

EXPERIMENTOS BASEADOS EM ARDUINO COMO FERRAMENTA PARA O ENSINO DE FÍSICA

Arduino-based experiments as tool for physics teaching

Carlos Augusto Cardoso Passos [carlos.passos@ufes.br]

Departamento de Física, Centro de Ciências Exatas, Universidade Federal do Espírito Santo (UFES), Campus Universitário "Alaor de Queiroz Araújo", Av. Fernando Ferrari, 514, Goiabeiras, 29075-910 Vitória - ES, Brasil

Recebido em: 08/09/2023

Aceito em: 20/12/2023

Resumo

Investigar, analisar, examinar, descobrir, interpretar e propor Modelos para fenômenos físicos são algumas das ações que um Físico desenvolve em sua prática diária. Mas para isto o Físico precisa de ferramentas adequadas para descrever a natureza. Uma possibilidade de realizar tal tarefa é por meio da Computação Física. A Física computacional é uma área de pesquisa interdisciplinar que envolve o uso de conceitos e técnicas da física e da ciência da computação. Além disso, a Computação Física engloba todas as disciplinas que permitem construir equipamentos digitais de computação que interagem com, e respondem à, realidade física analógica que os rodeia, usando software e hardware para este fim. Dentro deste universo surgiu em 2005 a placa Arduino. Esta placa de prototipação foi criada para não especialistas cujas principais características são: ser simples, barata e fácil de programar. Por causa disso, o Arduino tem “ganhado o coração” de vários professores das mais diversas áreas. Tudo isso pode tornar o processo de ensino/aprendizagem cada vez mais “mão na massa” em na linguagem popular. É um ambiente alternativo para desenvolver as competências necessárias dos alunos e dos professores. Objetivo deste trabalho é mostrar a versatilidade da placa Arduino para o Ensino de Física. Mostrar também alguns exemplos práticos de análise de fenômenos físicos relatados em artigos, dissertações de mestrado e projeto de pesquisa com a placa Arduino.

Palavras-chave: Ensino, Aprendizagem, Experimentos, Arduino, sensores.

Abstract

Investigating, analyzing, examining, discovering, interpreting, and proposing models for physical phenomena are some of the actions that a Physicist develops in his daily practice. But for this, the physicist needs adequate tools to describe nature. One possibility to accomplish such a task is through Physical Computing. Computational Physics is an interdisciplinary research area that involves the use of concepts and techniques from physics and computer science. In addition, Physical Computing encompasses all disciplines that allow building digital computing equipment that interact with and respond to the analog physical reality that surrounds them, using software and hardware for this purpose. Within this universe appeared in 2005 the Arduino board. This prototyping board was created for non-specialists whose main characteristics are being simple, cheap and easy to program. Because of this, Arduino has “won the hearts” of several teachers from the most diverse areas. All this can make the teaching/learning process increasingly “hands-on” in popular language. It is an alternative environment to develop the necessary skills of students and teachers. The objective of this work is to show the versatility of the Arduino board for Physics Teaching. Also show some practical examples of analysis of physical phenomena reported in articles, master dissertations and research projects with the Arduino board.

Keywords: Teaching, Learning, Experiments, Arduino, sensors.

1. Introdução

Nas últimas décadas se observa o desenvolvimento de metodologias ativas com a finalidade de enriquecer e melhorar o processo de ensino-aprendizagem em ciências. De modo particular, a experimentação é uma abordagem que está sendo discutida e utilizada com diferentes conjuntos de *kits* (Lourenço *et al.*, 2021). As atividades práticas promovem a curiosidade e a criatividade do estudante e permitem ao professor realizar interdisciplinaridade entre disciplinas de modo que as fronteiras entre elas não sejam visíveis, conforme argumentam Cardoso e João (2019). Assim o professor precisa discernir sobre as similaridades entre conceitos nas disciplinas (Oliveira *et al.*, 2018). Por exemplo, Silva (2018) descreveu que a disciplina de Introdução a Programação e Algoritmos tem alto índice reprovação nos cursos de Tecnologia da Informação e Computação. E, como metodologia ativa, sugeriu associar Computação Física com experimentos práticos (automação e robótica) porque é possível desenvolver habilidades de programação para resolver um problema concreto. Da mesma forma, Silva (2018) propôs utilizar a plataforma Arduino para o ensino de programação e argumentou que o ensino de programação, a partir de projetos práticos, oferece caráter lúdico, desperta a atração e o engajamento para os estudos.

Mas o que é Arduino? Do modo simplista, Arduino é uma plataforma computação física de código aberto (*open source*) baseado em uma placa microcontroladora e um ambiente integrado de desenvolvimento (*Integrated Development Environment* - IDE) para escrever o código para a placa (Arduino, 2018). Isto permite aos usuários criarem objetos eletrônicos interativos e independentes, usando o microcontrolador Atmel AVR ou ARM com suporte de entrada/saída (analogica e digital) embutido. Todos os recursos podem ser controlados pelo microcontrolador utilizando a linguagem de programação C/C++ (Kinchin, 2018). Sua versatilidade e baixo custo fazem dele uma base ideal para a construção de sensores de registro de dados ou dispositivos de controle. Na literatura há uma série de artigos e dissertações baseados na plataforma Arduino conforme relatado por Moreira *et al.* (2018) e por Monteiro *et al.* (2022).

Desde que Arduino foi lançado em 2005, diferentes versões foram e estão sendo desenvolvidas com finalidades distintas. E isto permite criar placas com funcionalidades estendidas com auxílio de *shields* (dispositivos que são diretamente plugados a placa), recursos extras, como o *Ethernet shield* que conecta a placa Arduino a internet (Mourão, 2018, p. 13). Há também uma enorme comunidade em fóruns e blogs que compartilham dúvidas e soluções para diversos códigos e diagramas de circuitos. Além disso, existem diversos cursos gratuitos em níveis introdutórios na plataforma de compartilhamento de vídeos como o “YouTube” para que iniciantes possam começar a utilizar a placa Arduino (Kinchin, 2018).

No ambiente escolar, esta ferramenta pode ser utilizada em aulas de ciências (no Ensino Fundamental, Ensino Médio e no Ensino Superior) para construir instrumentos de medidas de baixo custo que seja equivalente aos instrumentos científicos comerciais (Moreira *et al.*, 2018). De modo particular, no ensino de Física, uma das primeiras propostas em periódicos nacionais descreveu uma montagem simples para analisar um oscilador amortecido e transferência de calor por radiação (Souza *et al.*, 2011). No caso do oscilador amortecido, os autores propuseram uma montagem de um pêndulo simples com um apontador laser, resistor de 10 k Ω e um LDR conectado a placa Arduino. E no caso da radiação térmica, os autores utilizaram lâmpada incandescente de 150 W, duas superfícies metálicas pintadas uma de branco e a outra de preto, resistor de 4,7 k Ω e dois sensores temperatura NTC (*Negative Temperature Coefficient*) e utilizaram a mesma placa Arduino. Neste artigo, os autores não avaliaram como esta metodologia influencia na aprendizagem dos conceitos envolvidos.

Neste mesmo contexto, Estevão e colaboradores (2022) utilizaram Arduino para um experimento de movimento circular uniformemente variada, Cardoso e Zannin (2019) utilizaram a placa para tratar

variáveis de estado dos gases, Admiral *et al.* (2018) e Castilho *et al.* (2020) propuseram determinar a aceleração da gravidade local.

Outros autores empregaram a plataforma Arduino como instrumento para aquisição de dados em aulas prática (Barros e Dias, 2019; Caetano, 2021; Andrade, 2021) como também objeto de estudo em si, isto é, considera-se a oportunidade para o estudante desenvolver a capacidade criativa e de invenção, conforme destacam Admiral e outros (2018) e Castro e Santos (2020). Segundo Azevedo e Maltempi (2023), quando os estudantes resolvem problemas com invenções científico-tecnológica, ocorrem o desenvolvimento do raciocínio, argumentação, comunicação, representação e engajamento.

Uma das vantagens da placa Arduino é a possibilidade de trocar os sensores, usar a mesma placa e carregar o novo programa para nova função. Outra vantagem é custo/benefício. Se fizer uma pesquisa em buscadores de internet, é possível comprar os componentes e placa como aqueles da proposta de Souza e colaboradores (2011) cujo valor é da ordem de R\$155,00. Como já mencionado, é uma opção de baixo custo e acessível para maiorias de professores da área de ciências (Castro e Santos, 2020).

Neste cenário, propomos neste artigo mostrar algumas funções básicas e exemplos práticos para pessoas iniciantes em Arduino com a finalidade de desenvolver experimentos para aulas de ciências.

2. Materiais e Métodos

Em primeiro lugar, realizamos uma pesquisa para saber quais artigos nacionais em ensino de Física que utilizam o Arduino e procuramos identificar trabalhos de conclusão de curso e dissertações de mestrado onde foram desenvolvidos experimentos de Física baseados na placa microcontrolada Arduino. Para esta finalidade utilizamos Google Acadêmico, Portal CAPES e algumas revistas nacionais. As palavras-chaves foram “Arduino, microcontrolador, experiências, ensino e ensino de física”. O período de análise foi de 2005 a 2023, mas neste artigo citamos somente alguns trabalhos cuja aplicação da placa Arduino e sensores apresentam estrutura similar e sejam modelos para iniciantes. Por exemplo, Andrade (2021) propôs um curso de Arduino para iniciantes (estudantes/egressos de licenciatura em física e a formação continuada de docentes de física.). Nesta proposta a autora utilizou o simulador Tinkercad. Pode-se também citar Silva e colaboradores (2015) que desenvolveram um protótipo de estação meteorológica de código aberto como projeto de iniciação científica e com a participação de estudantes de diferentes níveis de ensino e áreas do conhecimento. Uma proposta similar foi publicada por Carvalho (2019) para medida de temperatura e umidade em reserva ecológica. Há também o produto educacional de Moura (2018) onde contém a descrição de automação de práticas experimentais abrangendo as temáticas de cinemática, dinâmica, eletricidade, hidrostática e termometria. Um outro exemplo interessante é utilizar a placa Arduino como multímetro para aulas de eletricidade como descrito por Coutinho Junior e colaboradores (2021). Estes autores indicaram utilizar tal protótipo em aulas sobre da lei dos nós da lei das malhas. Considerando estes modelos, propusemos uma sequência de exemplos práticos assistidos por computador com a utilização da placa Arduino UNO R3 e alguns sensores. Em cada exemplo tem descrito qual é a finalidade do programa (a funcionalidade da placa Arduino), o algoritmo para execução e a explicação dos principais comandos em cada programa. Os exemplos estão organizados por utilização da Serial Monitor da IDE, a placa Arduino na função multímetro e sensores que utilizam portas analógicas e portas digitais. Todos os programas foram testados em simuladores Tinkercad (<https://www.tinkercad.com/>) e Wokii (<https://wokwi.com/>). Estes dois *sites* são recomendados para iniciantes.

3. Exemplos Práticos

Exemplo 1: Imprimir na tela do computador uma frase.

Como primeiro programa, vamos imprimir na tela do computador a frase “Ola Mundo!”. Para isto conecte a placa Arduino ao computador. Só é possível testar o código (*sketch*) se a placa estiver conectada ao computador. Digite o seguinte *sketch* na IDE do Arduino:

```
// Primeiro programa
void setup( ){
  Serial.begin(9600); };           // Habilita a Serial Monitor
  Serial.print(“Ola Mundo!”); };   // Imprime a frase na Serial Monitor
}
void loop( ){ }
```

E transfira o programa para a placa através do botão *Upload*. Este processo faz a compilação (leitura e verificação dos comandos) que gera o código binário e, em seguida, transfere o programa para placa. Por fim, clique no botão *Serial Monitor* para abrir a janela de terminal da IDE. Estas linhas de código imprime uma única vez na tela do computador. É importante ressaltar que todos os comandos devem estar dentro de chaves “{ }” e finalizado com ponto e vírgula “;”. Para que a mensagem se repita é necessário colocar a mensagem dentro do ambiente da função `void loop()`, como se segue:

```
// Variação do primeiro programa
void setup( ){
  Serial.begin(9600); }
void loop( ){
  Serial.println("Ola Mundo!");
  delay(1000); }
```

Aqui o comando “//” significa comentário, o compilador não considera qualquer escrita na linha após dupla barra. Além disso, o comando “Serial.begin(9600)” inicializa a porta serial e permite visualizar “dados” na tela do computador quando há o comando “Serial.print”. Já o comando “Serial.println” serve também para imprimir e para pular linhas comumente utilizado dentro da função `loop`. Neste comando, aquilo que está entre aspas duplas é uma variável tipo dado *String* que serve para conjunto de caracteres. Podemos também usar aspas simples. O comando `delay(1000)` serve para aguardar 1000 ms (ou 1 s) para repetir a operação, ou seja, neste primeiro programa a mensagem “Ola mundo” é repetida após um segundo. É importante ressaltar que as funções `void setup` e `void loop` são obrigatórias. Por isso que na primeira parte o exemplo está escrito a função `void loop`.

3.1 Divisor de Tensão

É possível usar a placa Arduino na função ohmímetro DC usando qualquer uma das portas analógicas. Para esta finalidade, devemos usar um **divisor de tensão** como ilustrado na Figura 1. O resistor R_1 deve ser conectado ao pino Vcc da placa Arduino (ponto A) e ao resistor R_2 . O ponto comum dos resistores (ponto B) deve ser conectado a uma porta analógica do Arduino. O outro lado resistor R_2 deve ser conectado ao pino GND da placa Arduino.

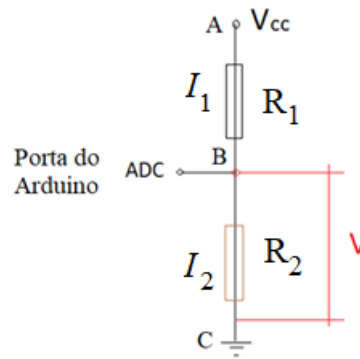


Figura 1: Circuito de resistores em série que neste exemplo é utilizado para medir a resistência elétrica de um resistor desconhecido R_2 . A tensão de alimentação V_{cc} é igual a 5 V.

Nossa hipótese é que a corrente elétrica é zero entre a porta analógica do Arduino e o ponto B. Com esta suposição, consideramos que a corrente elétrica é a mesma ($I_1 = I_2$) sobre os resistores R_1 e R_2 e isto significa que podemos afirmar que resistores estão em série. Portanto, para medir a resistência elétrica R_2 , é necessário saber qual é a corrente elétrica entre o ponto A e o ponto C, para isto utilizamos a lei de Ohm:

$$I = \frac{V_{cc}}{R_1 + R_2}, \quad (1)$$

onde V_{cc} é a tensão fornecida pelo Arduino (5 V), R_1 é uma resistência elétrica conhecida e igual a $10 \text{ k}\Omega$. A tensão elétrica medida na porta analógica do Arduino é

$$V = R_2 \cdot I. \quad (2)$$

Combinando estas equações, temos

$$R_2 = \frac{V}{(V_{cc} - V)} R_1. \quad (3)$$

Todo este desenvolvimento pode ser realizado pelo estudante para ser implementado no código durante a programação. Este é exemplo onde os cálculos matemáticos podem ser explorados para aprofundamento da teoria e fazer previsão. O divisor de tensão é um esquema de montagem muito útil para diversos circuitos (ver próximos exemplos).

Exemplo 2: A placa Arduino como ohmímetro.

Em uma entrada analógica da placa Arduino, a tensão elétrica pode variar entre 0 e 5 V dentro do mapeamento de dados inteiros que estão no intervalo de 0 a 1023. Isto permite ainda que criemos uma relação entre a variação de tensão nos terminais e as leituras. A função própria para leitura da porta analógica é `analogRead(porta)`. Para este exemplo o esquema do circuito é dado pela Figura 2.

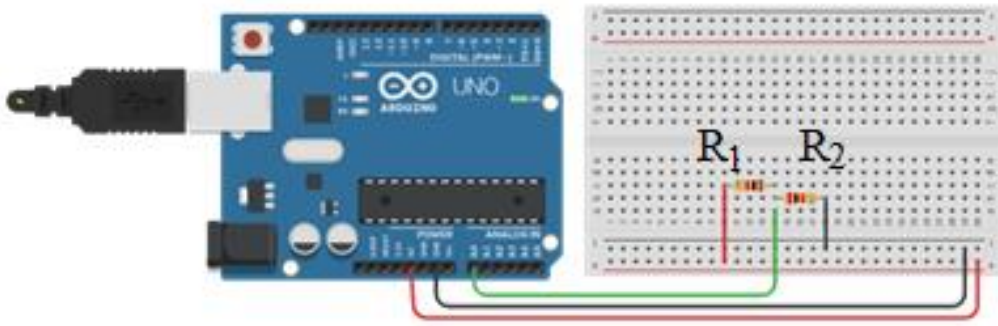


Figura 2: Esquema de montagem para a placa Arduino ser utilizada na função ohmímetro. A resistência elétrica R_1 é conhecida e a resistência elétrica R_2 é aquela a ser determinada.

O programa (*sketch*) para determinar a resistência elétrica de um resistor qualquer é:

```
// Arduino na função Ohmímetro
// ----- Declaração de variáveis -----
float R1 = 10000;           // Valor da resistência 1 medido com um multímetro
float R2;                  // Resistência a ser determinada
float tensao, Vcc = 5.0;
int pinoResistor = A0;
//----- Início do Programa -----
void setup() {
  Serial.begin(9600);      // Habilita a Serial Monitor
  pinMode(pinoResistor, INPUT); // Configuração da porta A0 como entrada.
}
// ----- Estrutura de repetição -----
void loop() {
  float valorLeitura = analogRead(pinoResistor); //Leitura da porta e guarda na variável.
  tensao = valorLeitura*Vcc/1024;                //Converter leitura da porta para valores de 0 a 5V
  R2 = (R1*tensao)/(Vcc - tensao);               //Cálculo da resistência
  Serial.println(R2); //Mostra o valor da resistência R2 na tela do computador (monitor serial).
  delay(100);
}
```

Observe que as linhas de comando estão indentadas. Isto facilita a revisão de todo o *sketch* se algo estiver errado. Além disso, a linguagem de programação é *case sensitive*, ou seja, há diferença entre letras maiúsculas e minúsculas.

Exemplo 3: A placa Arduino como voltímetro.

A placa Arduino para ser um voltímetro, vamos utilizar dois resistores conhecidos (R_1 e R_2) e utilizar o mesmo esquema da Figura 1 e a mesma montagem da Figura 2. Para medir a tensão elétrica sobre o resistor R_2 . E da mesma forma, a tensão elétrica pode variar entre 0 e 5 V dentro do mapeamento de dados inteiros que estão no intervalo de 0 a 1023. O programa (*sketch*) para determinar a tensão elétrica sobre o resistor R_2 é:

```
// Arduino na função Multímetro
// ----- Declaração de variáveis -----
float R1 = 1000000;        // Valor da resistência 1 medido com um multímetro
float R2;                  // Resistência a ser determinada
float tensao, Vcc = 5.0;
int pinoResistor = A0;
```



```
//----- Início do Programa -----
void setup() {
  Serial.begin(9600);          // Habilita a Serial Monitor
  pinMode(pinoResistor, INPUT); // Configuração da porta A0 como entrada.
}
//
void loop() {
  float valorLeitura = analogRead(pinoResistor); //Leitura da porta e guarda na variável.
  tensao = valorLeitura*Vcc/1024;                //Converter leitura da porta para valores de 0 a 5V.
  Serial.println(tensao);                        //Mostra o valor da variável na tela do computador (monitor serial).
  delay(100);
}
```

Se Combinarmos as equações 1 e 2, temos:

$$V = \frac{R_2}{(R_1+R_2)} V_{cc}. \quad (4)$$

Esta expressão pode ser utilizada para checar o valor obtido pelo Arduino. Observe que a razão entre os resistores é menor do 1 que significa que a tensão elétrica V é menor do que a tensão de entrada V_{cc} . Daí vem o nome divisor de tensão. Como podemos notar, um circuito elétrico simples e a montagem de um instrumento de medida tem riqueza de contexto.

Nos próximos exemplos, vamos utilizar divisor de tensão para obter informações de sensores.

Exemplo 4: Sensor LDR.

Para obter informações do sensor LDR, deve-se ligá-lo a uma porta analógica uma vez que este componente varia sua resistência em função da luminosidade do local. Para este exemplo o esquema do circuito é dado pela Figura 3 e estratégia para leitura da grandeza física é usar uma resistência elétrica conhecida R_1 .

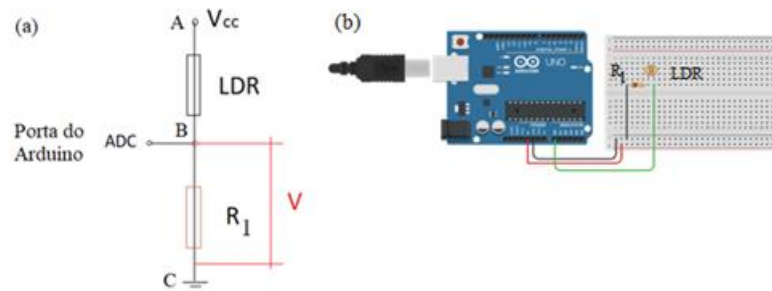


Figura 3: Esquema de ligação do sensor LDR. (a) No circuito, a porta analógica vai variar a tensão elétrica, ou seja, o Arduino opera na função voltímetro. (b) As ligações do sensor no Arduino devem ter a porta analógica, GND e Vcc onde o resistor é ligado ao sensor e ao pino GND.

```
// ----- Declaração de Variáveis -----
int pinLdr = A0;           // variável tipo inteiro
float valorLdr;           // variável tipo float (real) para armazenar
float R1 = 47000.0;       // Resistência elétrica conhecida.
float Rsensor;            // Resistência elétrica do Sensor LDR
float Vcc = 5.0;          // Tensão fornecida pelo Arduino
```

```

float tensao; // Tensão que será lida na porta analógica

// ----- Início do Programa -----
void setup()
{
  Serial.begin(9600); // Habilita Monitor Serial com taxa de 9600 bits/s.
  pinMode(pinLdr, INPUT); // Configura porta analógica onde o sensor está conectado.
}
// ----- Leitura da Porta Analógica -----
void loop()
{
  valorLdr = analogRead(pinLdr); //Leitura da porta e guarda na variável.
  tensao = valorLdr*Vcc/1024; //Converter leitura da porta para valores de 0 a 5V
  Rsensor = (R1/ tensao)*(Vcc - tensao); //Cálculo da resistência do LDR

  Serial.println(Rsensor); //Mostra o valor da variável na tela do computador.
  delay(300); //Aguarda por 300 ms para próxima leitura.
}

```

Nos próximos exemplo iremos somente trocar o sensor, mas o esquema de ligação é o mesmo e código do programa é muito similar. Usamos o mesmo conceito no esquema da Figura 3a.

Exemplo 5: Sensor de Temperatura NTC 10 kΩ.

Para fazer a leitura de temperatura, vamos utilizar componente Termistor. Este é um tipo de resistência elétrica especial que altera seu valor em função da temperatura. Neste exemplo vamos utilizar o Termistor NTC (*Negative Temperature Coefficient*) e a mesma estrutura do Exemplo 2. Além disso, vamos usar a equação de Steinhart-Hart para converter o valor da resistência elétrica em valores de temperatura:

$$\frac{1}{T} = \frac{1}{T_o} + \frac{1}{\beta} \log \left(\frac{R}{R_o} \right), \quad (5)$$

onde R é resistência elétrica do termistor em dada temperatura, T_o é o valor da temperatura ambiente (em Kelvin), R_o é o valor da resistência do termistor em T_o e β é uma constante fornecida pelo fabricante (denominado coeficiente beta do termistor). Para facilitar a escrita do *sketch*, vamos definir $R = R_2$ para aproveitar a estrutura do exemplo, como já mencionado e reescrever a eq. (5):

$$T = \frac{1}{\frac{1}{T_o} + \frac{1}{\beta} \log \left(\frac{R_2}{R_o} \right)}. \quad (6)$$

Vamos substituir R_2 (eq. (3)) na eq. (6) tal que

$$T = \frac{1}{\frac{1}{T_o} + \frac{1}{\beta} \log \left[\frac{\frac{V}{(V_{CC}-V)^{R_1}}}{R_o} \right]}, \quad (7)$$

Para nosso esquema de montagem vamos escolher $R_1 = R_o = 10 \text{ k}\Omega$. Lembrando que no circuito R_1 está em série com R_2 . Assim,

$$T = \frac{1}{\frac{1}{T_0} + \frac{1}{\beta} \log \left[\frac{1}{\left(\frac{V_{CC}}{C} - 1 \right)} \right]} \quad (8)$$

Uma vez que V é a tensão lida na porta analógica e quando se faz a conversão para volts o argumento do logaritmo simplifica para

$$T = \frac{1}{\frac{1}{T_0} + \frac{1}{\beta} \log \left[\frac{1}{\left(\frac{1024}{\text{leitura na porta analógica}} - 1 \right)} \right]}$$

O circuito está descrito na Figura 4. Observe que mantivemos a ligação dos fios ao Arduino e somente trocamos o sensor, ver Figura 2.

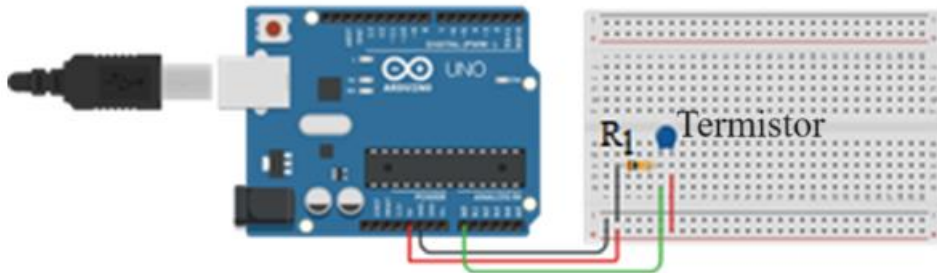


Figura 4: Esquema de montagem para medir temperatura com sensor de termistor NTC. Nesta montagem, o resistor $R_1 = 10 \text{ k}\Omega$.

```
// Medida de temperatura com Termistor NTC 10K com Arduino na função Ohmímetro
//----- Declaração de Variáveis -----
int pinoThermistor = A0;           // ligação do sensor a porta analógica
float tensao,                       //leitura da porta analógica
    R2,
    TempK,                           // temperatura em Kelvin
    TempF,                           // temperatura em Fahrenheit
    TempC;                           // temperatura em Celcius
const float beta = 3950.0;         //valor fornecido pelo fabricante (ver Datasheet)
const float To = 298.15;          //temperatura ambiente em Kelvin
const float Ro = 10000.0;         // resistência do termistor em To
const float R1 = 10000.0;         // Resistência conhecida

void setup() {
  Serial.begin(9600);
}

void loop() {
  int analogValue = analogRead(A0);
  // Cálculo da temperatura utilizando a equação de Steinhart-Hart
  TempK = 1 / ((1.0 / To) + log(1 / (1023. / analogValue - 1)) / beta);
  TempC = TempK - 273.15;
  Serial.print("Temperature: ");
  Serial.print(TempC);
  Serial.println(" C");
  delay(1000);
}
```

Sensor temperatura termistor pode ser encapsulado para realizar medidas em líquidos. Algumas aplicações estão descritas no Exemplo 8. Este exemplo é um exercício que pode ser realizado no Ensino Médio e Ensino Superior.

Exemplo 6: Sensor de Flexão.

O sensor de flexão pode ser utilizado como protótipo de prótese robótica empregado em luvas e roupas para reproduzir em tempo real movimentos específicos e controlados (Alves, 2018). O sensor de flexão tem como princípio varia a sua resistência elétrica quando é deformado cuja faixa de curvatura está entre 45 k Ω a 125 k Ω . Para este sensor de flexão, usamos o código:

```
//----- Declaração de Variáveis -----
int flexSensorPin = A0;           //Pino onde o sensor está ligado
//-----
void setup(){
  Serial.begin(9600);
}
//-----
void loop(){
  int flexSensor = analogRead(flexSensorPin);           // Valor do sensor de flexão
  Serial.println(flexSensor);
  delay(2000);           //Aguardar 2000 ms (2 s) para próxima medida
}
```

Neste exemplo não estamos interessado no valor específico para resistência elétrica ou tensão elétrica. A montagem está ilustrada na Figura 5.

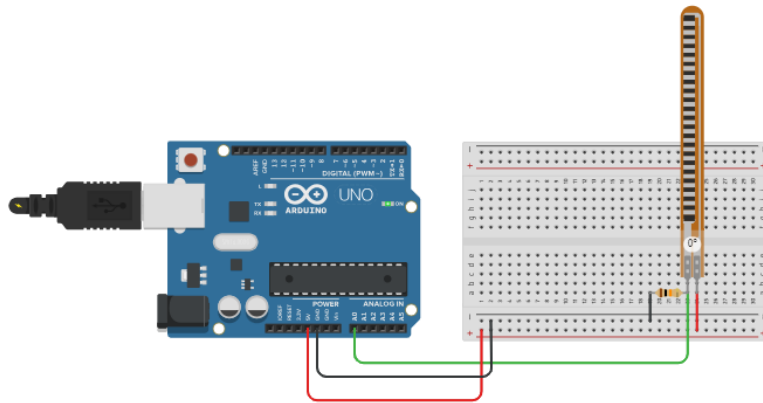


Figura 5: Esquema para utilizar o Sensor de Flexão. A resistência R_1 é igual a 10 k Ω .

Exemplo 7: Sensor de Força Resistiva

O sensor de força resistivo é capaz de fazer medidas de massas quando a área do sensor é pressionada. Quanto maior a for pressão no sensor, menor será a resistência medida. Neste exemplo, o valor lido não está convertido em unidades de massa. A montagem está descrita na Figura 6. O *sketch* é como está descrito a seguir.

```
//----- Declaração de Variáveis -----
```

```

int flexSensorPin = A0; //Pino onde o sensor está ligado
//-----
void setup(){
  Serial.begin(9600);
}
//-----
void loop(){
  int flexSensor = analogRead(flexSensorPin); //Valor do sensor de flexão.
  Serial.println(flexSensor);
  delay(2000); //Aguardar 2000 ms (2 s) para próxima medida.
}

```

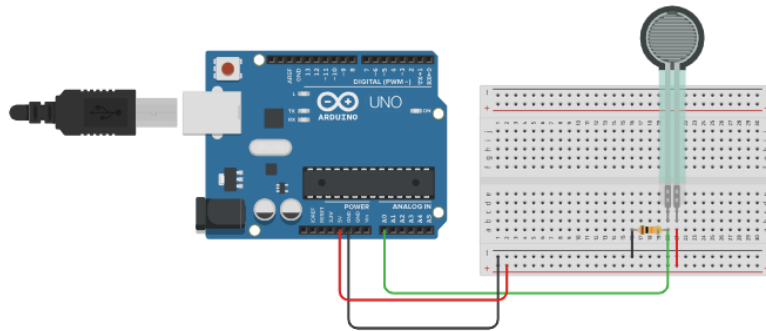


Figura 6: Esquema para utilizar o Sensor de Força Resistivo. No circuito foi utilizado um resistor de R_1 igual a $10\text{ k}\Omega$.

Estes códigos exemplificam que o esquema de ligação é o mesmo, o que modifica é somente o sensor que é utilizado. Além disso, é necessário conhecer o sensor para fazer a conversão de valores para obter a grandeza física investigada.

Exemplo 8: Sensor de Temperatura LM35.

O sensor de temperatura LM35 é um circuito integrado muito utilizado em pequenos projetos e funciona na faixa de -55 a $155\text{ }^\circ\text{C}$ e é linear com uma taxa de $10\text{ mV}/^\circ\text{C}$. A montagem deste exemplo não necessita de um divisor de tensão (ver Figura 4). A leitura do sensor fornece um número entre 0 e 1023 que correspondem a 0V e 5V. Além disso, 10 mV equivale $1\text{ }^\circ\text{C}$, valor de temperatura é dado por:

$$temperatura = leitura * 5V / ((1024 * 0,010(V/^\circ\text{C})))$$

A montagem está ilustrada na Figura 7.

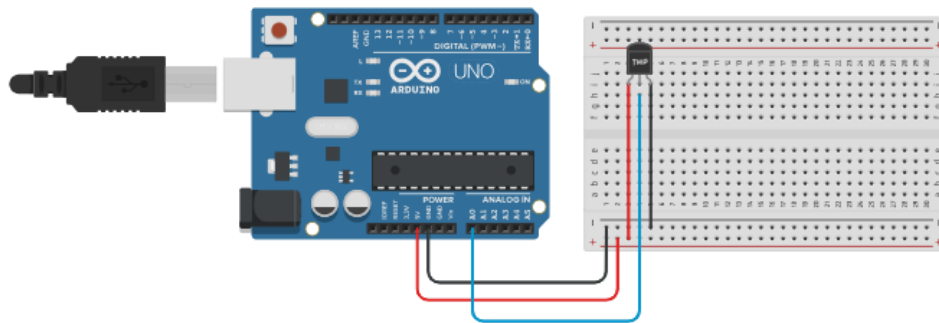


Figura 7: Esquema de montagem para o sensor LM35. Nesta montagem não é necessário um resistor.

```
//----- Declaração de Variáveis -----
int sensorTemp = A0; //Pino do sensor conectado à porta A0
float temperatura;
//-----
void setup(){
  Serial.begin(9600);
  pinMode(sensorTemp, INPUT);
}
//-----
void loop(){
  int leitura = analogRead(sensorTemp); // leitura do sensor de temperatura
  temperatura = leitura*5.0 /(1024*0.010); //Conversão do valor de tensão em temperatura.
  Serial.println(temperatura);
  delay(1000); //Aguardar 1000 ms (1 s) para próxima medida
}
//-----
```

O monitor serial da IDE do Arduino permite habilitar o relógio do computador. Assim as leituras da temperatura podem ser visualizadas no horário local. Depois de finalizar a medida e antes de fechar a monitor serial da IDE, basta copiar os dados e inserir em uma planilha eletrônica.

Este código pode ser aplicado para investigar sistemas termodinâmicos, citamos três exemplos:

1. Processo de aquecimento de objetos sólidos com materiais diferentes composições (ferro, alumínio, etc.);
2. Transferência de Calor por Condução e por Radiação;
3. Estudar a lei de resfriamento de Newton (Tufaile e Tufaile, 2022).

De modo específico, para avaliar a mudança de temperatura de duas latas de alumínio de diferentes cores (pintada um de cor branco e outra de cor preto), o código é similar ao caso anterior (Monteiro, 2017, p. 66). Vejamos:

```
//----- Declaração de Variáveis -----
int sensorTemp1 = A0; //Pino do sensor conectado a porta A0 e a lata de alumínio preto
int sensorTemp2 = A1; //Pino do sensor conectado a porta A0 e a lata de alumínio branco
//-----
void setup(){
  Serial.begin(9600);
  pinMode(sensorTemp1, INPUT);
  pinMode(sensorTemp2, INPUT);
}
```

```

}
//-----
void loop(){
  int leitura1 = analogRead(sensorTemp1);    // leitura do sensor de temperatura 1
  int leitura2 = analogRead(sensorTemp2);    // leitura do sensor de temperatura 2
  float temp1 = leitura1*5.0/(1024*0.010); //Conversão do valor de tensão em temperatura
  float temp2 = leitura2*5.0/(1024*0.010); //Conversão do valor de tensão em temperatura
  Serial.print(temp1);
  Serial.print(", ");
  Serial.println(temp2);
  delay(1000);                               //Aguardar 1000 ms (1 s) para próxima medida.
}

```

É importante salientar que o sensor LM35 não é à prova d'água. Para realizar medidas em líquidos é necessário fazer encapsulamento metálico (capsula de proteção em aço inox 304).

3.2 Sensores conectados a porta digital na placa Arduino

Em uma placa Arduino UNO R3 possui 14 pinos configurados como entrada ou saídas digitais conforme a necessidade de seu projeto e estão numerados de 0 a 13. Estas portas são caracterizadas por dois estados: ligado e desligado. O modo de configuração é INPUT, OUTPUT e INPUT_PULLUP (entrada digital com resistor de pull-up, que está ligado ao Vcc, interno habilitado). Coloca-se uma sintaxe do tipo `digitalWrite(pino, valor)` para ligar ou desligar algo. Esta função “digitalWrite” só aceita nível lógico alto (HIGH ou 1) ou nível lógico baixo (LOW ou 0) em um pino configurado como saída digital. Como exemplo, vamos simular o acendimento de uma lâmpada em função da luminosidade.

Exemplo 9: Ascender um LED usando a variação de resistência do LDR.

Neste exemplo desejamos usar a montagem do Exemplo 4 (LDR) para ascender um LED. O esquema de ligação está resumido na Figura 8.

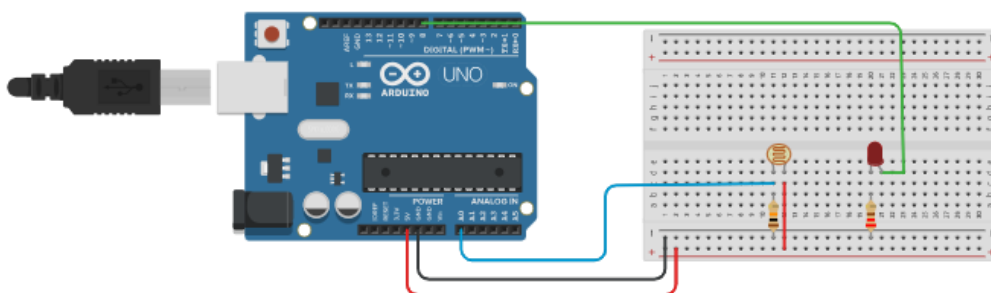


Figura 8: Esquema de ligação para acender um LED a partir da leitura de valores do sensor LDR. O resistor do LDR é 10 kΩ e o resistor do LED é de 220Ω.

Vamos aproveitar a estrutura do Exemplo 4. Para acender o LED precisa saber o nível de luminosidade do ambiente. Então, definimos um valor lido do sensor LDR que se deve acender ou apagar o LED. Isto é, na lógica de programação precisa ter uma condição, se a leitura do LDR for menor do que um número x , acenda o LED; caso contrário deixe apagado. Para isto é necessário na lógica de programação inserir a função “if” e “else”. A condição deve estar entre parênteses, como descrito a seguir.

```
// ----- Definições -----
int pinLdr = A0; //Define pino analógico para o LDR
#define pinLed 8 //Define pino digita para o LED

// ----- Declaração de Variáveis -----
float valorLdr; // variável tipo float (real) para armazenar
float R = 10000.0; // Resistência elétrica conhecida.

// ----- Início do Programa -----
void setup()
{
  Serial.begin(9600); // Habilita Monitor Serial com taxa de 9600 bits/s.
  pinMode(pinLdr, INPUT); // Configura porta analógica onde o sensor está
  conectado.
  pinMode(pinLed, OUTPUT);
}
// ----- Leitura do sensor LDR -----
void loop()
{
  valorLdr = analogRead(pinLdr); //Leitura da porta e guarda na variável valorLdr.
  Serial.println(valorLdr);
//Estrutura de condição para ascender o LED. O número que está na condição deve ser testado para o
// funcionamento do projeto.
  if (valorLdr <= 500){ //Comparação, se for verdadeiro ascende o LED.
    digitalWrite(pinLed, HIGH);
  }
  else{ //Comparação, se for falso apague o LED.
    digitalWrite(pinLed, LOW);
  }
  delay(500); //Aguardar por 0,5 s para próxima leitura.
}
```

Este tipo de exemplo em sala de aula pode ser permeado por debate acerca do resistor de proteção do LED, pode se medir a diferença de potencial sobre LED e a corrente elétrica necessária para ascendê-lo. Pode-se trocar a cor do LED e verificar quais mudanças ocorrem nas grandezas físicas. Outra possibilidade é solicitar aos estudantes que escrevam o algoritmo para que o circuito com Arduino funcione.

Exemplo 10: Sensor de temperatura e umidade DHT11 ou DHT22.

Estes sensores são opções de baixo custo para montar uma miniestação meteorológica. Estes sensores permitem medir somente temperatura e umidade relativa do ar. No caso do DHT11, o sensor pode medir temperaturas entre 0 a 50° Celsius com uma precisão de 2 °C, e umidade entre 20 a 90 % com uma precisão de 5%. É importante usar um resistor entre 4,7 k a 10 kΩ ligando o pino de saída de dados ao pino de alimentação (Vcc). Já o DHT22 mede temperatura na faixa de - 40 a 80 °C com uma precisão de 0,5 graus e a medida de umidade entre 0 e 100% com precisão de 2%. É também importante usar um resistor entre 4,7 k a 10 kΩ ligando o pino de saída de dados ao pino de alimentação (Vcc). O esquema de medida está ilustrado na Figura 9 e observe que este sensor precisa de alimentação de 5V.

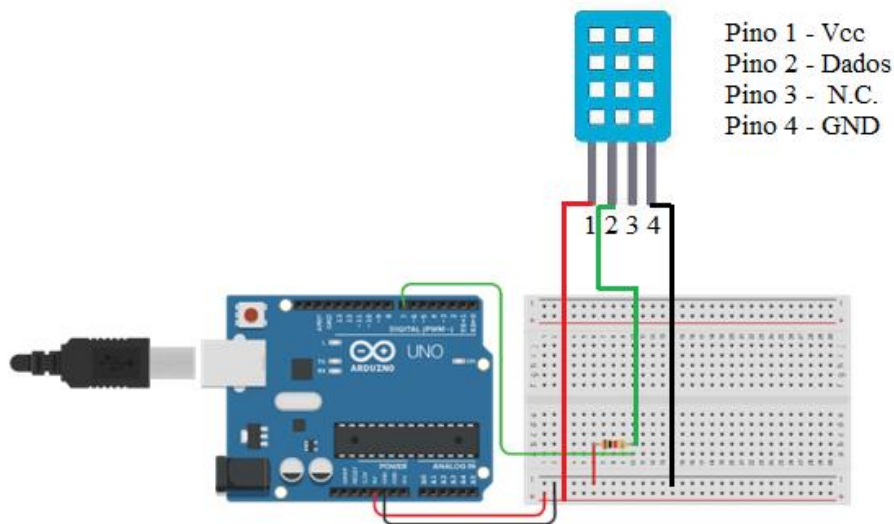


Figura 9: Esquema de montagem para o sensor de DHT11 (ou DHT22). Usamos neste circuito um resistor de 4,7 k Ω entre o pino de dados e o pino Vcc.

O código para as leituras da temperatura e da umidade através do sensor DHT11 é:

```
// Miniestação Meterológica
#include "DHT.h"          // Biblioteca para o sensor DHT
#define DHTPIN 2         // Pino digital conectado ao sensor
#define DHTTYPE DHT11    // Senosr DHT 11
// #define DHTTYPE DHT22 // Usar se o sensor for DHT 22 (AM2302), AM2321
// Inicializar o sensor DHT.
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  // Leitura da temperatura e da umidade a cada 2 segundos
  float h = dht.readHumidity();    // Leitura da umidade
  float t = dht.readTemperature(); // Leitura da temperatura em Celsius (padrão)
  float f = dht.readTemperature(true); // Leitura da temperatura em Fahrenheit
  // Checar se qualquer leitura falhou e saia do looping.
  // Isto é bom para checar se as conexões estão certas.
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
  // Enviar os dados para Serial Monitor
  Serial.print(F("Umidade: "));
  Serial.print(h);
  Serial.print(F("% Temperatura: "));
  Serial.print(t);
```

```
Serial.print(F("C "));  
Serial.print(f);  
Serial.println(F("F"));  
delay(2000);  
}
```

Para utilizar o sensor DHT22, é necessário comentar a linha do DHT11 (3ª. linha do programa) e retirar a dupla barra da linha seguinte.

Neste código optamos por usar definições em vez variáveis para definir o pino de ligação. Para usar definição basta escrever `#define + pino`. Isto economiza memória no Arduino e variável ocupa espaço de memória no microcontrolador. Utilizamos também uma biblioteca para o sensor com o comando `#include "DHT.h"`. Em geral, usuários do Arduino simplificam o código com a utilização de bibliotecas e normalmente são criadas por pessoas com nível avançado de programação. A biblioteca que testemos foi obtida do site da Adafruit Industries.

Este sensor é bastante simples de montar e usar. Neste exemplo usamos um resistor entre o pino de dados e o pino Vcc para estabilizar a leitura pelo Arduino. O sensor requer cuidado com o tempo entre leituras consecutivas, uma vez que é necessário um intervalo de, no mínimo, 1 segundo entre uma leitura e outra. O pino 3 não foi utilizado.

4. Considerações Finais

Ao longo de várias décadas, a pesquisa em Ensino de Física está apresentando diferentes abordagens com a finalidade de melhorar o processo de ensino-aprendizagem. Entre as principais propostas, as metodologias ativas têm sido utilizadas como uma das principais linhas de pesquisa brasileira em Ensino de Física. De modo particular, propusemos a utilização didática do computador e microcontroladores com ferramenta adicional às aulas práticas em laboratórios. Levando isto em consideração, a proposta deste artigo foi indicar caminhos para construir equipamentos didáticos e de baixo custo para aquisição automática de dados baseia-se na lógica digital e analógica da porta de jogos, porta serial, porta paralela e nas portas USB (Porta Serial Universal, em português) dos computadores pessoais e notebooks. Apresentamos sistemas de aquisição de dados baseado na placa microcontrolada Arduino e sensores.

Os exemplos descritos neste trabalho permitem o professor trabalhar diversos conceitos de física e testar teorias ou leis. A plataforma Arduino é muito versátil, é de baixo custo e, também, pode ser utilizada com propósito de desenvolvimento de experimentos didáticos que permitam um ensino e uma aprendizagem de Física mais significativos.

Agradecimentos

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Ensino Superior (CAPES) e a Fundação de Apoio à Pesquisa do Espírito Santo (FAPES) pelas bolsas concedidas aos mestrados.

Referências

Admiral, T. D.; Júnior, E. R.; & Linhares, M. P. Utilização de Arduino como motivador no ensino de física para alunos de graduação em matemática. *Revista Espacios*, v. 39, n. 52, p. 15, 2018.

Andrade, A. de. Um estudo das Representações sociais sobre a automatização da coleta de dados no laboratório didático de física durante a formação docente. 205 f. Dissertação (Mestrado). Programa

de Pós-Graduação em Ensino de Física - Mestrado Profissional em Ensino em Ciências e Matemática, Instituto Federal de Educação, Ciência e Tecnologia. São Paulo, Cariacica, ES, 2021.

Andrade A. de. Um estudo das Representações Sociais sobre a automatização da coleta de dados no laboratório didático de física durante a formação docente. Dissertação (Mestrado). 205f. Programa de Pós-Graduação em Ensino de Ciências e Matemática) - Mestrado Profissional em Ensino de Física, Instituto Federal de São Paulo, São Paulo, SP, 2021. Pág. 118 – 156.

Alves, F. A. S. Protótipo de prótese robótica de punho e mão utilizando Arduino. Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Sistemas de Informação da Unidade Acadêmica de Serra Talhada da Universidade. 2018. 64f.

Adafruit Industries. Biblioteca para o sensor DHT. <https://github.com/adafruit/DHT-sensor-library>. Acesso em 27 ago. 2023.

Arduino. Introdução ao Arduino (Última Versão). Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Acesso em 27 ago. 2023.

Azevedo, G. T.; & Maltempi, M. V. Formação em Matemática por meio de invenções científico-tecnológicas voltadas aos impactos sociais: uma revisão de literatura. *Revista Baiana de Educação Matemática*, v. 4, n. 1, e202302 (17p.) 2023. <https://doi.org/10.47207/rbem.v4i01.15487>.

Barros, T. R.; & Dias, W. S. Práticas experimentais de Física a distância: Desenvolvimento de uma aplicação com Arduino para a realização do Experimento de Millikan remotamente. *Revista Brasileira de Ensino de Física*, vol. 41, nº 4, e20190049 (8p.) 2019. <http://dx.doi.org/10.1590/1806-9126-RBEF-2019-0049>.

Caetano, T. C. O experimento “curva de luz” do Laboratório Remoto de Física: uma proposta de atividade investigativa contextualizada epistemologicamente. *Revista Brasileira de Ensino de Física*, vol. 43, e20210169 (14p.) 2021. <https://doi.org/10.1590/1806-9126-RBEF-2021-0169>.

Cardoso, J. M.; & João, J. J. Contextualização e Experimentação: Uma Abordagem Interdisciplinar de Química e Física Utilizando Experimentos de Simulação de um Motor a Vapor. *Revista Virtual Química*, v. 11, n.1, p. 339-352, 2019. <http://dx.doi.org/10.21577/1984-6835.20190024>.

Cardoso, J. M.; & Zannin, M. Proposta experimental para análise das variáveis de estado dos gases com Arduino *Revista Brasileira de Ensino de Física*, v. 41, n. 4, e20190028 (6p) 2019). <http://dx.doi.org/10.1590/1806-9126-RBEF-2019-0028>.

Carvalho, E. S. Papel dos traços florais na atração de abelhas da restinga. 42f. Dissertação (Mestrado). Programa de Pós-Graduação em Ciências Biológicas. Universidade Federal do Espírito Santo, Vitória, ES, 2019, p. 12-17. Castilho W. S.; Lima, D.; & Dutra, M. V. G. O ensino de física e a aprendizagem significativa: Um kit experimental com Arduino para o ensino de queda livre. *Experiências em Ensino de Ciências*. v.15, n.3, p. 247-262, 2020.

Castro, L. H. M. de; & Santos, R. dos. O uso do Arduino e acriação de objetos educacionais em tempos e espaços desarticulados. *Revista de Ciências da Computação*. v. 2, n. 2, p. 5-12, 2020. <https://doi.org/10.22481/recic.v2i1.6550>.

COUTINHO JÚNIOR, A. de L. .; MONTEIRO, J. A.; COSTA, D. F. da .; SALES, G. L. . An experimental proposal for electricity using the Arduino prototyping board for teaching physics. *Research, Society and Development*, v. 10, n. 2, p. e11110212302, 2021. <http://dx.doi.org/10.33448/rsd-v10i2.12302>.

- Estevão, H. V.; Samento, R. G.; Frazão, N. F.; & Madeira, M. de P. Produção de experimento didático com Arduino para o estudo do movimento circular uniformemente variado. *A Física na Escola*, v. 20, n. 1, 210505 (5p.) 2022.
- Kinchin, J. Using an Arduino in physics teaching for beginners. *Physics Education*, v. 53, n. 6, 063007 (3pp), 2018.
- Lourenço, R. W. de; Alves, J. G. S.; & Silva, A. Pa. R. da. Por uma aprendizagem significativa: metodologias ativas para experimentação nas aulas de ciências e química no Ensino Fundamental II e Médio. *Brazilian Journal of Development*, v.7, n.4, p. 35037-35045, 2021. <https://doi.org/10.34117/bjdv7n4-311>.
- Monteiro, L. P. Concepções da dualidade da luz onda-partícula para estudantes da 3ª. série do ensino médio: uma abordagem com “microcontroladores”. 173 f. Dissertação (Mestrado). Programa de Pós-Graduação em Ensino de Física - Mestrado Profissional em Ensino de Física, Instituto Federal do Espírito Santo, Cariacica, ES, 2017.
- Monteiro, J. A.; Vilhena, M. F.; Silva, F. H. S. da; Lucena, I. C. R. de; & Júnior, A. L. C. Arduino no Ensino de Física: uma Revisão Sistemática de Literatura de 2011 a 2021. *Revista de Educação em Ciências e Matemática*, v.18, n. 40 p. 177-190, 2022. <http://dx.doi.org/10.18542/amazrecm.v18i40.12175>.
- Moreira, M. M. P. C.; Romeu, M. C.; Alves, F. R. V.; & Silva, F. R. O. Contribuições do Arduino no ensino de Física: uma revisão sistemática de publicações na área do ensino. *Caderno Brasileiro de Ensino de Física*, v. 35, n. 3, p. 721-745, 2018. <http://dx.doi.org/10.5007/2175-7941.2018v35n3p721>.
- Mourão, O. Arduino & Ensino de Física: Automação de Práticas Experimentais. Tianguá: Clube dos Autores, 2018, 166 f.
- Oliviera, H. G.; Antonello, R.; Fidélis, A. J.; & Rinaldi, B. J. D. Energia, Sociedade e Meio Ambiente no Desenvolvimento de Um Biodigestor: a Interdisciplinaridade e a Tecnologia Arduino para Atividades Investigativas. *Química Nova na Escola*. v. 40, n 3, p. 144-152, 2018. <http://dx.doi.org/10.21577/0104-8899.20160127>.
- Silva, R. B.; Leal, L. S.; Alves, L. S.; Brandão, R. V.; Alves, R. C. M.; Klering, E. V.; & Pezzi, R. P. Estações meteorológicas de código aberto: um projeto de pesquisa e desenvolvimento tecnológico. *Revista Brasileira de Ensino de Física*, São Paulo, v. 37, n. 1, p. 1505-1 – 1505-8, 2015. <https://doi.org/10.1590/S1806-11173711685>.
- Souza, A. R. de; Paixão, A. C.; Uzêda, D. D.; Dias, M. A.; Duarte, S.; & Amorim, H. S. de. A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC. *Revista Brasileira de Ensino de Física*, v. 33, n. 1, p. 1702-1 – 1702-5, 2011.
- Tufaile, A; & Tufaile A. P. B. Efeito da evaporação num termômetro secando. **Revista Brasileira de Ensino de Física**, vol. 44, e20220225 (9p.), 2022. <https://doi.org/10.1590/1806-9126-RBEF-2022-0225>.