5,-8, NOTE\_A5,-8, NOTE\_A5,-8, NOTE\_G5,-8, NOTE\_F5,-8,

NOTE\_E5,8, NOTE\_C5,4, NOTE\_A4,8, NOTE\_G4,2, //26

NOTE\_E5,8, NOTE\_C5,4, NOTE\_G4,8, REST,4, NOTE\_GS4,4,

NOTE\_A4,8, NOTE\_F5,4, NOTE\_F5,8, NOTE\_A4,2,

NOTE\_B4,8, NOTE\_F5,4, NOTE\_F5,8, NOTE\_F5,-8, NOTE\_E5,-8, NOTE\_D5,-8,

NOTE\_C5,8, NOTE\_E4,4, NOTE\_E4,8, NOTE\_C4,2,

NOTE\_C5,8, NOTE\_C5,4, NOTE\_C5,8, REST,8, NOTE\_C5,8, NOTE\_D5,8, NOTE\_E5,8,

REST,1,



VOLTAR AO  
SUMÁRIO



### 3.25 Continuação da Programação

```
NOTE_C5,8, NOTE_C5,4, NOTE_C5,8, REST,8, NOTE_C5,8, NOTE_D5,4, //33
NOTE_E5,8, NOTE_C5,4, NOTE_A4,8, NOTE_G4,2,
NOTE_E5,8, NOTE_E5,8, REST,8, NOTE_E5,8, REST,8, NOTE_C5,8, NOTE_E5,4,
NOTE_G5,4, REST,4, NOTE_G4,4, REST,4,
NOTE_E5,8, NOTE_C5,4, NOTE_G4,8, REST,4, NOTE_GS4,4,
NOTE_A4,8, NOTE_F5,4, NOTE_F5,8, NOTE_A4,2,
NOTE_D5,-8, NOTE_A5,-8, NOTE_A5,-8, NOTE_A5,-8, NOTE_G5,-8, NOTE_F5,-8,
```

```
NOTE_E5,8, NOTE_C5,4, NOTE_A4,8, NOTE_G4,2, //40
NOTE_E5,8, NOTE_C5,4, NOTE_G4,8, REST,4, NOTE_GS4,4,
NOTE_A4,8, NOTE_F5,4, NOTE_F5,8, NOTE_A4,2,
NOTE_B4,8, NOTE_F5,4, NOTE_F5,8, NOTE_F5,-8, NOTE_E5,-8, NOTE_D5,-8,
NOTE_C5,8, NOTE_E4,4, NOTE_E4,8, NOTE_C4,2,
```

```
//game over sound
```

```
NOTE_C5,-4, NOTE_G4,-4, NOTE_E4,4, //45
NOTE_A4,-8, NOTE_B4,-8, NOTE_A4,-8, NOTE_GS4,-8, NOTE_AS4,-8, NOTE_GS4,-8,
NOTE_G4,8, NOTE_D4,8, NOTE_E4,-2,
```

```
};
```

```
// sizeof gives the number of bytes, each int value is composed of two bytes (16 bits)
// there are two values per note (pitch and duration), so for each note there are four bytes
int notes = sizeof(melody) / sizeof(melody[0]) / 2;
```

```
// this calculates the duration of a whole note in ms
int wholenote = (60000 * 4) / tempo;
```

```
int divider = 0, noteDuration = 0;
```

```
void setup() {
```

```
 // iterate over the notes of the melody.
```

```
 // Remember, the array is twice the number of notes (notes + durations)
```

```
 for (int thisNote = 0; thisNote < notes * 2; thisNote = thisNote + 2) {
```



VOLTAR AO  
SUMÁRIO



### 3.26 Continuação da Programação:

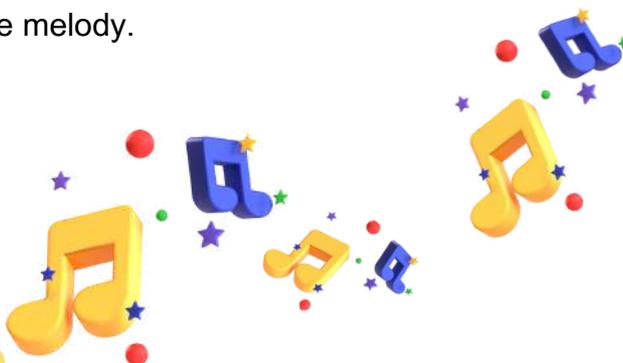
```
// calculates the duration of each note
divider = melody[thisNote + 1];
if (divider > 0) {
 // regular note, just proceed
 noteDuration = (wholenote) / divider;
} else if (divider < 0) {
 // dotted notes are represented with negative durations!!
 noteDuration = (wholenote) / abs(divider);
 noteDuration *= 1.5; // increases the duration in half for dotted notes
}

// we only play the note for 90% of the duration, leaving 10% as a pause
tone(buzzer, melody[thisNote], noteDuration * 0.9);

// Wait for the specief duration before playing the next note.
delay(noteDuration);

// stop the waveform generation before the next note.
noTone(buzzer);
}
}

void loop() {
 // no need to repeat the melody.
}
```



#### Programação de várias músicas:

- No SITE abaixo temos sugestões de diversas músicas.
- Disponível em: <https://dragaosemchama.com/2019/02/musicas-para-arduino/>



VOLTAR AO  
SUMÁRIO



- Atenção copie todo o código se esquecer qualquer caractere da programação, ele não irá funcionar.
- Na aba Sketch temos a opção de Verificar/Compilar que serve para ver se seu código está escrito corretamente, além disso tem um botão acima do espaço , que são respectivamente Verificar/Compilar e Carregar, onde o carregar serve para enviar o código para o Arduino.
  -  Verificar
  -  Compilar
- Agora é fazer o teste, se der certo o buzzer irá executar uma música.

### 3.2 Lendo a Programação:

```
101010101
101011101
010101010
101010101
101011101
```

- Nessa seção não iremos analisar a programação, já que as músicas vem montadas.
- O objetivo dessa atividade é montar o projeto e ver que é possível tocar músicas utilizando a plataforma Arduino.
- Professores de física, Arte e Música podem explorar bem esse projeto.
- Espero que vocês tenham gostado da atividade.

### 3.3 Alterando a Programação:



- Professor, faça as seguintes sugestões para seus estudantes:
- Peça para eles trocarem a músicas entrando no Site: <https://dragaosemchama.com/2019/02/musicas-para-arduino/>.
- Deixe que eles alterem a programação;
- Peça para eles escolherem músicas do Site de forma aleatória.
- No final da atividade peça para cada grupo apresentar uma música.
- É bem legal eles tentarem adivinhar, qual é a música.



VOLTAR AO  
SUMÁRIO

## Etapa

### Sistematização

- Número de aulas para essa atividade: 01 aula.
- Montagem do projeto e discussão.
- Professor, divida seus estudantes em grupos de 3 pessoas ou duplas.
- As aulas devem priorizar a discussão, o debate para tomada de decisões, afim de resolver a situação problema.

## Etapa

### Questionamentos

**1. Viabilidade técnica:** Quais são os componentes necessários para construir um reproduzidor de música com Arduino? Como o Arduino pode ser programado para reproduzir músicas ou sons? Quais são as limitações técnicas a serem consideradas, como armazenamento de áudio, qualidade sonora, entre outros?

**2. Aspectos criativos:** Como as melodias ou músicas podem ser criadas ou escolhidas para o reproduzidor de música? Existem bibliotecas ou recursos disponíveis para facilitar a criação ou reprodução de músicas? Como o Arduino pode ser usado para criar efeitos sonoros ou personalizar as músicas?

**3. Aplicações práticas:** Além de ser uma atividade educativa e divertida, quais são as possíveis aplicações práticas de um reproduzidor de música com Arduino? Ele pode ser integrado a outros projetos eletrônicos ou sistemas? Quais são os benefícios de adicionar funcionalidades sonoras a projetos de automação ou dispositivos eletrônicos?

**4. Desafios e soluções:** Quais são os principais desafios técnicos ou criativos ao desenvolver um reproduzidor de música com Arduino? Como esses desafios podem ser superados? Existem recursos, tutoriais ou comunidades online que podem oferecer suporte e orientação durante o processo de desenvolvimento?

**5. Aprendizado e potencial educativo:** Como a construção e programação de um reproduzidor de música com Arduino podem promover o aprendizado em áreas como programação, eletrônica e áudio? Ele pode ser usado como uma ferramenta educativa em ambientes de ensino formal ou informal? Como pode ser adaptado para diferentes faixas etárias ou níveis de habilidade?



VOLTAR AO  
SUMÁRIO

## Respostas esperadas:



**1. Viabilidade técnica:** As questões relacionadas à viabilidade técnica do projeto podem incluir:

**Componentes necessários:** Os componentes básicos incluem um Arduino, um módulo de armazenamento de áudio (como um módulo MP3 ou um cartão SD com um leitor), um amplificador de áudio e um alto-falante.

- **Programação do Arduino:** O Arduino pode ser programado para reproduzir áudio usando bibliotecas específicas que suportam a reprodução de arquivos de áudio armazenados no módulo de armazenamento.
- **Limitações técnicas:** As limitações incluem o tamanho do armazenamento disponível no módulo de áudio, a qualidade do áudio reproduzido, a capacidade de processamento do Arduino e a alimentação de energia para o amplificador e alto-falante.

**2. Aspectos criativos:** As respostas para questões sobre aspectos criativos podem incluir:

**Criação de músicas:** As músicas podem ser criadas manualmente usando uma linguagem de programação específica para o módulo de áudio ou podem ser pré-gravadas e armazenadas no módulo de armazenamento.

- **Recursos disponíveis:** Existem bibliotecas como a DFPlayerMini que facilitam a reprodução de músicas em dispositivos Arduino.
- **Personalização de músicas:** O Arduino pode ser programado para reproduzir efeitos sonoros ou ajustar parâmetros como volume, tonalidade e ritmo das músicas reproduzidas.

**3. Aplicações práticas:** As respostas para questões sobre aplicações práticas podem incluir:

**Integração com outros projetos:** O reprodutor de música com Arduino pode ser integrado a projetos de automação residencial para fornecer feedback sonoro ou para entretenimento.

**Benefícios:** Adicionar funcionalidades sonoras a projetos eletrônicos pode aumentar a interatividade e a experiência do usuário, além de fornecer feedback ou alertas sonoros em sistemas automatizados.

**4. Desafios e soluções:** As respostas para questões sobre desafios e soluções podem incluir:

**Desafios técnicos:** Desafios podem incluir a configuração correta do hardware, a programação para reprodução de áudio e a resolução de problemas de compatibilidade entre diferentes componentes.

**Recursos de suporte:** Existem recursos online, como fóruns de discussão, tutoriais e documentações de bibliotecas, que podem ajudar a resolver desafios técnicos.

**Aprendizado e potencial educativo:** As respostas para questões sobre aprendizado e potencial educativo podem incluir:

**5. Promoção do aprendizado:** A construção e programação do reprodutor de música com Arduino podem promover o aprendizado em áreas como programação, eletrônica e áudio, proporcionando uma experiência prática e hands-on.

**Ferramenta educativa:** O projeto pode ser adaptado para uso em ambientes educacionais, fornecendo uma introdução prática à programação, eletrônica e áudio para estudantes de todas as idades.



VOLTAR AO  
SUMÁRIO



# Projeto 11

## Usando a IA como programador

### Etapa da Educação Básica:

Ensino Fundamental anos finais e Ensino médio

### Público Alvo:

Alunos no 9º Ano, 1ª, 2ª ou 3ª Séries.

### Modalidade:

(  ) Sequência Didática. (  ) Projeto (  ) Iniciação Científica.

### BNCC da Computação:

(EF05CO04) Criar e simular algoritmos representados em linguagem oral, escrita ou pictográfica, que incluam sequências, repetições e seleções condicionais para resolver problemas de forma independente e em colaboração.

(EF15CO08) Reconhecer e utilizar tecnologias computacionais para pesquisar e acessar informações, expressar-se crítica e criativamente e resolver problemas.

(EF69CO02) Elaborar algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.

(EM13CO13) Analisar e utilizar as diferentes formas de representação e consulta a dados em formato digital para pesquisas científicas.

(EM13CO15) Analisar a interação entre usuários e artefatos computacionais, abordando aspectos da experiência do usuário e promovendo reflexão sobre a qualidade do uso dos artefatos nas esferas do trabalho, do lazer e do estudo.

(EM13CO16) Desenvolver projetos com robótica, utilizando artefatos físicos ou simuladores.

(EM13CO18) Planejar e gerenciar projetos integrados às áreas de conhecimento de forma colaborativa, solucionando problemas, usando diversos artefatos computacionais.

(EM13CO10) Conhecer os fundamentos da Inteligência Artificial, comparando-a com a inteligência humana, analisando suas potencialidades, riscos e limites.

### Objeto(s) de conhecimento:

- Reconhecer que podemos utilizar as Inteligências Artificiais para fazer os scripts de programação.
- Utilizar a programação da IA, para a resolução de um problema.



VOLTAR AO  
SUMÁRIO



# Vamos aos projetos!

## Usando IA como programador



### Objetivo da Aula:



- Utilizar o ChatGPT, como programador;
- Fazer prompt para programação;
- Aplicar o propt para criação do programa, de um cruzamento sinalizado, onde dois conjuntos de semáforos, operam de maneira coordenada.

### Contextualização



Controlar um cruzamento com dois conjuntos de semáforos operando de maneira coordenada é essencial para garantir a segurança e fluidez do tráfego. Nesse contexto, um programa Arduino pode ser desenvolvido para gerenciar a alternância de sinais luminosos, concedendo o direito de passagem aos veículos em diferentes direções em intervalos específicos. Isso envolve a sincronização dos semáforos para evitar colisões e congestionamentos, priorizando fluxos de tráfego mais pesados e garantindo tempos adequados de espera. Ao utilizar sensores de presença ou temporizadores, o programa pode adaptar dinamicamente os tempos de sinalização de acordo com as condições de tráfego em tempo real, contribuindo para a eficiência e segurança do cruzamento.

### Etapa



### Orientações para o professor

#### Apresentando componentes da plataforma:

- ✓ Professor você pode começar pedindo para os estudantes separarem os componentes a serem utilizados.
- ✓ Comente sobre o uso das IAs, explique como é utilizado o ChatGPT.
- ✓ Explique como pedir os Prompts para a IA (Inteligência Artificial).
- ✓ Leia os questionamentos no final desse projeto, para discutir com seus alunos.
- ✓ Mãos a obra!



VOLTAR AO SUMÁRIO



## Etapa

### 2. Materiais Utilizados

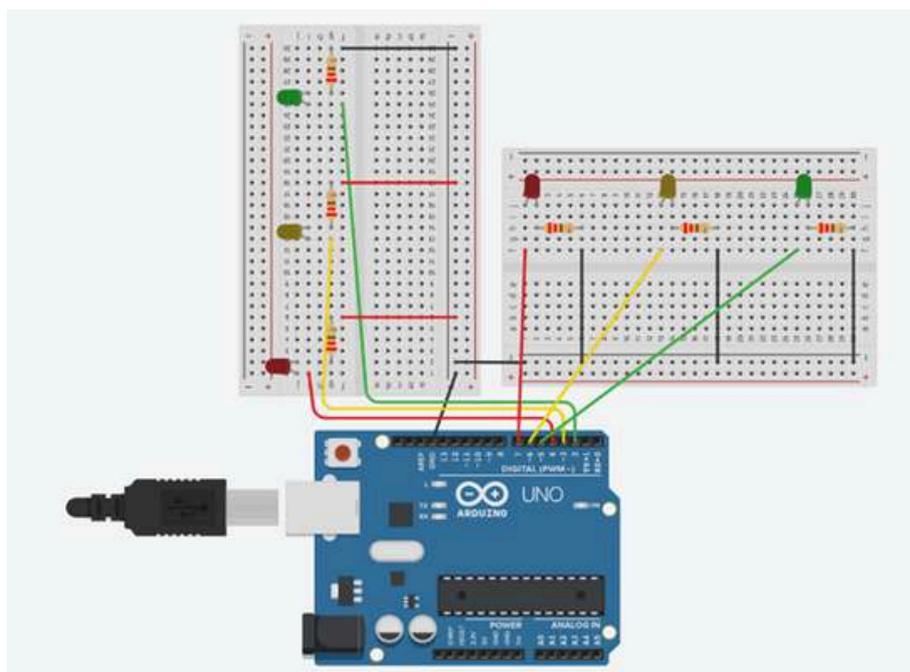
- 1 x PC
- 1 x Cabo USB
- 1 x Arduino Uno
- 1 x Protoboard
- 6 x LED
- 6 x Resistores 220  $\Omega$  para o LED
- 6 x Fios Jumpers

## Etapa

### 3. Metodologia

#### 3.1 Ligando Componentes:

- Separe os materiais a serem utilizados;
- Tome cuidado ao ligar os LEDs;
- Lembre-se que a perna maior deve ser ligado nas portas digitais da placa e a perna menor no GND.
- Faça a ligação dos componentes conforme a figura a seguir.



**TINKERCAD** AUTODESK  
Tinkercad

Quer montar esse projeto virtualmente? Ele está disponível em:

<https://www.tinkercad.com/things/gA1K2GD7Con-usando-ia-como-programador/editel?returnTo=%2Fdashboard%3Ftype%3Dcircuits%26collection%3Ddesigns>

Ou aponte o celular para o QR-code



VOLTAR AO SUMÁRIO

## 3.2 Continuação da Programação:

```
/*
 *
 *Projeto 11 – Usando IA como programador
 * SEDU-ES
 * GECEB
 *
 */

// Definição dos pinos para os LEDs do primeiro conjunto de semáforos (vertical)
const int vermelhoVertical = 2;
const int amareloVertical = 3;
const int verdeVertical = 4;

// Definição dos pinos para os LEDs do segundo conjunto de semáforos (horizontal)
const int vermelhoHorizontal = 5;
const int amareloHorizontal = 6;
const int verdeHorizontal = 7;

// Definição dos tempos de cada fase do semáforo (em milissegundos)
const int tempoVermelho = 5000; // 5 segundos
const int tempoAmarelo = 2000; // 2 segundos
const int tempoVerde = 5000; // 5 segundos

void setup() {
 // Configuração dos pinos dos LEDs como saída
 pinMode(vermelhoVertical, OUTPUT);
 pinMode(amareloVertical, OUTPUT);
 pinMode(verdeVertical, OUTPUT);
 pinMode(vermelhoHorizontal, OUTPUT);
 pinMode(amareloHorizontal, OUTPUT);
 pinMode(verdeHorizontal, OUTPUT);
}

void loop() {
 // Sinal de trânsito vertical
 digitalWrite(vermelhoVertical, HIGH);
 digitalWrite(amareloVertical, LOW);
 digitalWrite(verdeVertical, LOW);
}
```



VOLTAR AO  
SUMÁRIO



## 3.2 Continuação da Programação:

```
// Sinal de trânsito horizontal
digitalWrite(vermelhoHorizontal, LOW);
digitalWrite(amareloHorizontal, LOW);
digitalWrite(verdeHorizontal, HIGH);

delay(tempoVermelho);

// Sinal de trânsito vertical
digitalWrite(vermelhoVertical, LOW);
digitalWrite(amareloVertical, HIGH);
digitalWrite(verdeVertical, LOW);

// Sinal de trânsito horizontal
digitalWrite(vermelhoHorizontal, LOW);
digitalWrite(amareloHorizontal, HIGH);
digitalWrite(verdeHorizontal, LOW);

delay(tempoAmarelo);

// Sinal de trânsito vertical
digitalWrite(vermelhoVertical, LOW);
digitalWrite(amareloVertical, LOW);
digitalWrite(verdeVertical, HIGH);

// Sinal de trânsito horizontal
digitalWrite(vermelhoHorizontal, HIGH);
digitalWrite(amareloHorizontal, LOW);
digitalWrite(verdeHorizontal, LOW);

delay(tempoVerde);
}
```

### Atenção!

□ código de programação tem que ser **copiado todo**.  
Caso, falte alguma linha de código o projeto **não irá funcionar**.



VOLTAR AO  
SUMÁRIO



- Atenção copie todo o código se esquecer qualquer caractere da programação, ele não irá funcionar.
- Na aba Sketch temos a opção de Verificar/Compilar que serve para ver se seu código está escrito corretamente, além disso tem um botão acima do espaço , que são respectivamente Verificar/Compilar e Carregar, onde o carregar serve para enviar o código para o Arduino.

 Verificar

 Compilar

- Agora é fazer o teste, faça a compilação, deixe o programa rodar e veja se o semáforo vertical e o horizontal trabalham de forma sincronizada.

```
101010101
101011101
010101010
101010101
101011101
```

### 3.2 Lendo a Programação:

- **É importante destacar essa parte da programação:**
- **// Definição dos pinos para os LEDs do primeiro conjunto de semáforos (vertical)**
- const int vermelhoVertical = 2;
- const int amareloVertical = 3;
- const int verdeVertical = 4;
- 
- **// Definição dos pinos para os LEDs do segundo conjunto de semáforos (horizontal)**
- const int vermelhoHorizontal = 5;
- const int amareloHorizontal = 6;
- const int verdeHorizontal = 7;
- 
- **// Definição dos tempos de cada fase do semáforo (em milissegundos)**
- const int tempoVermelho = 5000; // 5 segundos
- const int tempoAmarelo = 2000; // 2 segundos
- const int tempoVerde = 5000; // 5 segundos

### 3.3 Alterando a Programação:



- Professor, faça as seguintes sugestões para seus estudantes:
- Peça para eles criarem o prompt no ChatGPT;
- Deixe que os estudantes alterem os prompts e testem a programação;
- Peça para os estudantes fazerem as ligações, e alterarem a programação, até o semáforo ficar correto
- Sugestão, peça que a inteligência artificial, chat-GPT faça a programação para o projeto;
- Use para a programação acima os seguintes comandos:
- Prompt:/ **Você vai atuar como se fosse um programador.**
- Prompt:/ **Faça a programação de um cruzamento onde dois conjuntos de semáforos operam de maneira coordenada, para a plataforma Arduino.**



VOLTAR AO  
SUMÁRIO



## Etapa

### Sistematização

- Número de aulas para essa atividade: 01 aula.
- Montagem do projeto e discussão.
- Professor, divida seus estudantes em grupos de 3 pessoas ou duplas.
- As aulas devem priorizar a discussão, o debate para tomada de decisões, afim de resolver a situação problema.

## Etapa

### Questionamentos

1. Como as inteligências artificiais (IAs) estão sendo utilizadas atualmente na realização de programações?
2. Quais são os benefícios e desafios associados ao uso de IAs na realização de programações?
3. Até que ponto as IAs podem automatizar tarefas de programação complexas e criativas?
4. Quais são as principais áreas da programação que podem se beneficiar do uso de IAs?
5. Como as IAs podem auxiliar no desenvolvimento ágil de software e na detecção de bugs?
6. Qual é o papel dos programadores humanos no processo de utilização de IAs para realização de programações?
7. Quais são as implicações éticas e sociais do uso generalizado de IAs na programação?
8. Como as IAs estão impactando a forma como os programadores aprendem e praticam programação?
9. Quais são as perspectivas futuras para o uso de IAs na realização de programações e no campo da programação em geral?
10. Quais são as limitações atuais das IAs na realização de programações e como essas limitações podem ser superadas no futuro?



VOLTAR AO  
SUMÁRIO

## Respostas esperadas:



1. As inteligências artificiais (IAs) estão sendo utilizadas de diversas formas na realização de programações. Isso inclui desde a automação de tarefas simples, como a geração de código repetitivo, até a criação de sistemas mais complexos capazes de escrever códigos inteiros com base em requisitos específicos.
2. Os benefícios do uso de IAs na realização de programações incluem a automação de tarefas tediosas e repetitivas, o aumento da produtividade dos desenvolvedores, a identificação e correção de erros mais rapidamente, e a capacidade de explorar soluções criativas e inovadoras. No entanto, há desafios relacionados à qualidade e confiabilidade dos códigos gerados, à necessidade de supervisão humana para garantir que as soluções atendam aos requisitos do projeto e às questões éticas e sociais associadas ao uso generalizado de IAs na programação.
3. As IAs podem automatizar tarefas de programação complexas em certa medida, especialmente aquelas que envolvem análise de dados, reconhecimento de padrões e tomada de decisões baseadas em regras. No entanto, tarefas que requerem criatividade, compreensão contextual e raciocínio abstrato ainda são desafiadoras para as IAs replicarem com precisão.
4. As principais áreas da programação que podem se beneficiar do uso de IAs incluem a geração automática de código, a otimização de algoritmos, o teste de software, a depuração de erros, a previsão de falhas e a automação de processos de desenvolvimento.
5. As IAs podem auxiliar no desenvolvimento ágil de software ao automatizar tarefas como a geração de testes automatizados, a implementação contínua, o monitoramento de desempenho e a coleta de feedback do usuário. Além disso, elas podem ajudar na detecção de bugs ao analisar grandes volumes de código e identificar padrões de erros comuns.
6. O papel dos programadores humanos no processo de utilização de IAs para realização de programações é crucial. Embora as IAs possam automatizar muitas tarefas, os programadores ainda são necessários para definir os objetivos do projeto, supervisionar as operações das IAs, garantir a qualidade e segurança do código gerado e realizar tarefas que exigem julgamento humano e criatividade.
7. As implicações éticas e sociais do uso generalizado de IAs na programação incluem preocupações relacionadas à privacidade, viés algorítmico, substituição de empregos, desigualdade digital e dependência excessiva de tecnologia. É importante abordar essas questões de forma ética e responsável ao desenvolver e implementar soluções baseadas em IA.
8. As IAs estão impactando a forma como os programadores aprendem e praticam programação ao oferecer ferramentas de desenvolvimento mais inteligentes, acesso a recursos educacionais personalizados, feedback automatizado sobre o código escrito e oportunidades de colaboração em projetos de IA.
9. As perspectivas futuras para o uso de IAs na realização de programações são promissoras, com previsões de avanços significativos em áreas como geração automática de código, assistência inteligente ao desenvolvedor, automação de processos de desenvolvimento de software e melhoria da eficiência e qualidade dos sistemas de IA.
10. As limitações atuais das IAs na realização de programações incluem dificuldades em lidar com a ambiguidade e a incerteza, compreender contextos complexos, aprender com dados limitados e adaptar-se a mudanças inesperadas no ambiente de desenvolvimento. Essas limitações podem ser superadas com o avanço da pesquisa em IA, o desenvolvimento de algoritmos mais sofisticados e a colaboração entre humanos e máquinas.



VOLTAR AO  
SUMÁRIO



## Referências Bibliográficas

Revista Tecnológica da FATEC-PR Curitiba, v.1, n. 5, Jan/Dez 2014 – ISSN 2179-3778. Disponível em: <https://www.unifatecpr.com.br/conteudo/Revista%20Tecnologica%20da%20FATEC-%20PR%20V%201%20N%205-%20JAN-DEZ%202014.pdf>. Acessado em Fevereiro de 2024.

BRASIL. Ministério da Educação. Base Nacional Comum Curricular. Brasília: MEC, 2018. Disponível em: [https://www.gov.br/mec/pt-br/escola-em-tempo-integral/BNCC\\_EI\\_EF\\_110518\\_versaofinal.pdf](https://www.gov.br/mec/pt-br/escola-em-tempo-integral/BNCC_EI_EF_110518_versaofinal.pdf). Acessado em: Fevereiro de 2024.

BRASIL. Ministério da Educação. **BNCC-COMPUTAÇÃO**. Brasília: MEC, 2018. Disponível em: <http://portal.mec.gov.br/docman/fevereiro-2022-pdf/236791-anexo-ao-parecer-cneceb-n-2-2022-bncc-computacao/file>. Acessado em: maio de 2024.

BRASIL. Ministério da Educação. **RESOLUÇÃO Nº 1, DE 4 DE OUTUBRO DE 2022**. Brasília: MEC, 2020. Disponível em: <http://portal.mec.gov.br/docman/outubro-2022-pdf/241671-rceb001-22/file>. Acessado em: maio de 2024

BLIKSTEIN, P.; CAMPOS, F. R. **Inovações Radicais na Educação Brasileira**. Porto Alegre: Penso, 2019.

MANFREDI, S. M. **Metodologia do ensino: diferentes concepções**. Campinas-SP: F.E./UNICAMP, mimeo, 1993.

CAMPOS, Flavio Rodrigues. Robótica educacional no Brasil: questões em aberto, desafios e perspectivas futuras. **Revista Ibero-Americana de Estudos em Educação**, Araraquara, v. 12, n. 4, p. 2108-2121, out./dez. 2017. Disponível em: . E-ISSN: 1982-5587. Acessado em: abril de 2024.



VOLTAR AO  
SUMÁRIO

