

Automação com Arduino

Acendendo uma Lâmpada com sensor de presença

Arthur Caio Vargas e Pinto

MANUAL DO PROFESSOR



Arthur Caio Vargas e Pinto

Automação com Arduino
Acendendo uma Lâmpada com sensor de presença
1ª Edição

Belo Horizonte
Instituto Federal de Minas Gerais
2022

© 2022 by Instituto Federal de Minas Gerais

Todos os direitos autorais reservados. Nenhuma parte desta publicação poderá ser reproduzida ou transmitida de qualquer modo ou por qualquer outro meio, eletrônico ou mecânico. Incluindo fotocópia, gravação ou qualquer outro tipo de sistema de armazenamento e transmissão de informação, sem prévia autorização por escrito do Instituto Federal de Minas Gerais.

Pró-reitor de Pesquisa	Fernando Gomes Braga
Coordenador de Inovação	Norimar de Melo Verticchio
Autor do curso	Arthur Caio Vargas e Pinto
Arte gráfica	Ângela Bacon
Diagramação	Eduardo dos Santos Oliveira

FICHA CATALOGRÁFICA
Dados Internacionais de Catalogação na Publicação (CIP)
Índice para catálogo sistemático:

1. Não preencher

Não preencher

2022

Direitos exclusivos cedidos à
Instituto Federal de Minas Gerais
Avenida Mário Werneck, 2590,
CEP: 30575-180, Buritis, Belo Horizonte – MG,
Telefone: (31) 2513-5157

Sobre o material

Este curso é autoexplicativo e não possui tutoria. O material didático, incluindo suas videoaulas, foi projetado para que você consiga desenvolver e ministrar esse curso de forma autônoma e suficiente.

Caso opte por imprimir este *e-book*, você não perderá a possibilidade de acessar os materiais multimídia e complementares. Os *links* podem ser acessados usando o seu celular, por meio do glossário de Códigos QR disponível no fim deste manual.

Embora o material passe por revisão, somos gratos em receber suas sugestões para possíveis correções (erros ortográficos, conceituais, *links* inativos etc.). A sua participação é muito importante para a nossa constante melhoria. Para relatar qualquer erro ou sugestão de melhoria, envie um e-mail para nit@ifmg.edu.br, com o seguinte assunto: MELHORIA NO MATERIAL DO LIC

Para saber mais sobre a Inovação do IFMG acesse

www.ifmg.edu.br

Apresentação dos Ícones

Os ícones são elementos gráficos para facilitar os estudos, fique atento quando eles aparecem no texto. Veja aqui o seu significado:



Atenção: indica pontos de maior importância na atividade e que exige mais atenção do aluno e instrutor.



Dica do professor: novas informações ou curiosidades relacionadas ao tema em estudo.



Mídia digital: sugestão de recursos audiovisuais para enriquecer a aprendizagem.

1.1. Introdução

Neste curso os alunos irão aprender a usar um sensor de presença para detectar pessoas e acender um LED com Arduino. Pode até parecer um projeto simples, mas ele serve como base para a utilização do Arduino no controle de diversos outros dispositivos. Os conceitos aprendidos neste curso servirão tanto para o acionamento de LEDs quanto de outros dispositivos externos como motores, lâmpadas, e etc. Antes de nos direcionarmos de fato à solução do problema, a primeira parte deste curso visa promover um contato inicial dos alunos com conceitos de eletrônica e programação. Provavelmente os alunos do curso terão nenhum ou pouco contato com esses assuntos, portanto por se tratar de um curso de curta duração o objetivo não é ensiná-los de fato a programar. O foco desta atividade é o de apresentar os principais conceitos básicos sobre essas áreas, explicar o funcionamento e permitir a manipulação dos componentes, com o intuito de despertar o interesse nos alunos neste campo do conhecimento até então inacessível. Montar um circuito elétrico e criar o código de programação para fazer o LED piscar com Arduino geralmente é o primeiro passo para iniciantes, pois mostra que o aluno é capaz de programá-lo. Essa primeira parte é, portanto, um pequeno grande passo no aprendizado de eletrônica e programação e permite que o aluno possa se aventurar em projetos mais complexos depois.

Os alunos irão começar aprendendo o que são e como funcionam os materiais e componentes utilizados na atividade, principalmente sobre o Arduino. Verão que o LED possui polaridade e que deve ser ligado do jeito certo, além de conhecer a tabela de cores de resistores.

Após esta etapa, o professor irá explicar o funcionamento de circuitos elétricos básicos, incluindo a ligação de componentes e como a corrente “se movimenta” pelo circuito. Haverá uma discussão sobre fontes de energia e o primeiro circuito será montado. Na terceira etapa, os alunos aprenderão como utilizar o Arduino para atuar no papel de fonte para o circuito, possibilitando ligar e desligar o circuito utilizando programação. Eles irão aprender a montar um circuito com Arduino e a enviar o código de programação para a placa, além de entender o que cada palavra do código faz. Depois serão propostos desafios para que os alunos testem seus conhecimentos em circuitos elétricos e programação para Arduino. Ao final do curso, os alunos terão sido apresentados à componentes básicos de circuitos eletrônicos e aos principais conceitos sobre eletrônica básica. Além disso, terão um contato inicial com lógica de programação e serão capazes de desenvolver seus próprios códigos a partir de um código de exemplo que será fornecido.

Por fim, os alunos poderão utilizar o que aprenderam para resolver o problema principal deste curso: acender o LED quando o sensor de presença detectar movimento. Nesta parte do curso os alunos irão conhecer os diversos tipos de sensores disponíveis para uso junto com o Arduino. Depois, poderão programar o sensor de presença para resolver o problema proposto.

1.2. Situação problema

Podemos construir grandes projetos de automação com a placa Arduino, capazes de receber dados e controlar grandes equipamentos. Mas antes de nos aventurarmos por esses caminhos mais complexos, precisamos conhecer e aprender a criar projetos mais simples, mas de grandes ensinamentos. Assim, este curso propõe o desafio de usar o Arduino juntamente com um sensor de presença para acender um LED, ou seja, queremos automatizar o processo de acender a luz sem que alguém precise acionar uma tomada. Ao término do curso, o aluno terá conhecimento técnico e motivação suficientes para buscar por mais fontes de informação e aprender mais sobre Arduino, eletrônica e programação.

1.3. Questão motriz

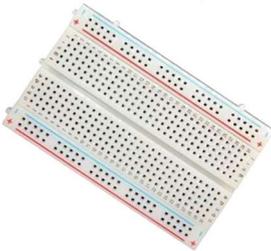
A questão motriz que nos motiva a realizar esta atividade é: precisamos de um sistema que acenda um LED toda vez que uma pessoa passar em frente ao Arduino. Como fazer isso? Para resolver esse problema utilizaremos uma lâmpada de LED de baixa potência, um sensor de presença e veremos circuitos e códigos de programação para controlar estes componentes. Os códigos que estudaremos para o LED e sensor de presença servirão de base para o controle de dispositivos mais complexos.

Materiais necessários

Para este curso utilizaremos alguns componentes eletrônicos juntamente com a placa Arduino. A seguir é apresentada a Tabela 1 em que são descritos os materiais necessários para a realização da atividade, juntamente com imagens dos equipamentos, quantitativo, e opções de substituição.

Tabela 1 – Materiais para o curso.

Item	Nome do item	Descrição	Quant.	Foto	Substituição
1	LED vermelho	LED monocromático na cor vermelha	2	 Fonte: Eletrogate (2022a)	LEDs de outras cores
2	Bateria	Bateria alcalina de 9V (qualquer marca)	1	 Fonte: Robocore (2022)	Caso o uso da bateria não seja possível, utilizar o pino de saída de 5V do Arduino
3	Resistor	Resistor de 510Ω	1	 Fonte: Baú da Eletrônica (2022a)	Resistores na faixa de 510Ω a 1kΩ
4	Resistor	Resistor de 300Ω	2	 Fonte: Baú da Eletrônica (2022b)	Resistores na faixa de 270Ω a 330Ω
5	Fios conectores	Fios para conexão de componentes	10	 Fonte: Eletrogate (2022b)	Não há substituto

6	Protoboard	Protoboard 400 pontos	1	 <p>Fonte: Mota (2022)</p>	Protoboard de outros tamanhos
7	Arduino	Placa Arduino Uno R3 + Cabo USB para Arduino	1	 <p>Fonte: Filipeflop (2022)</p>	Não há substituto
8	Sensor de presença	Sensor de Movimento e Presença PIR HC-SR501 Verificar se tem esse sensor no kit arduino	1	 <p>Fonte: Eletrogate (2022c)</p>	Não há substituto

2.1 O Arduino UNO

O Arduino é uma placa de código aberto projetada na Itália que surgiu para ser utilizada como forma de prototipagem eletrônica, com o objetivo de tornar a robótica e a eletrônica mais acessíveis à artistas, designers, *hobbistas* e qualquer pessoa interessada em criar objetos ou ambientes interativos. Idealizada por Massimo Banzi e David Cuartielles em 2005, a placa surgiu como uma maneira mais barata de estudar eletrônica e programação, além de ser uma excelente ferramenta para iniciantes no estudo das linguagens de programação, tornando o aprendizado de eletrônica e engenharia muito menos maçante e difícil. Desde sua criação, cada versão lançada do Arduino se mostrava mais simples e mais fácil de ser usada (MCROBERTS, 2018). Resumidamente, pode-se dizer que o Arduino é um pequeno computador que o programamos de acordo com a nossa necessidade. O Arduino possibilita a criação de projetos de diferentes tamanhos e complexidades, desde programas que acendem LEDs a sistemas de captação de dados provenientes de uma infinidade de sensores que possuam compatibilidade com a placa.

Outra característica essencial do Arduino é que todo o material, incluindo bibliotecas, hardware e o próprio software, é de código aberto, ou seja, pode ser reproduzido e usado por qualquer pessoa sem a necessidade do pagamento de direitos autorais a seus criadores (EVANS, NOBLE e HOCHENBAUM, 2013). Além disso, a existência de uma forte comunidade online, com sites e fóruns dedicados, permitiu a criação de um extenso acervo de bibliotecas e códigos na internet. Tanto entusiastas quanto profissionais compartilham seus códigos na internet, auxiliando ainda mais iniciantes em busca de aprendizado.

O Arduino pode ser programado por meio de uma linguagem de programação própria baseada na linguagem *Wiring*, semelhante à linguagem C++. Utilizando o IDE (ambiente de desenvolvimento integrado) específico, é possível escolher a placa que se está utilizando, além de contar com diversas bibliotecas e exemplos de utilização para programadores iniciantes (SOUZA, 2014).

De forma geral, cada placa Arduino é constituída por um microcontrolador, pinos digitais e analógicos de entrada e saída e uma porta USB. Além disso, o Arduino também acompanha um ambiente de desenvolvimento para a programação no computador. Existem diversos modelos de Arduino disponíveis no mercado, cada qual com sua peculiaridade, variando número de portas, tipo de conexão USB, etc. Uma das placas mais populares e a que utilizaremos nesta aula prática é a UNO, que apresenta baixo custo e possui recursos suficientes para a execução de variados projetos.

O Arduino UNO é equipado com um microcontrolador da série ATMEGA de 8 bits. A placa possui 14 pinos digitais de entrada e saída, sendo que seis destes podem ser usados como saídas analógicas. Possui também seis entradas analógicas, conexão USB e um conector de alimentação, podendo então ser alimentado via USB, ou por meio de baterias ou fontes externas (SOUZA, 2014).



Dica do Professor: Enquanto realiza a leitura e explicação sobre o Arduino UNO, pegue uma placa Arduino e mostre aos alunos os pinos e conexões descritas.

Além das portas digitais e analógicas, o Arduino UNO possui pinos de energia de 5V e 3,3V para alimentar outros componentes eletrônicos do circuito. Resumidamente, o *hardware* de um Arduino é descrito na Figura 1.

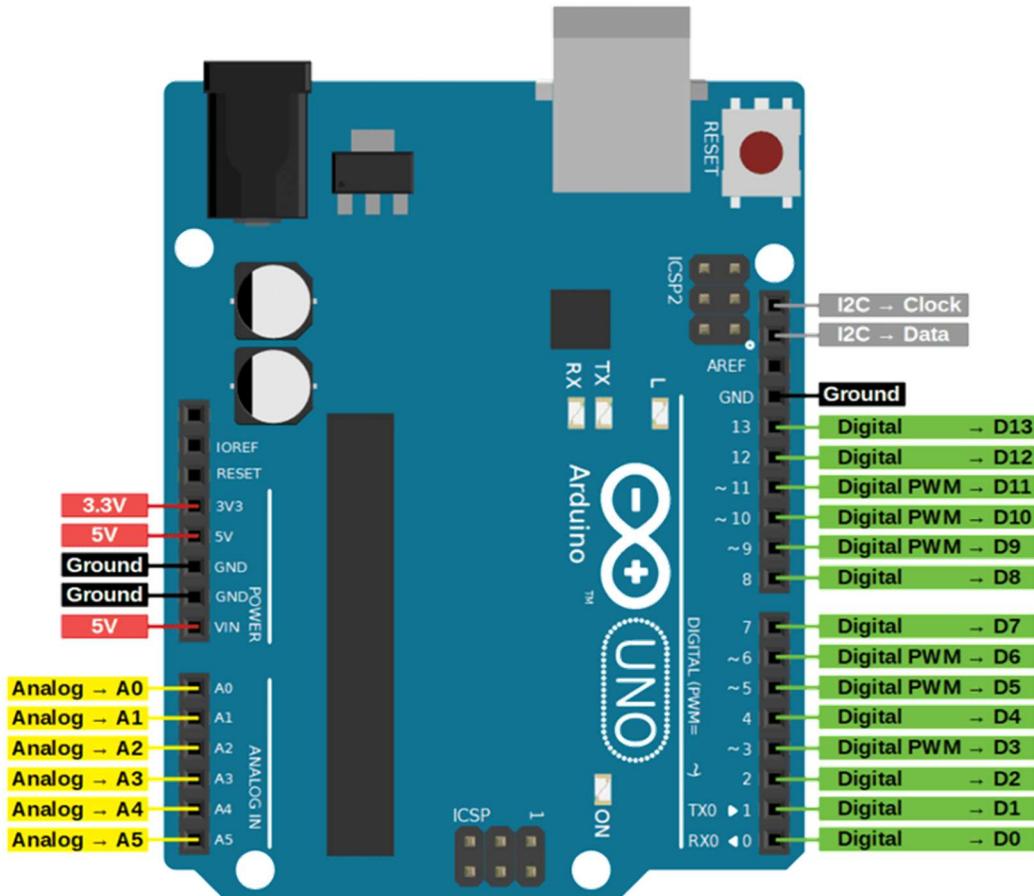


Figura 1 – Arduino UNO.
Fonte: DIYIOT (2022).

Para realizar o download do IDE do Arduino em um computador, vá até a página oficial do Arduino (arduino.cc), clique em “Software” e escolha a opção relativa ao seu sistema operacional. Você pode escolher fazer uma doação para os criadores do Arduino, ou apenas fazer o download do software. Para baixar o programa, clique em “Just download” e escolha onde em seu computador irá salvar o arquivo.

Para instalar o IDE, abra a pasta onde o arquivo foi baixado e clique duas vezes nele. Assim que o assistente de instalação se abrir, aceite o termo da licença do software, clicando em “I Agree” e depois selecione as opções de instalação. Se você não é um usuário avançado, marque todas as opções. Entre as opções, está a instalação do driver de comunicação com a placa Arduino, que é muito importante, além da criação de atalhos. Em seguida, o assistente pedirá para que escolha o local de instalação. Caso esteja em dúvida,

deixe como está. Clique em “Install” e espere o procedimento terminar. Ao final, o programa estará instalado e pronto para uso.

2.2 Diodo Emissor de Luz (LED)

Um LED é um diodo emissor de luz (do inglês, *Light Emitting Diode*) que realiza um procedimento eletroquímico chamado de eletroluminescência, que é quando um material emite resposta luminosa se percorrido por uma corrente elétrica, sendo basicamente uma lâmpada que consome pouca energia. Um LED pode ser produzido em muitas cores e é utilizado principalmente como indicador luminoso em projetos eletrônicos, estando presente em quase todos os dispositivos eletrônicos para indicar que os mesmos estão energizados.

O LED, assim como alguns componentes, possui polaridade. Assim, possui um lado positivo “+” e um lado negativo “-“. Para que um LED funcione corretamente, primeiramente é necessário que a polarização esteja correta. Para que isso ocorra você deve ligar o terminal positivo do LED no VCC e o negativo no GND do seu circuito, do contrário não haverá condução e isso prejudicará todo o restante do circuito. A identificação de polos de um LED pode ser feita pela estrutura do corpo do mesmo, como mostra a Figura 2. O polo positivo (anodo) conta com maior haste metálica para a ligação com a placa, além de encapsulamento plástico circular do lado positivo, enquanto o lado negativo (catodo) apresenta menor haste de contato e encapsulamento achatado, como ilustrado.

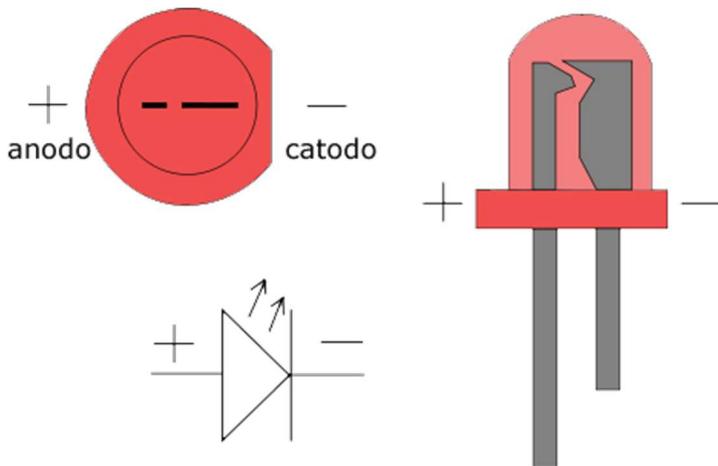


Figura 2 – Diodo emissor de luz - LED.
Fonte: Locatelli (2022).

2.3 Resistores

Resistores são componentes eletrônicos cuja principal função é limitar o fluxo de corrente elétrica que passa em um circuito por meio da conversão da energia elétrica em energia térmica e estão entre os dispositivos mais simples e mais comuns entre os circuitos eletrônicos.

Uma boa notícia é que diferentemente dos LEDs, resistores não possuem polaridade, por isso você pode ligar um resistor tranquilamente pois ele não tem lado positivo ou negativo, ou seja, tanto faz o lado em que é conectado.

Os resistores possuem diferentes valores de resistência, sendo que quanto maior a resistência, mais ele irá limitar a corrente que passa por ele. O valor de um resistor é fixo e é indicado pelas faixas de cores pintadas nele, de acordo com a Figura 3.



Dica do Professor: Peça aos alunos para usarem a tabela de cores e confirmarem qual o valor da resistência dos resistores que serão utilizados na prática.

Cor	1ª Faixa	2ª Faixa	Faixa multiplicadora	Tolerância
Preto	0	0	$\times 1\Omega$	-
Marrom	1	1	$\times 10\Omega$	$\pm 1\%$
Vermelho	2	2	$\times 100\Omega$	$\pm 2\%$
Laranja	3	3	$\times 1k\Omega$	-
Amarelo	4	4	$\times 10k\Omega$	-
Verde	5	5	$\times 100k\Omega$	$\pm 0,5\%$
Azul	6	6	$\times 1M\Omega$	$\pm 0,25\%$
Violeta	7	7	$\times 10M\Omega$	$\pm 0,1\%$
Cinza	8	8	-	$\pm 0,05\%$
Branco	9	9	-	-
Dourado	-	-	$\times 0,10\Omega$	$\pm 5\%$
Prateado	-	-	$\times 0,01\Omega$	$\pm 10\%$
Sem cor	-	-	-	$\pm 20\%$

Figura 3 – Tabela de códigos de resistores.
Fonte: Embarcados (2022).

2.4 Protoboard

Protoboards são ferramentas que auxiliam no desenvolvimento de protótipos de circuitos eletrônicos. Com elas, torna-se desnecessária a soldagem de componentes eletrônicos em uma placa de circuito impresso. Sem dúvida, a protoboard é uma das ferramentas mais importantes para iniciantes na eletrônica.

Como dito, a grande vantagem de uma protoboard é permitir montar e testar circuitos sem que se tenha que soldar os componentes. Dessa forma, você pode testar diferentes configurações e componentes para o circuito, até encontrar a configuração final que atenda

seus objetivos. Cada protoboard é composta por furos interconectados por um material condutor localizado abaixo da camada de plástico, conforme mostra a Figura 4:

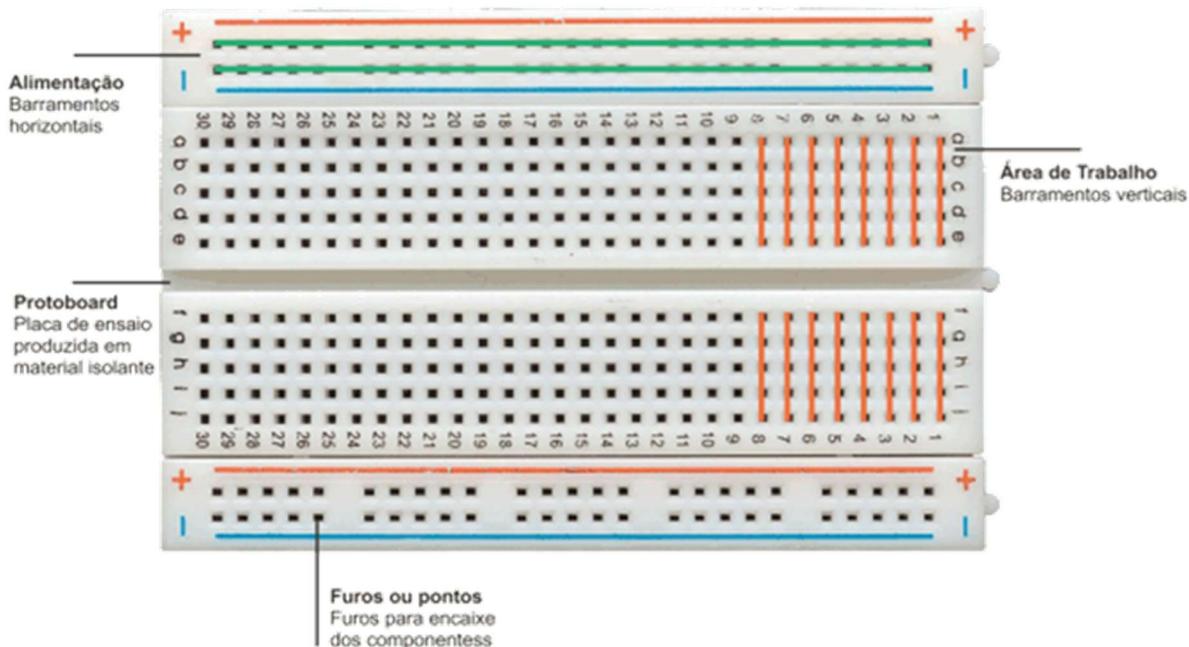


Figura 4 – Barramentos de uma protoboard.
Fonte: Portal Vida de Silício (2018).

As linhas representam as ligações internas da protoboard, que podem ser divididas em duas seções. A primeira seção, formada pelas duas linhas superiores e inferiores, geralmente é utilizada para distribuir a alimentação elétrica do circuito. A segunda seção, formada pela parte central da protoboard, é destinada à montagem dos componentes. É importante você entender que como cada coluna (linha vertical na imagem) possui 5 furos, e estes furos estão interconectados, ao inserir um componente em um deles, ele estará eletricamente conectado aos outros quatro furos da coluna.

2.5 Sensor de Movimento e Presença

Todos os objetos que conhecemos emitem alguma quantidade de energia térmica na forma de radiação infravermelha e, quanto mais quente for o objeto, mais radiação é emitida. O sensor de movimento e presença HC-SR501 PIR é um componente eletrônico que usa um sensor infravermelho passivo (PIR) como detector de movimento, sendo capaz de medir a luz infravermelha que irradia de objetos em seu raio de detecção. O sensor PIR é construído com um sensor que pode detectar níveis de radiação infravermelha e uma lente especial chamada lente de Fresnel, responsável por focalizar os sinais infravermelhos do ambiente diretamente no sensor. O sensor de HC-SR501 PIR possui três pinos: GND (terra), saída digital e VCC (alimentação).

Para conectar o sensor ao Arduino, ligue o pino de 5V da placa ao VCC do sensor e o GND da placa ao GND do sensor. O pino de dados refere-se ao sinal de saída do sensor, que será ALTO indicando a detecção de movimento ou BAIXO quando não for detectado movimento. Quando há movimento, a saída é acionada pelo movimento detectado, esta permanecendo em nível ALTO por um curto período de tempo, mesmo se não houver mais movimento. O tempo em ALTO pode ser configurado variando-se o potenciômetro 'Time' na parte inferior do sensor, sendo que o outro potenciômetro altera a sensibilidade.

Parte 1

3.1 Conhecendo os materiais (Etapa 1)

Nesta primeira etapa de execução da prática, peça aos alunos que leiam as informações sobre os materiais e componentes utilizados na atividade. Eles irão conhecer sobre o Arduino, LEDs, resistores e protoboard. Se preferir, faça a leitura em voz alta juntamente a eles. Durante a leitura sobre o Arduino, pegue uma placa e mostre onde cada pino e conexão mencionados se encontram. Durante a leitura sobre resistores, se possível tenha à mão alguns resistores de diferentes valores de resistência e peça para que identifiquem as resistências usando a tabela de cores presente no material. Este primeiro momento fornecerá a base teórica necessária para a realização das atividades seguintes.

Os alunos devem responder as seguintes questões:

1) Complete a tabela a seguir com informações sobre os materiais:

Pergunta sobre os materiais	Resposta
Quantos pinos digitais possui o Arduino?	14
Quantos pinos analógicos possui o Arduino?	6
O LED possui polaridade?	SIM
O resistor possui polaridade?	NÃO
A “perna” maior do LED corresponde ao polo positivo ou negativo?	POSITIVO
A “perna” menor do LED corresponde ao polo positivo ou negativo?	NEGATIVO

2) Porque devemos usar um resistor quando vamos ligar um LED?

Resistores são componentes eletrônicos cuja principal função é limitar o fluxo de corrente elétrica que passa em um circuito, por isso devem ser usados com o LED para limitar a corrente que passa por ele, evitando que se queime.

3.2 Entendendo um circuito elétrico (Etapa 2)

Nesta etapa o aluno aprenderá sobre o funcionamento de circuitos elétricos básicos, incluindo a ligação em série de componentes e como a corrente “se movimenta” pelo circuito.

Comece explicando que um circuito elétrico basicamente é um laço por onde circula “energia” elétrica, energizando cada componente e fazendo-o funcionar.

Explique como ligar os componentes em série usando os jumpers (fios) e peça aos alunos que montem o circuito exibido na Figura 5 composto por bateria + resistor (510 Ω) + LED.

A seguir destaco alguns pontos importantes a serem explicados:

- Explique que o lado positivo da fonte é chamado de VCC e o lado negativo é chamado de GND. Uma fonte pode ser uma bateria ou pilha que quando ligada faz com que o circuito funcione. Nesse exemplo, o LED se acende quando a fonte de energia é ligada e se apaga ao se desligar a fonte.
- O resistor é utilizado para reduzir a corrente que passa pelo circuito inteiro. Como a bateria possui 9V, o resistor de 510 Ω atende as necessidades.
- O pino maior do LED é o polo positivo e o menor, o negativo.

Dê um tempo para que concluam esta atividade, e ajude os alunos que não conseguirem montar. Após a montagem, reforce o papel de cada componente do circuito e como a corrente flui da bateria para o LED, passando pelo resistor. Após esta etapa, o aluno já terá uma noção melhor sobre os componentes eletrônicos e como construir um circuito com eles.



Atenção: No caso de falta de baterias, utilize o pino de 5V do Arduino como fonte de alimentação.

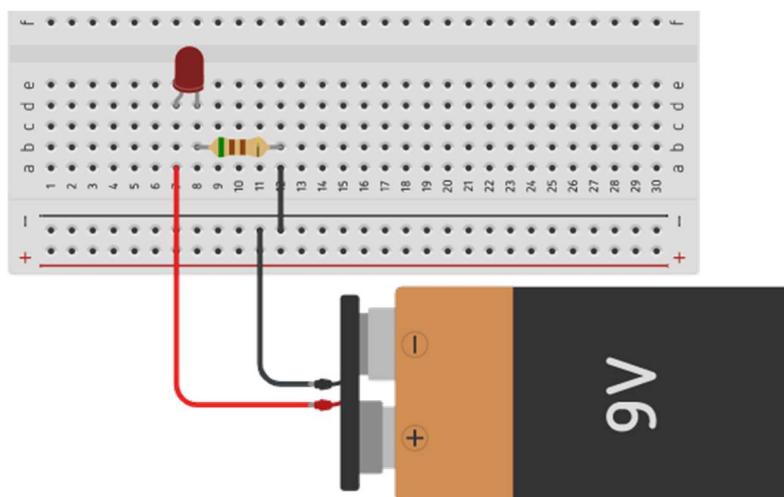


Figura 5 – Circuito para a Etapa 2.
Fonte: Construído com o auxílio de TinkerCAD (2022).

- 1) Qual é o papel do resistor nesse circuito?

O resistor limita o fluxo de corrente elétrica que passa no circuito, “protegendo” o LED de uma corrente maior que poderia danificá-lo.

3.3 Automatizando o circuito (Etapa 3)

Na etapa 3 o aluno verá que o Arduino pode fazer o papel de fonte para o circuito, possibilitando ligar e desligar o circuito utilizando programação.

Para começar, o aluno deverá remover a bateria e construir um circuito composto por Arduino + resistor (300 Ω) + LED (conectado ao pino 13 do Arduino) como exibido na Figura 6. Durante a construção, mostre novamente onde estão os pinos digitais do Arduino.

Comente com os alunos que o pino 13 está conectado ao lado positivo do LED, enquanto o GND ao pino negativo, por conta da polaridade deste componente. Além disso, mostre que o resistor agora é de 300 Ω , porque a tensão da fonte que era de 9V da bateria caiu para 5V do Arduino.

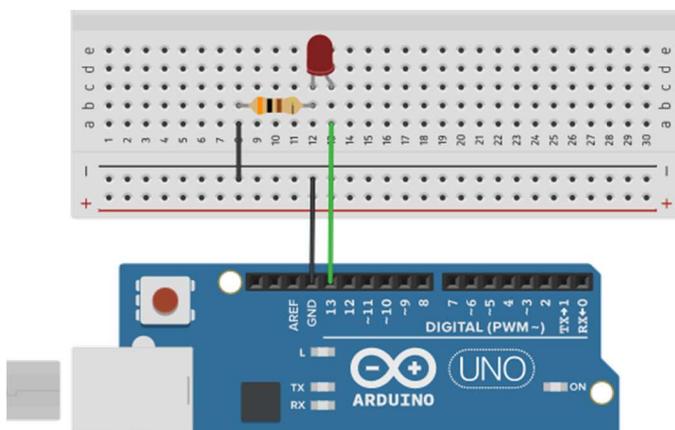


Figura 6 – Circuito para a Etapa 3.
Fonte: Construído com o auxílio de TinkerCAD (2022).

Após a construção, os alunos deverão conectar o cabo USB ao Arduino e ao computador e abrir o IDE do Arduino pelo atalho que se encontra na área de trabalho. Caso o IDE ainda não esteja instalado, consulte o passo-a-passo detalhado na seção 2.1 deste curso. O professor deve apresentar o IDE do Arduino, destacando os principais conceitos exibidos na Figura 7. Destaque a função dos botões Verificar e Carregar e mostre aos alunos as configurações de placa e porta COM em Ferramentas. Como mostra a Figura 8, em **Ferramentas -> Placa** deve ser escolhida a placa Arduino UNO e em **Ferramentas -> Porta** deve ser escolhida a porta COM em que se encontra conectado o Arduino. Este procedimento deve ser realizado todas as vezes em que se conectar a placa ao computador.

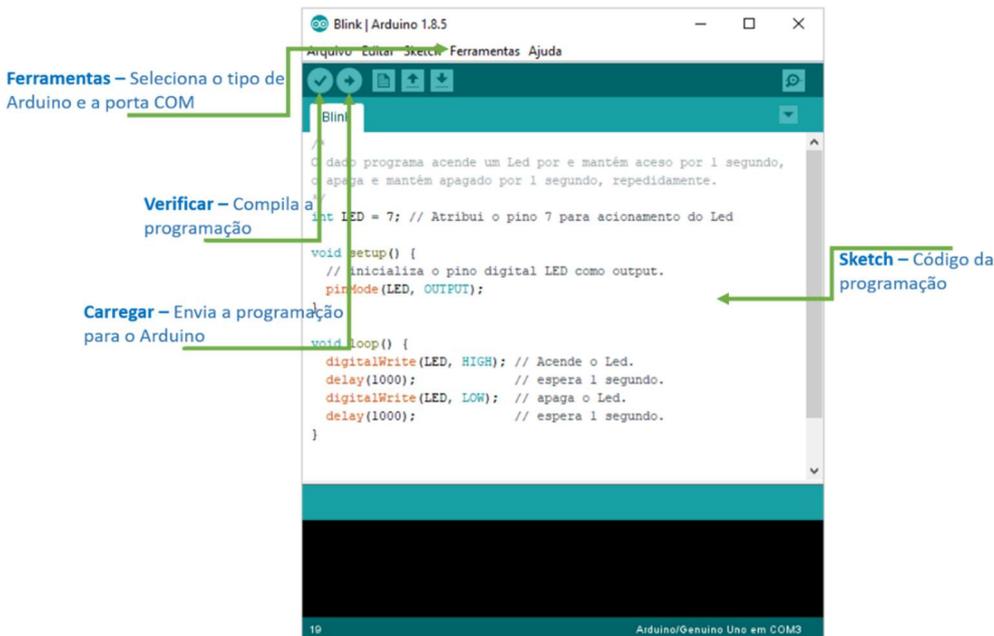


Figura 8 – IDE do Arduino.

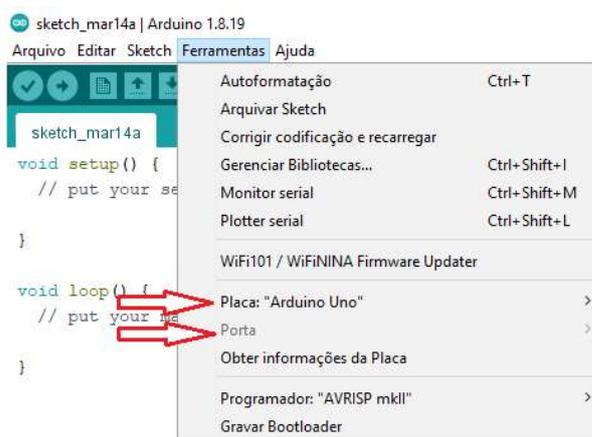


Figura 9 – Configurações do IDE do Arduino.



Atenção: A não definição da placa e porta COM no IDE é uma das principais causas de erros de funcionamento de código feitos por iniciantes, por isso reforce isso com os alunos.

Entendido o IDE do Arduino, peça para que digitem o código de programação da Figura 9. Após os alunos digitarem, o professor deve seguir para o código e explicar brevemente aos alunos cada linha do código de programação. O objetivo aqui não é ensinar os alunos a programarem, mas sim que tenham um primeiro contato com um código de programação e entendam os conceitos básicos de lógica de programação, como declaração

e manipulação de variáveis e o papel das funções **digitalWrite()**, **delay()**, **loop()** e **setup()** no código. A seguir, a explicação do código:

```
int pinoLED = 13;
void setup() {
  pinMode(pinoLED, OUTPUT);
}
void loop() {
  digitalWrite(pinoLED, HIGH);
  delay(1000);
  digitalWrite(pinoLED, LOW);
  delay(1000);
}
```

Figura 9 – Código para a Etapa 3.

A instrução **int pinoLED = 13** atribui o valor 13 à variável LED, que será utilizada para representar o pino digital 13. As variáveis são utilizadas para armazenar dados. A vantagem em se utilizar variáveis é que se você decidir remover o LED do pino 13 e passar para o 10, por exemplo, não será necessário alterar o código em vários locais, basta alterar o valor da variável **pinoLED**.

A função **void setup()** é executada somente uma vez assim que o Arduino é ligado. Nela, o comando **pinMode(pinoLED, OUTPUT)** define o pino digital 13 do Arduino como um pino de saída.

A função **void loop()** é executada indefinidamente enquanto o Arduino estiver ligado. No **loop()**, usamos a função **digitalWrite(pinoLED, HIGH)** para colocar o pino 13 em nível alto (5V), acendendo o LED. A palavra **HIGH** é uma constante que significa 1, ou seja, nível alto. Logo após, aguardamos um intervalo de 1 segundo através da função **delay(1000)**. Essa função funciona com milissegundos, por isso utilizamos 1000.

Para apagar o LED usamos a função **digitalWrite(pinoLED, LOW)**, colocando o pino 13 em nível baixo (0V). A palavra **LOW** é uma constante que significa 0, ou seja, nível baixo. Logo após, adicionamos mais um intervalo de 1 segundo com a função **delay(1000)**.

Reforce aos alunos que a fonte de energia agora vem do pino 13 do Arduino. Com a programação acima, é possível ligar e desligar a energia do pino 13, fazendo com que o LED se acenda e apague.

Após essa explicação, oriente os alunos para que cliquem no botão em formato de “seta” (veja a Figura 7, trata-se do botão de “carregar”) para carregar o código para a placa. Assim que o código estiver carregado, aparecerá uma mensagem dizendo ‘Carregado’ na parte inferior do IDE.

O LED deve permanecer aceso por 1 segundo e depois apagado por mais um segundo, em um comportamento de “pisca-pisca”. Este comportamento é observado indefinidamente devido à função **loop()**. Reforce com os alunos como este código os

aproxima do conceito inicial de automação, uma vez que o estado de um componente está sendo alterado sem a interação humana.

1) O que faz a linha de código **pinMode(pinoLED,OUTPUT)** ?

A função **pinMode()** define o pino digital ao qual o LED está conectado no Arduino como um pino de saída.

2) O que significam as palavras HIGH e LOW no código?

A palavra **HIGH** é uma constante que significa 1, ou seja, nível alto. A palavra **LOW** significa 0, ou seja, nível baixo.

3.4 Desafios (Etapa 4)

Por fim, na etapa 4 os alunos devem fazer modificações no código para realizar alguns desafios. Eles devem utilizar a explicação do professor sobre o código e por analogia, construir os códigos dos seguintes desafios:

Desafio 1: Trocar o pino ao qual o LED está conectado para o pino 10 e fazê-lo piscar assim como feito anteriormente.

Peça aos alunos que retirem a conexão do pino 13 e passem para outro pino digital. Esta mudança na conexão do circuito deve ser acompanhada pela mudança no código da declaração do pino. Considerando-se que o aluno passe o LED para o pino 10, o código deverá ser alterado para o exibido na Figura 10.

```
int pinoLED = 10;
void setup() {
  pinMode(pinoLED, OUTPUT);
}
void loop() {
  digitalWrite(pinoLED, HIGH);
  delay(1000);
  digitalWrite(pinoLED, LOW);
  delay(1000);
}
```

Figura 10 – Código para o Desafio 1.

1) O que acontece se você apenas trocar o fio do pino ao qual o LED está conectado, mas não mudar a programação? O LED ainda acenderá?

Devemos alterar tanto o circuito quanto a programação, por isso, além de trocar o fio do pino ao qual o LED está conectado, temos que mudar também a linha em que é declarada a variável pinoLED.

- 2) Qual linha do código você teve que alterar para que o Arduino soubesse qual o novo pino ao qual o LED estava conectado?

Deve-se alterar a linha de código em que é declarada a variável pinoLED: int pinoLED = 10;

Desafio 2: Alterar o tempo em que o LED fica ligado para 3 segundos e desligado para 2 segundos.

O professor já terá explicado aos alunos o papel da função **delay()** no código, por isso, peça para que alterem o tempo que serve de parâmetro para esta função. A Figura 11 exibe a resposta a esse desafio.

```
int pinoLED = 10;
void setup() {
  pinMode(pinoLED, OUTPUT);
}
void loop() {
  digitalWrite(pinoLED, HIGH);
  delay(3000);
  digitalWrite(pinoLED, LOW);
  delay(2000);
}
```

Figura 11 – Código para o Desafio 2.

- 1) Qual linha de código você teve que alterar para que o tempo mudasse?

Deve-se alterar as linhas correspondentes à função delay(): O primeiro delay(1000) para delay(3000) e o segundo delay(1000) para delay(2000);

Desafio 3: Adicionar mais um LED ao circuito e fazer com que os dois LEDs se acendam de forma alternada.

Agora, os alunos devem usar o circuito e código das atividades anteriores para adicionar mais um LED ao circuito. Supondo que o novo LED 1 seja conectado ao pino 5 e o LED 2 ao pino 6, as Figuras 12 e 13 exibem, respectivamente, o circuito e código que resolvem este desafio.

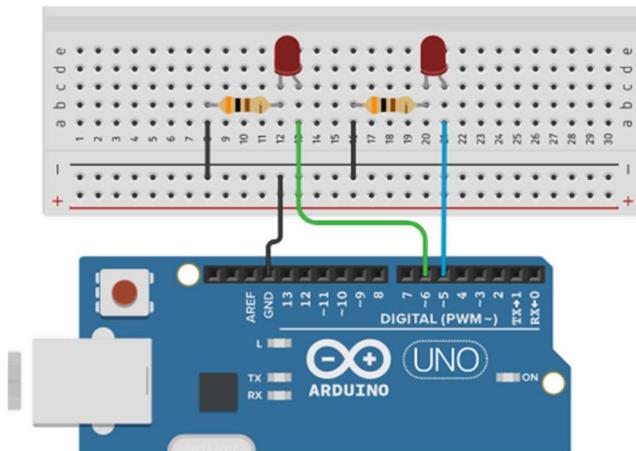


Figura 12 – Circuito para o Desafio 3.
Fonte: Construído com o auxílio de TinkerCAD (2022).

```
int pinoLED = 5;
int pinoLED2 = 6;
void setup() {
  pinMode(pinoLED, OUTPUT);
  pinMode(pinoLED2, OUTPUT);
}
void loop() {
  digitalWrite(pinoLED, HIGH);
  digitalWrite(pinoLED2, LOW);
  delay(1000);
  digitalWrite(pinoLED, LOW);
  digitalWrite(pinoLED2, HIGH);
  delay(1000);
}
```

Figura 13 – Código para o Desafio 3.

- 1) Quais linhas você terá que adicionar ao código para “informar” ao Arduino sobre o pino de conexão e modo de operação (saída ou entrada) do novo LED?

Int pinoLED2 = 6; e pinMode(LED2,OUTUPUT);

- 2) Qual função deverá ser usada para acender o novo LED? E para apaga-lo?

digitalWrite(pinoLED2,HIGH); e digitalWrite(pinoLED2,LOW); respectivamente.

Parte 2

3.5 Entendendo o problema (Etapa 5)

Nesta segunda parte da aula iremos usar nossos conhecimentos para ajudar a resolver um problema. A questão que irá direcionar nossos esforços a partir de agora é a seguinte: precisamos de um sistema que acenda um LED toda vez que uma pessoa passar em frente ao Arduino. Como fazer isso?

Apresente aos alunos os diversos sensores que o Kit Arduino possui: sensor ultrassônico de distância, sensor de umidade, sensor infravermelho de proximidade, sensor de luz, etc. Deixe que peguem os sensores e percebam como é a construção de cada um. O objetivo desta etapa é que tenham contato e conheçam a ampla gama de possíveis acessórios que podem ser utilizados com o Arduino. Não se apresse nas explicações e deixe que façam perguntas sobre possíveis projetos com cada um destes sensores. Por exemplo, pergunte a eles: “Se eu quiser saber quantos °C está esta sala agora, qual sensor devo utilizar?”.

Selecione os principais sensores do kit e explique o funcionamento e função de cada um. A partir dessa explicação, busque promover o diálogo entre os alunos para que opinem sobre qual o melhor sensor para ser aplicado na resolução do problema apresentado acima. Peça para que expliquem o porquê de suas decisões.

Os alunos devem responder à seguinte questão:

- 1) Porque o sensor infravermelho PIR é uma boa opção para resolver este problema?

O sensor PIR é capaz de detectar movimentos e a presença de alguma pessoa, por isso é uma boa opção para nos ajudar a resolver esse problema que quer justamente acender uma lâmpada quando uma pessoa passar pelo sensor.

3.6 Resolvendo o problema (Etapa 5)

Nesta aula, utilizaremos o sensor de proximidade infravermelho PIR para solucionar o problema apresentado. Definido o sensor de presença infravermelho, explique novamente o funcionamento e ligações do sensor e peça aos alunos que façam as ligações de acordo com o circuito exibido na Figura 14 composto por Arduino + resistor (300 Ω) + LED + Sensor PIR.

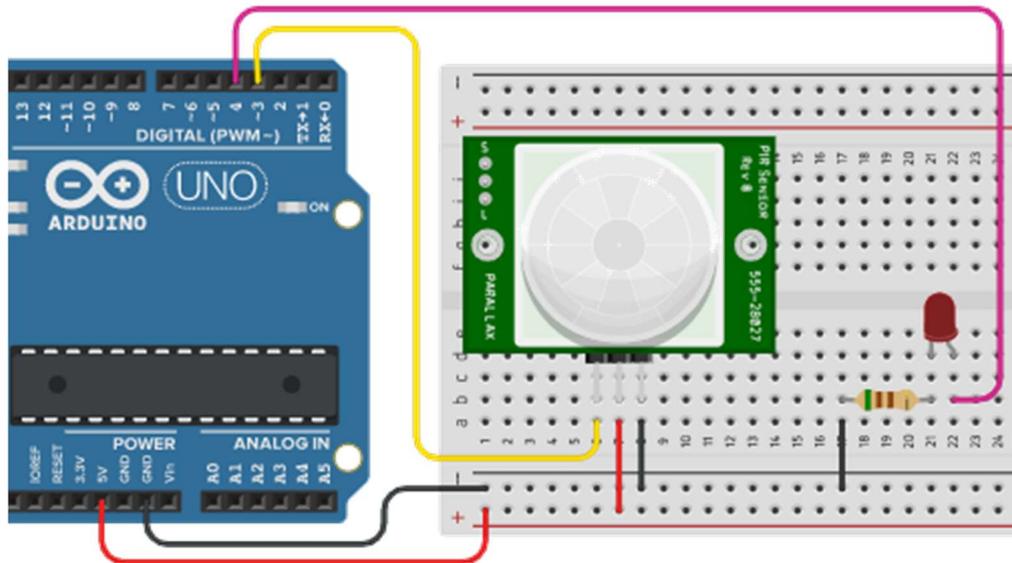


Figura 14 – Circuito para a Etapa 5.

Montado o circuito, peça para que digitem o código exibido na Figura 15 no IDE do Arduino.

```
int pinoLED = 4;
int sensor = 3;
int estado_sensor;
void setup() {
  pinMode(pinoLED, OUTPUT);
  pinMode(sensor, INPUT);
}
void loop() {
  estado_sensor = digitalRead(sensor);
  if(estado_sensor == 1)
    digitalWrite(pinoLED, HIGH);
  else
    digitalWrite(pinoLED, LOW);
}
```

Figura 15 – Código para a Etapa 5.

Este é código mais complexo que vimos até agora. Com ele, conseguiremos acender o LED quando alguém passar pelo sensor. Explique aos alunos o papel de cada linha e função do código. Ressalto que o objetivo aqui não é explicar a lógica de programação de forma aprofundada, apenas o suficiente para entenderem as funções e conseguirem utilizá-las. Vemos que há a declaração das variáveis **pinoLED**, **sensor** e **estado_sensor**. A variável **pinoLED** representa o pino digital ao qual o LED está conectado; **sensor** representa o pino digital ao qual o sensor PIR está conectado; e **estado_sensor** é uma variável que será utilizada posteriormente para sabermos se o sensor está identificando movimento ou não. A função **pinMode()** define o estado dos pinos: **OUTPUT** (saída) ou **INPUT** (entrada).

A função `digitalRead()` realiza uma “leitura no pino digital” e retorna o estado do sensor. O sensor PIR irá emitir o sinal **1** caso detecte algum objeto e **0** caso não detecte. A estrutura `if()` (“se” em português) serve para comparar o valor da variável `estado_sensor` a um valor. Nesse caso, se `estado_sensor` for igual a **1**, quer dizer que o sensor identificou algum objeto, então o LED se acende por meio da função `digitalWrite()`, que já vimos antes. Caso contrário, `estado_sensor` será igual a **0**, representando que o sensor não identificou nenhum objeto, então o LED continuará apagado.

1) Qual o objetivo da função `digitalRead()` no código?

A função `digitalRead()` serve para realizar a leitura do pino digital ao qual o sensor está conectado. O sensor envia a informação para o Arduino, que pode ser 0 ou 1, e a função `digitalRead()` é utilizada para ler essa informação.

3.7 Desafio (Etapa 6)

Agora que conseguimos identificar quando uma pessoa passa, os alunos devem fazer modificações no código para realizar um desafio. Eles devem utilizar a explicação do professor sobre o código e por analogia, construir o seguinte código:

Desafio: Adicionar outro LED ao circuito e fazer com que este fique aceso quando o sensor não identificar movimento.

Neste desafio teremos dois LEDs no circuito, cada um para representar um estado de identificação do sensor. Enquanto não for identificado movimento, o LED1 deve permanecer apagado e o LED2 aceso. Quando movimento for detectado pelo sensor, o LED1 se acende e o LED2 se apaga. Supondo que o LED 1 seja conectado ao pino 4 e o LED 2 ao pino 5. As Figuras 16 e 17 exibem a resposta a esse desafio.

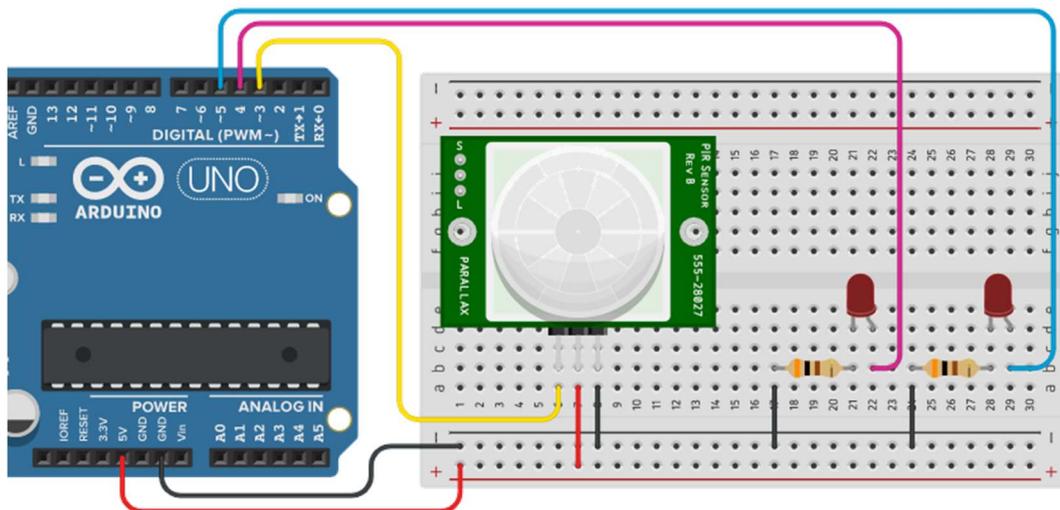


Figura 16 – Circuito para o desafio da Etapa 6.

```
int pinoLED = 4;
int pinoLED2 = 5;
int sensor = 3;
int estado_sensor;
void setup() {
  pinMode(pinoLED, OUTPUT);
  pinMode(pinoLED2, OUTPUT);
  pinMode(sensor, INPUT);
}
void loop() {
  estado_sensor = digitalRead(sensor);
  if(estado_sensor == 1){
    digitalWrite(pinoLED, HIGH);
    digitalWrite(pinoLED2, LOW);
  }
  else{
    digitalWrite(pinoLED, LOW);
    digitalWrite(pinoLED2, HIGH);
  }
}
```

Figura 17 – Código para o desafio da Etapa 6.

4.1 Erros comuns

Os alunos provavelmente irão cometer erros de iniciantes, por isso esta seção visa apresentar os erros comuns que podem acontecer ao longo do curso e suas respectivas soluções.

1. Deixar de selecionar a placa Arduino e a porta COM

Caso o IDE do Arduino não consiga completar o carregamento do código para a placa, verifique se as configurações de uso foram feitas corretamente. Como mostra a Figura 14, em **Ferramentas -> Placa** deve ser escolhida a placa Arduino UNO e em **Ferramentas -> Porta** deve ser escolhida a porta COM em que se encontra conectado o Arduino. Este procedimento deve ser realizado todas as vezes em que se conectar a placa ao computador.

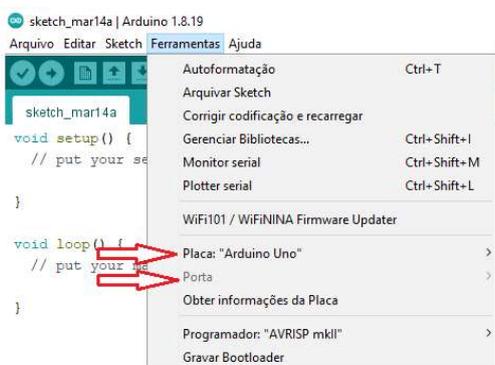


Figura 14 – Configurações iniciais do IDE do Arduino.

2. Conexões incorretas na protoboard e no Arduino

Caso o projeto não funcione, verifique se os jumpers estão na mesma coluna dos terminais dos componentes, completando assim a conexão. Além disso, verifique se os jumpers estão ligados aos pinos corretos no Arduino de acordo com o que foi definido pelo código de programação. Exemplo: Se no código foi definido o pino 13, o jumper deve estar conectado ao pino 13 do Arduino.

3. LED não acende

Se o LED não acende, você já verificou o código e não encontrou erros, verifique se a conexão do LED está de acordo com sua polaridade ou se está invertida. Caso esteja invertida, o LED não irá acender. O terminal positivo do LED deve estar conectado ao pino digital do Arduino e o terminal negativo ao GND.

4. Sensor de presença PIR não funciona corretamente

Quando há movimento, a saída é acionada pelo movimento detectado, esta permanecendo em nível ALTO por um curto período de tempo, mesmo se não houver mais movimento. O tempo em ALTO pode ser configurado variando-se o potenciômetro 'Time' na parte inferior do sensor, sendo que o outro potenciômetro altera a sensibilidade. Caso o sensor não esteja detectando corretamente o movimento, é possível ajustar a duração de seu tempo de espera ou sensibilidade para estabilização através dos potenciômetros amarelos em baixo do sensor.

Considerando-se o sensor com a lente de Fresnel para cima, o potenciômetro à direita serve para ajustarmos a distância de detecção do sensor. Girando no sentido horário, quanto mais à direita, maior a faixa de detecção, que pode ser de aproximadamente 3 a 7 metros. O potenciômetro à esquerda serve para ajustarmos o tempo de duração da saída, sendo que quanto mais à direita, maior o tempo, que pode ser de aproximadamente 5 a 300 segundos.

5. Último recurso

Caso você já tenha feito todas as sugestões anteriores e mesmo assim o código não funciona, um último recurso é reiniciar a conexão do Arduino. Para isso, siga o seguinte procedimento:

- a) Copie o seu código (Selecione tudo -> CTRV + C);
- b) Feche o IDE do Arduino;
- c) Retire o cabo USB do computador;
- d) Abra novamente o IDE;
- e) Cole o seu código (CTRV + V);
- f) Reconecte o cabo USB ao computador e aguarde alguns segundos;
- g) Configure novamente placa e porta COM;
- h) Transfira o código novamente para a placa e verifique o funcionamento.

Referências

BAU DA ELETRONICA. **Resistor 300 ohms**. 2022b. Disponível em: <<http://www.baudaeletronica.com.br/resistor-300r-5-1-4w.html>> Acesso em: 25 set. 2022.

BAU DA ELETRONICA. **Resistor 510 ohms**. 2022a. Disponível em: <<http://www.baudaeletronica.com.br/resistor-510r-5-1-4w.html>> Acesso em: 25 set. 2022.

CASA DA ROBÓTICA. **Protoboard**. 2022. Disponível em: <<http://www.casadarobotica.com/prototipagem-ferramentas/prototipagem/protoboard/protoboard-400-furos-65-fios-jumpers-macho-x-macho>> Acesso em: 25 set. 2022.

DIYIOT. **Tutorial Arduino UNO**. 2022. Disponível em: <<http://diyi0t.com/Arduino-uno-tutorial/>> Acesso em: 25 set. 2022.

ELETROGATE. **Jumpers**. 2022a. Disponível em: <<http://www.eletrogate.com/jumpers-macho-macho-40-unidades-de-10-cm>> Acesso em: 25 set. 2022.

ELETROGATE. **LED**. 2022b. Disponível em: <<http://www.eletrogate.com/led-difuso-5mm-vermelho>>. Acesso em: 25 set. 2022.

ELETROGATE. **Sensor de Movimento Presença PIR HC-SR501**. 2022c. Disponível em: <<http://www.eletrogate.com/sensor-de-movimento-presenca-pir>>. Acesso em: 06 nov. 2022.

EMBARCADOS **Código de cores de resistores**. 2022. Disponível em: <<http://embarcados.com.br/codigo-de-cores-de-resistores>> Acesso em: 26 set. 2022.

EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. **Arduino em ação**. São Paulo: Novatec Editora, 2013.

FILIFELOP. **Arduino UNO**. 2022. Disponível em: <<http://www.filipeflop.com/produto/placa-uno-r3-cabo-usb-para-arduino>> Acesso em: 25 set. 2022.

LOCATELLI, Caroline. **O que é um LED?**. Curto Circuito, 2022. Disponível em: <<http://www.curtocircuito.com.br/blog/eletronica-basica/o-que-e-um-led>> Acesso em: 25 set. 2022.

MICROBERTS, Michael. **Arduino básico**. Novatec Editora, 2018.

MOTA, Allan. **Protoboard**. Portal Vida de Silício, 2018. Disponível em: <<http://portal.vidadesilicio.com.br/protoboard>> Acesso em: 25 set. 2022.

ROBOCORE. **Bateria de 9V**. 2022. Disponível em: <<http://www.robocore.net/bateria/bateria-de-9v>> Acesso em: 25 set. 2022.

SOUZA, Vítor A. **Programação Para Arduino**. Clube de Autores, 2014.

TINKERCAD. 2022. Disponível em: <<http://www.tinkercad.com>> Acesso em: 20 set.

Currículo do autor



Formado em Engenharia de Controle e Automação pela Universidade Federal de Ouro Preto – UFOP, Mestre em Engenharia Agrícola pela Universidade Federal de Viçosa – UFV e Doutorando em Engenharia Elétrica pela Universidade Federal de Juiz de Fora - UFJF. Durante a graduação, participou da Equipe Rodetas de Futebol de Robôs, alcançando dois prêmios internacionais. Atuou como professor substituto no Centro Federal de Educação Tecnológica de Minas Gerais - CEFETMG/Campus Leopoldina, lecionando para o nível superior (curso de Engenharia de Controle e Automação) e para o nível técnico (cursos técnicos em eletrotécnica, mecânica e informática). Além disso, atuou na Faculdade Presidente Antônio Carlos - FUPAC/Leopoldina como professor adjunto de Cálculo I e Cálculo II para o curso de Engenharia Civil. É professor efetivo do Instituto Federal de Minas Gerais - IFMG /Campus Itabirito e atua nas seguintes linhas de pesquisa: microcontroladores e sistemas embarcados, sistemas digitais, modelos preditivos, previsão de séries temporais, inteligência computacional e lógica fuzzy.

Currículo Lattes: <http://lattes.cnpq.br/5648657901674725>.

Feito por (professor-autor)	Data	Revisão de <i>layout</i>	Data	Versão
Arthur Caio Vargas e Pinto	05/10/2022	Designado pela proex	xx/xx/xxxx	1.0

