



UNIVERSIDADE FEDERAL DO CEARÁ

Centro de Tecnologia

Departamento de Engenharia Mecânica

INSTRUMENTAÇÃO COM TINKERCAD - CAPÍTULO 3: TRANSDUTORES



AUTODESK®
TINKERCAD®

Transdutores

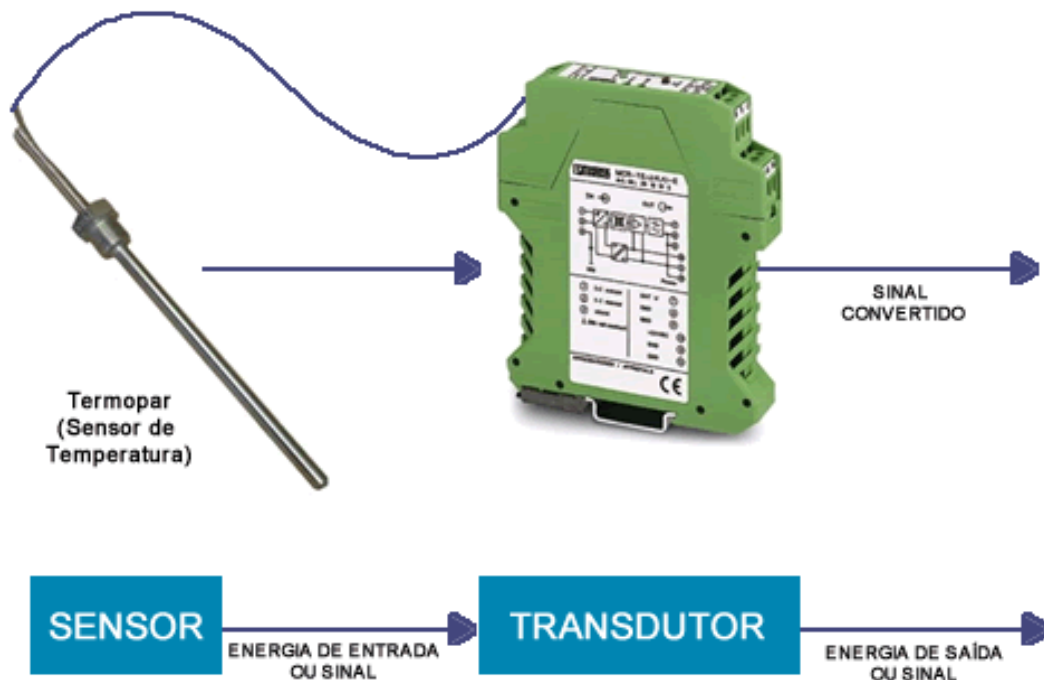
Introdução:

Transdutores são dispositivos capazes de converter uma informação sentida por um sensor em um sinal detectável, ou seja, o dispositivo é utilizado para transformar energia em outra. Por exemplo, o microfone pode ser considerado um transdutor, pois transforma a energia das ondas sonoras em sinal elétrico.

Também é válido comentar sobre os sensores, que usam um fenômeno natural para sentir a variável que está sendo medida. Portanto, sensores detectam a variável e o transdutor transforma essa variável em outra forma que seja fácil de ser medida, como um sinal elétrico.

Para exemplificar, a Fig. 1 mostra um processo de sensoriamento, o qual um sensor detecta a temperatura do meio e o transdutor transforma essa detecção em uma energia de saída para que a temperatura seja facilmente descrita.

Figura 1 - Processo de sensoriamento.



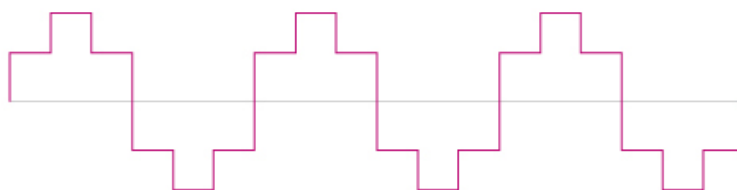
O PROCESSO DE SENSORIAMENTO

Fonte: Citisystems (2021).

Além disso, podemos citar dois tipos de sinais: analógico e digital. O primeiro é uma função contínua associada ao processo que se mede, um potenciômetro, por exemplo, fornece um sinal analógico, pois é possível variar a resistência continuamente em um determinado intervalo. Por outro lado, os sinais digitais são representados por valores inteiros, ou seja, valores discretos. Para finalizar, a Fig. 2 mostra como esses dois sinais se comportam em um determinado tempo.

Figura 2 - Sinal digital e analógico.

sinal digital



sinal analógico



Fonte: HomeIT (2021).

Projeto Sinal Analógico e Digital

Descrição:

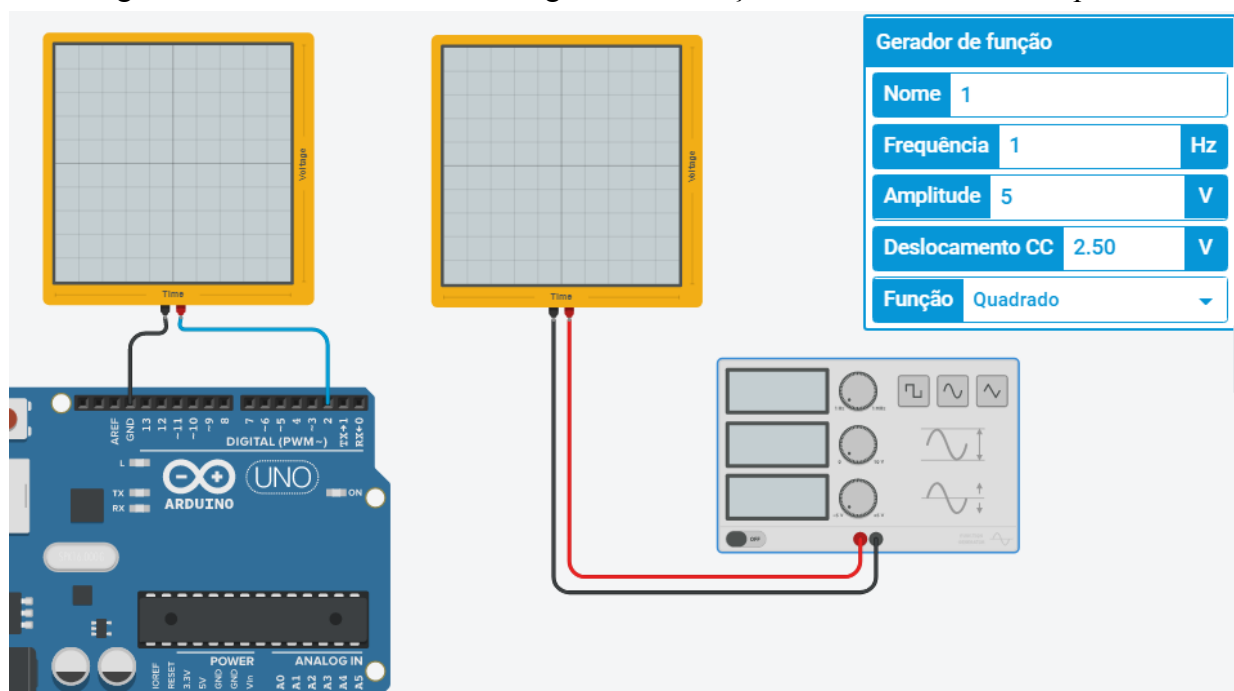
O presente projeto irá simular sinais digitais e analógicos lidos por osciloscópios. Para simular o primeiro sinal, será utilizado um arduino, o qual irá ligar e desligar uma porta digital de saída a cada 0,5 segundos. Para o segundo sinal, será utilizado a variação de resistência dado por um potenciômetro ao girar uniformemente seu pino. Além disso, também será utilizado dois geradores de função para criar tanto um sinal digital quanto um analógico na frequência de 1 Hz.

Circuito:

- O circuito é composto:
 - 2 Osciloscópio;
 - 1 Arduino UNO;
 - 4 Pilhas de 1,5 V tipo AA;
 - 1 Potenciômetro;
- Montagem do circuito

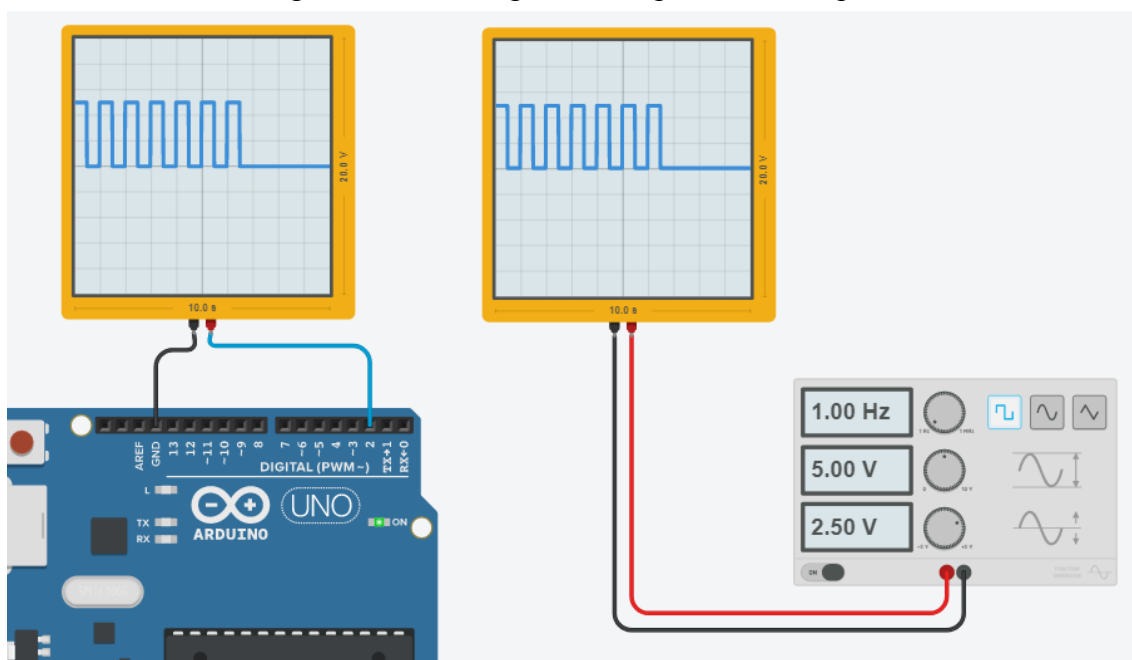
A Fig. 3 mostra as ligações entre o arduino e o osciloscópio. A programação do dispositivo será configurada para a porta 2 variar entre 0 V e 5 V a cada segundo para gerar um “sinal digital”. Além disso, também é necessário configurar o gerador de sinal para criar uma função de 1 Hz, amplitude de 5 V, função tipo quadrado e deslocamento CC de 2.5 V. Essa última parte é necessária para os dois sinais gerados serem iguais e, como o arduino não configura a porta em tensões negativas, é necessário deslocar a função para cima, como mostrado na Fig. 4

Figura 3 - Circuito do arduino e do gerador de função associado ao osciloscópio.



Fonte: Autor.

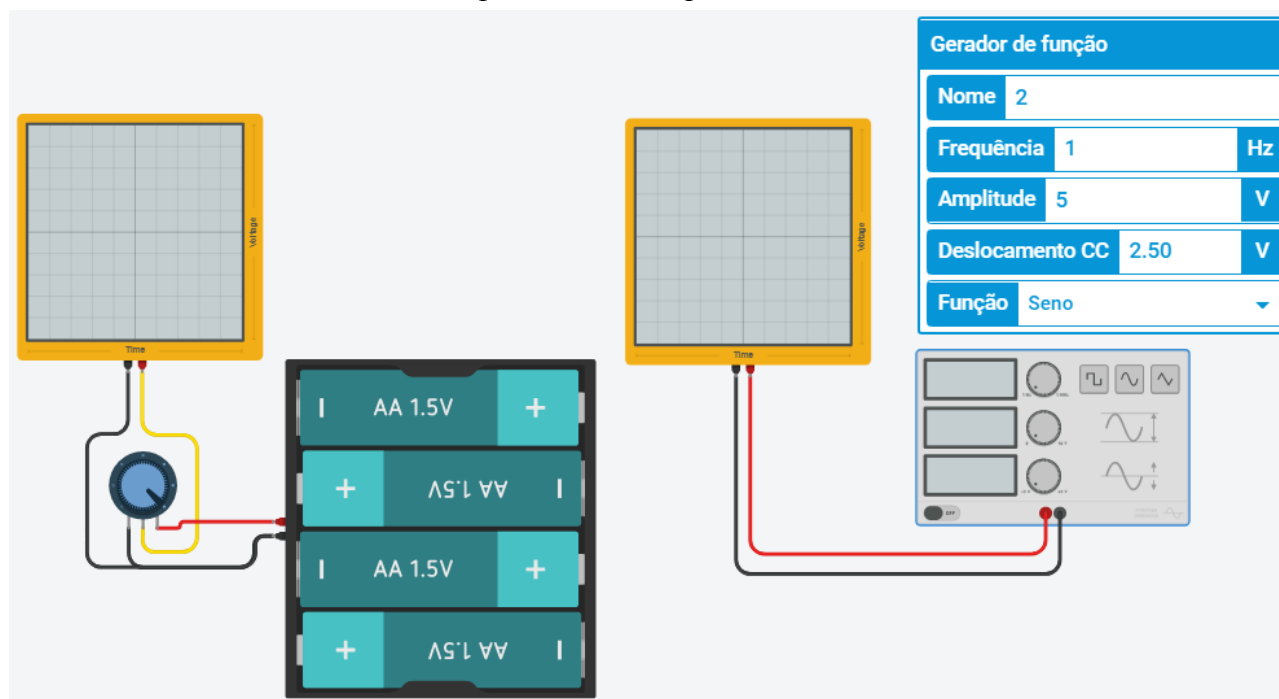
Figura 4 - Sinais digitais lidos pelo osciloscópio.



Fonte: Autor.

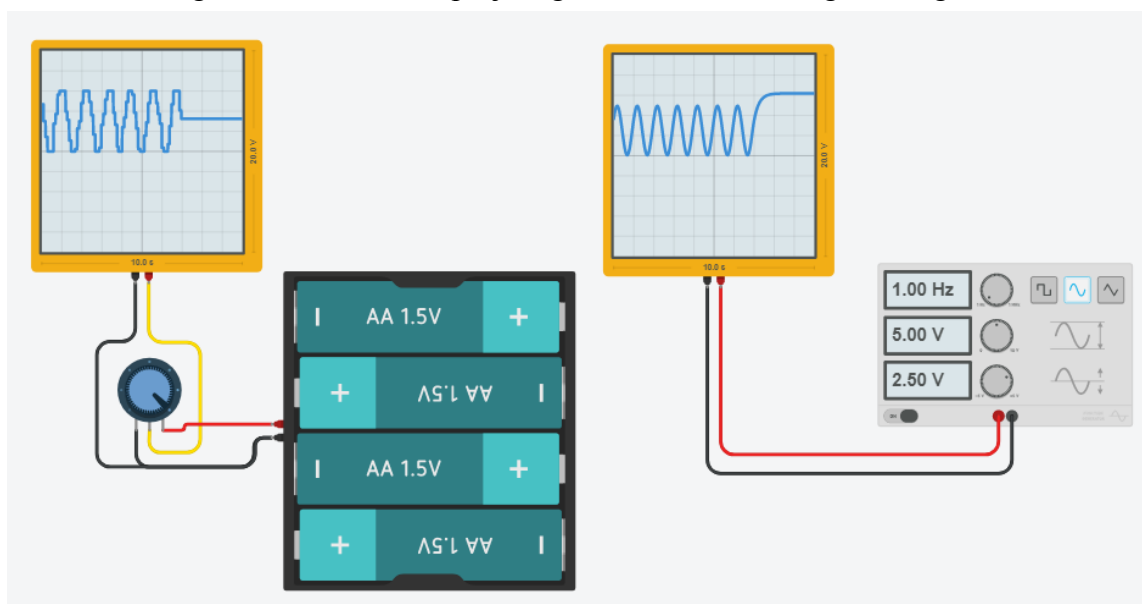
Já na Fig. 5, é possível visualizar o circuito que conecta um potenciômetro em um osciloscópio para detectar a variação de resistência. Para isso, é necessário rodar o pino do potenciômetro da capacidade mínima até a máxima repetidamente e uniformemente, implicando no sinal mostrado na Fig. 6. De forma análoga ao gerador de sinal da última função, é necessário configurar o gerador para criar uma função de 1 Hz, amplitude de 5 V, deslocamento de 2,5 V, mas a função agora é senoidal, pois é contínua com o tempo.

Figura 5 - Circuito do potenciômetro e do gerador de função para transmitir o sinal analógico para o osciloscópio.



Fonte: Autor.

Figura 6 - Circuito do projeto gerando o sinal analógico e digital.



Fonte: Autor.

Programação:

O código é bem simples, apenas é necessário estabelecer uma porta de saída que liga e desliga a cada 1 segundo, conforme a Fig. 7.

Figura 7 - Código do arduino para transmitir um sinal digital.

```

1 //Autor: Thiago Victor A. de Freitas - Estudante de Engenharia Mecânica (UFC)
2 //Inicialmente, devemos nomear quem vai está conectado a cada porta
3 int sinal_dig = 2; //a porta 2 irá transmitir o sinal digital
4 void setup() {
5   /*Comando pinMode(porta,MODO): Este função permite configurar um pino específico
6   para se comportar como um pino de entrada ou de saída. Modos:
7     INPUT: Pino de entrada
8     Output: Pino de saída */
9   pinMode(sinal_dig, OUTPUT); //Configura verd_pedestre como porta de saída
10 }
11 void loop() {
12   /*Comando digitalWrite(porta,valor): Caso a porta seja OUTPUT,
13   esta função permite configurar a tensão de saída do pino. Valores:
14     HIGH: 5 V.
15     LOW: 0 v.*/
16   digitalWrite(sinal_dig, HIGH); //Faz a porta transmitir 5V
17   /*Comando delay(tempo): pausa o programa por uma quantidade especifica de tempo
18   em milissegundos( 1000 = 1 s)*/
19   delay(500);
20   digitalWrite(sinal_dig, LOW); //Faz a porta transmitir 0V
21   delay(500);
22 }

```

Fonte: Autor.

Projeto Buzzer Como um Transdutor Piezo

Descrição:

O presente projeto irá utilizar um arduino para transmitir uma frequência que será convertida em sinal sonoro por um buzzer. É importante ressaltar que o buzzer será usado como transdutor que transforma sinal elétrico em sinal sonoro.

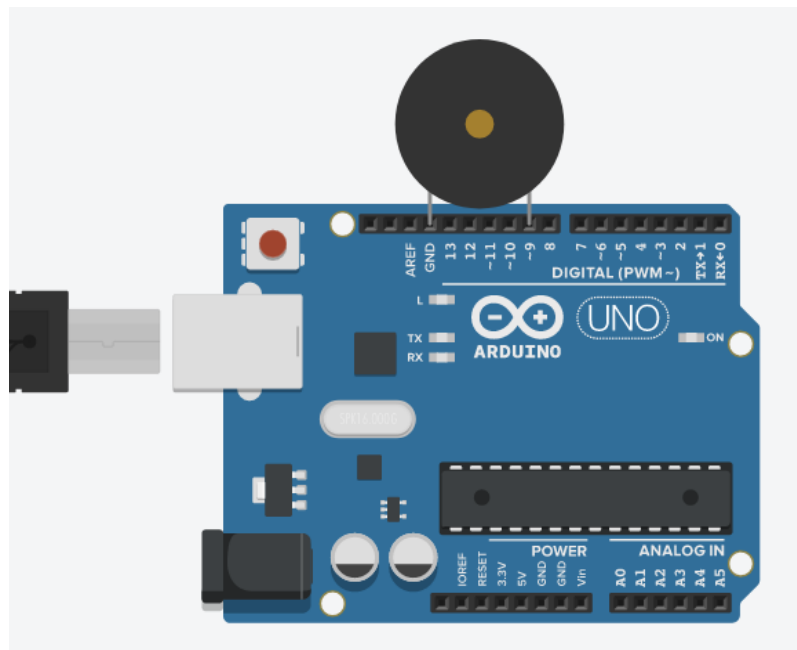
Circuito:

- O circuito é composto:
 - 2 Osciloscópio;
 - 1 Arduino UNO;
 - 4 Pilhas de 1,5 V tipo AA;
 - 1 Potenciômetro;

- Montagem do circuito:

Essa parte é bem simples, apenas é necessário conectar um pino do buzzer no GND e o outro em uma porta digital do arduino.

Figura 8 - Circuito para utilizar um buzzer.



Fonte: Autor.

Programação:

Na programação, é utilizada uma variável seno para armazenar valores de uma função seno do intervalo de 0 graus até 180 graus, como mostrado na Fig. 9. Após a declaração das variáveis utilizadas (seno, frequência e porta do buzzer), a parte do void setup() é utilizada para declarar a porta do buzzer como de saída, assim é possível transmitir a frequência.

Figura 9 - Código do arduino para transmitir sinal para o buzzer.

```
1 //Autor: Thiago Victor A. de Freitas - Estudante de Engenharia Mecânica (UFC)
2 //Inicialmente, devemos nomear quem vai está conectado a cada porta
3 float seno; //Variável para armazenar o valor do seno
4 int frequencia; //Variável que irá armazenar o valor da frequencia que o buzzer irá tocar
5 int porta_do_buzzer = 9; //O sinal para o buzzer será transmitido da porta 9
6 void setup() {
7     pinMode(porta_do_buzzer, OUTPUT); //define o porta do buzzer como saída
8 }
9 void loop() {
10     for(int x=0; x<180; x++){
11         seno=(sin(x*3.1416/180)); //Comando sin(x) dar o valor do seno de x
12         //Na ultima linha, converte-se graus para radiando e depois obtém o valor do seno
13         frequencia = (int)(seno*1000); //gera uma frequência a partir do valor do seno
14         /*Comando tone(porta,frequencia,tempo): Gera uma onda quadrada na frequência especificada
15         em um pino.
16         porta: o pino do Arduino no qual gerar o tom
17         frequência: a frequência do tom em Hertz - unsigned int*/
18         tone(porta_do_buzzer, frequencia);
19         delay(2); //espera 0,002;
20     }
21 }
```

Fonte: Autor.

Comentários:

Como é possível estabelecer a frequência que o buzzer irá tocar, é possível ajustar várias notas musicais para o buzzer tocar determinada música. O site ["dragão sem chama"](https://dragao-sem-chama.com.br/) apresenta vários códigos que tocam trilhas sonoras conhecidas, como a abertura da série Game of Thrones e a trilha sonora da marcha imperial de Star Wars.

Projeto Calibrando Sensor de Temperatura

Descrição:

O presente projeto irá medir valores de tensão de saída do sensor de temperatura TMP36, o qual fornece uma tensão que é linearmente proporcional à temperatura. Além disso, será feita a calibração do sensor conforme os valores de temperatura de referência mostrados no Tinkercad.

Circuito:

- O circuito é composto:
 - 1 Sensor de temperatura TMP36 ([Características](#));
 - 1 Fonte de tensão de 5 V (Power supply);
 - 1 Multímetro (Multimeter);

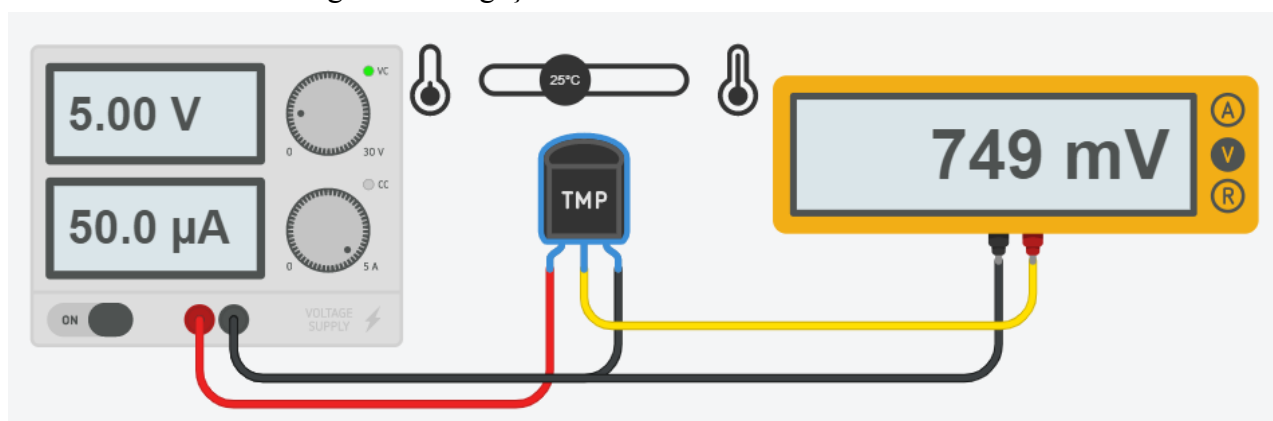
- Montagem do circuito:

Etapla 1: Adicione a lâmpada e a fonte de alimentação ao circuito buscando “TMP36” e “Power supply”, respectivamente, nos componentes.

Etapla 2: Conecte o polo positivo da fonte de alimentação no pino “POWER” do sensor e o polo negativo no “GND” do sensor.

Etapla 3: Adicione o multímetro no circuito pesquisando “Multimeter” nos componentes. Além disso, conecte o pino preto do multímetro no polo negativo da fonte, já o pino vermelho, conecte Vout do sensor, como é mostrado na Fig. 10.

Figura 10 - Ligações do multímetro no circuito.

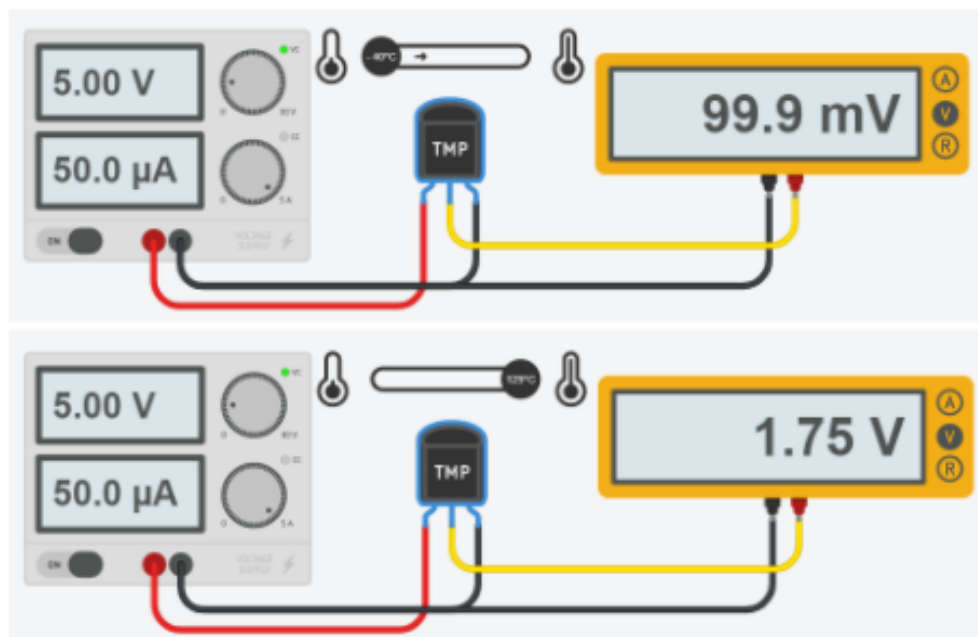


Fonte: Autor.

Comentários:

O multímetro serve para medir valores de tensão fornecidos pelo sensor e, essa tensão varia conforme a temperatura. Para uma temperatura de -40°C e de 125°C, que são os valores extremos do sensor, tem-se, respectivamente, uma tensão de 0,0999 e 1,75 V, como é mostrado na Fig. 11.

Figura 11 - Medições dos valores extremos do sensor de temperatura.



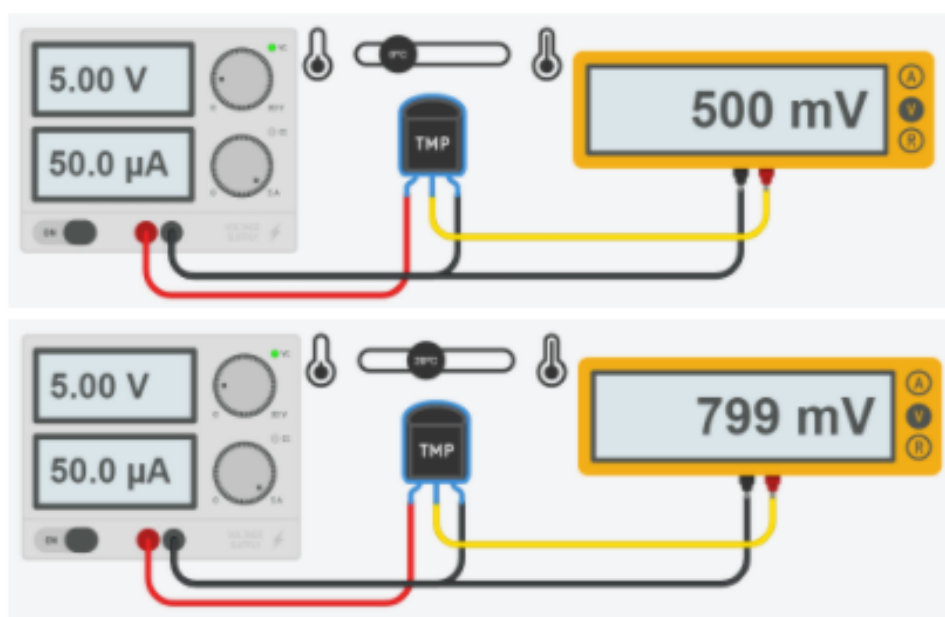
Fonte: Autor.

Interpolando esses valores, considerando V_{out} como tensão de saída e Temp a temperatura que fornece essa tensão, temos que a relação entre esses dois é

$$\frac{0,0999 - 1,75}{-40 - 120} = \frac{V_{out} + 1,75}{Temp - 120} \rightarrow V(Temp) = 0,01Temp + 0,5$$

Com base nessa equação, sabe-se que as temperaturas de 0°C e 30°C fornecem tensões de 0,5 e 0,8 V, respectivamente. Isso é comprovado na Fig. 12, a qual mostra o valor medido pelo multímetro para essas temperaturas. O próximo projeto irá calibrar esse sensor utilizando o microcontrolador Arduino e mostrar o valor de temperatura em um Display.

Figura 12 - Valores de tensões para temperaturas de 0°C e 30°C.



Fonte: Autor.

Projeto Display Como um Transdutor

Descrição:

O presente projeto irá utilizar o sensor de temperatura TMP36 em conjunto com o Display LCD 16x2 para atuar como um transdutor e mostrar os dados obtidos em Celsius e Fahrenheit. O arduino servirá para processamento, pois irá receber os dados do sensor e enviá-los para o Display.

Além disso, o circuito tem aplicações nos seguintes sistemas:

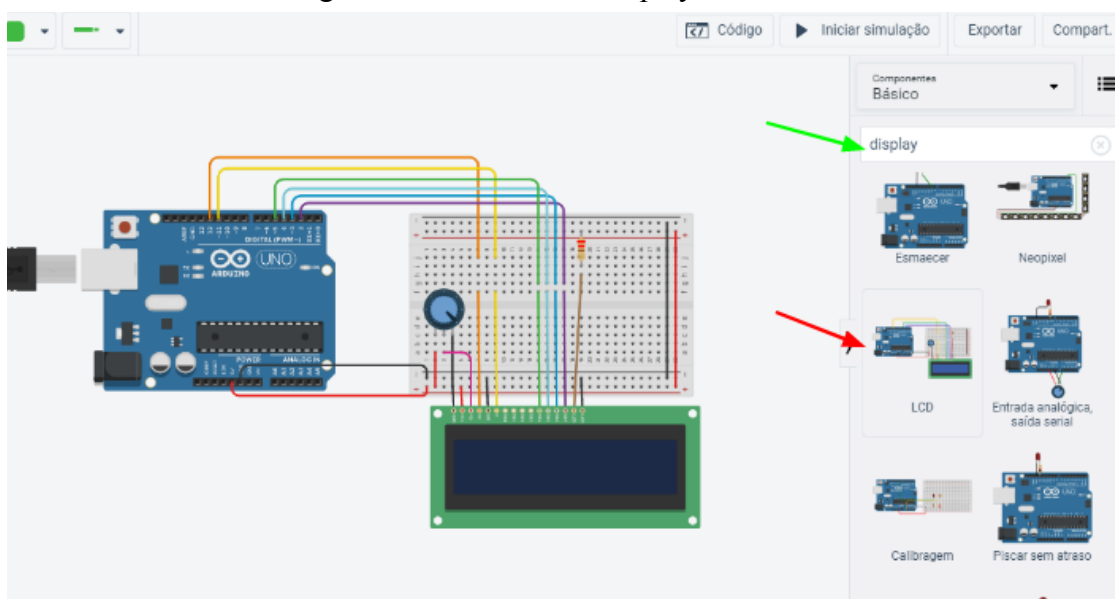
- Sistemas de controle ambiental;
- Proteção térmica;
- Controle de processos industriais;
- Alarmes de incêndio;
- Monitores do sistema de energia;
- Gerenciamento térmico com arduino;

Circuito eletrônico:

- O circuito é composto:
 - 1 Arduino UNO;
 - 1 Resistor de 10k ohms;
 - 1 Sensor de temperatura TMP36 ([Características](#));
 - 1 Display LCD 16x2;
- Passo a passo:

Etapas 1: Ao digitar display na busca dos componentes, irá aparecer o circuito com todas as ligações já feitas com o LCD 16x2, assim clique nele e arraste para a tela de circuitos, como mostrado na Fig. 13, o qual a seta verde sinaliza a opção de busca e, a seta vermelha, o circuito que deve ser escolhido.

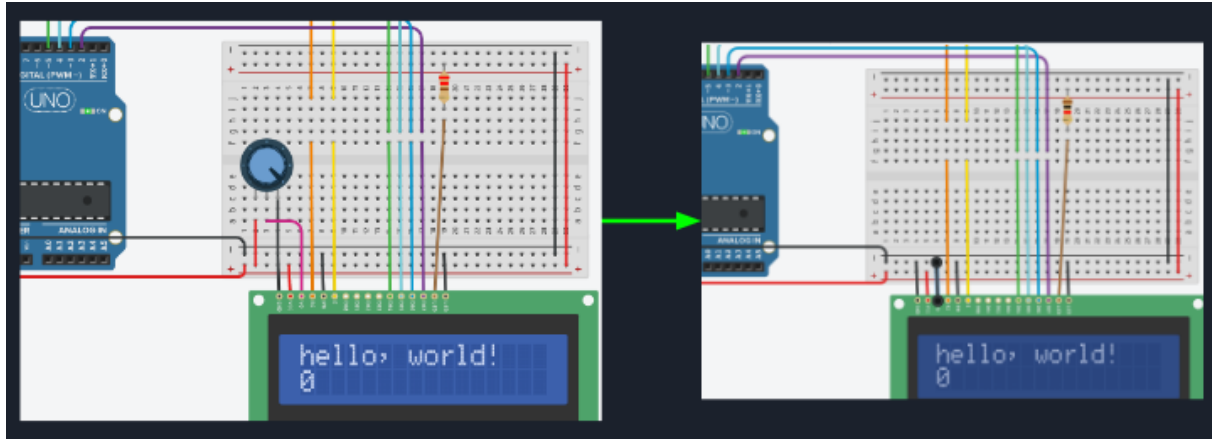
Figura 13 - Circuito do Display LCD16x2.



Fonte: Autor.

Etapa 2: Para simplificar o circuito, tire o potenciômetro e conecte o GND na terceira porta do LCD (este é o lugar que estava conectado o fio rosa). Também é importante ressaltar que o resistor serve para definir o brilho no display (quanto maior a resistência, menor será o brilho). Esse processo pode ser visualizado na Fig. 14.

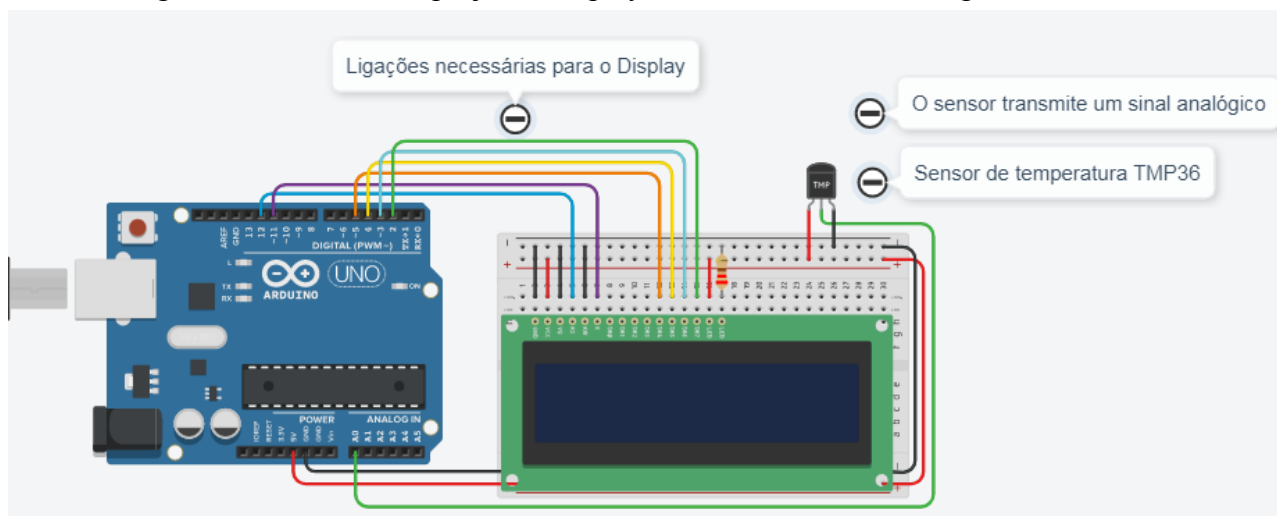
Figura 14 - Processo de remoção do potenciômetro e troca da resistência.



Fonte: Autor.

Etapa 3: Organize o circuito para ficar o mais simples possível, como o da Fig. 15, e inclua o sensor TMP36 com o pino do meio conectado na porta analógica A0, pois este sensor irá transmitir sinal analógico e a conversão de seu valor será aplicado na programação.

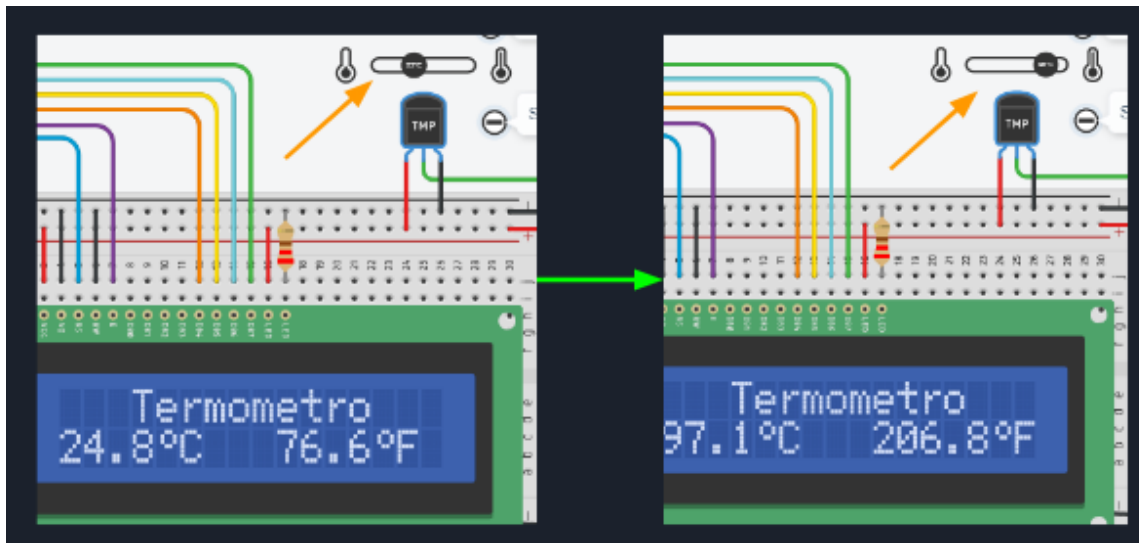
Figura 15 - Circuito do projeto “Display como um transdutor” organizado.



Fonte: Autor.

Por fim, é possível variar a temperatura do meio clicando no sensor TMP36 e deslocando a esfera preta acima do sensor, conforme mostrado na Fig. 16.

Figura 16 - Ajuste da temperatura no circuito.



Fonte: Autor.

Programação:

Para converter o sinal de saída do sensor TMP36 na temperatura de graus Celsius, é necessário utilizar a seguinte equação, conforme o guia do sensor no site [CAP Sistema](#):

$$Temperatura(^{\circ} C) = \frac{V_{out} - 500}{10}$$

Sendo o V_{out} , a tensão lida pela porta analógica do arduino em miliVolt (mV).

Além disso, o código também deve ter a conversão do valor lido pela porta analógica no valor equivalente de tensão, pois a função `analogRead()` do arduino retorna um número entre 0 e 1023, logo deve-se utilizar a seguinte fórmula para encontrar o valor da tensão de saída do sensor:

$$V_{out} = \frac{analogRead(porta) * 5000}{1024.0}$$

Com base nisso, a parte inicial da programação servirá para definir as portas/pinos e as configurações iniciais do Display LCD, conforme mostrado na Fig. 17. Além disso, também é criado um byte para visualizar o símbolo do grau (“°”) no LCD.

Figura 17 - Configurações para o uso do sensor TMP36 e do Display LCD no código do arduino.

```

1 //Configurações para o uso do sensor
2 /*Comando #define: utilizado para definir um nome a um valor constante
3  #define NomeDaConstante valor*/
4 #define Porta_do_sensor A0//Porta de entrada de dados do sensor TMP36
5 //Configurações para o uso do Display
6 #include <LiquidCrystal.h> //Biblioteca do display
7 #define RS 12 //Portas do LCD
8 #define E 11
9 #define DB4 5
10 #define DB5 4
11 #define DB6 3
12 #define DB7 2
13 #define LINHAS 2
14 #define COLUNAS 16
15 //Criação objeto lcd da classe LiquidCrystal
16 LiquidCrystal lcd(RS, E, DB4, DB5, DB6, DB7);
17 byte grau[8] = { //Desenho do grau
18     B00000110,
19     B00001001,
20     B00001001,
21     B00000110,
22     B00000000,
23     B00000000,
24     B00000000,
25     B00000000
26 };

```

Fonte: Autor.

Também é importante criar uma função para posicionar a mensagem "Termômetro" no centro da linha de cima do Display, a função está descrita na Fig. 15 e seu resultado foi mostrado na Fig. 16.

Figura 18 - Configurações do setup e da função de centralização da mensagem.

```

33 void setup(){
34     Serial.begin(9600);
35     lcd.begin(COLUNAS, LINHAS); //Inicia o Display e especifica a altura e largura(matriz)
36     lcd.createChar(0, grau); //Criar o caractere grau e referencia ele ao valor 0
37     pinMode(Porta_do_sensor, INPUT); //Configura a porta do sensor como Entrada
38 }
39
40 int posicionarMensagem(String msg){
41     /*Função para posicionar o cursor do LCD para deixar a mensagem centralizada.
42     Se o tamanho da mensagem é par, então o intervalo que sobra
43     é simétrico, quando o tamanho é ímpar o intervalo é dividido
44     de forma que a parte à esquerda é uma unidade maior que a direita.*/
45     if(msg.length() % 2 == 0){
46         return (COLUNAS - msg.length())/2;
47         //msg.length() é utilizado para calcular o tamanho da mensagem
48     }else{
49         return ((COLUNAS - msg.length())/2)+1;
50     }
51 }

```

Fonte: Autor.

Para finalizar, os comandos do loop, apresentados na Fig. 19, contém o lógica para apresentar, de forma organizada, os valores medidos pelo sensor no Display e, também,

utiliza-se das equações acima para convertê-lo em seus respectivos valores nas escalas de temperatura Celsius e Fahrenheit.

Figura 19 - Comandos do loop no código do arduino.

```
48 void loop()
49 {
50   String mensagem = "Termometro"; //Define a mensagem
51   int pos = posicionarMensagem(mensagem);
52   //Utilizado para indicar a posição que centraliza a mensagem
53   //Calcula o valor da temperatura em Celsius
54   float Vout = analogRead(Porta_do_sensor)*(5000/1024.0);
55   float tempCelsius = (Vout - 500)/10;
56   //Calcula o valor da temperatura em Fahrenheit
57   float tempFahrenheit = (tempCelsius * 1.8) + 32;
58   //Print da mensagem
59   lcd.setCursor(pos, 0); //Posiciona o cursor para o início da mensagem
60   lcd.print(mensagem); //Escreve a mensagem
61   //Print da temperatura em Celsius
62   lcd.setCursor(0, 1); //Posiciona o cursor na coluna 0 e linha 1
63   lcd.print(tempCelsius, 1); //Escreve a temperatura
64   lcd.write((byte)0); //Escreve o grau
65   lcd.print("C "); //Escreve "C"
66   //Print da temperatura em Fahrenheit
67   lcd.setCursor(9, 1); //Posiciona o cursor na coluna 9 e linha 1
68   lcd.print(tempFahrenheit, 1); //Escreve a temperatura
69   lcd.write((byte)0); //Escreve o grau
70   lcd.print("F "); //Escreve "F"
71 }
```

Fonte: Autor.

Projeto Sensor de Gás

Descrição:

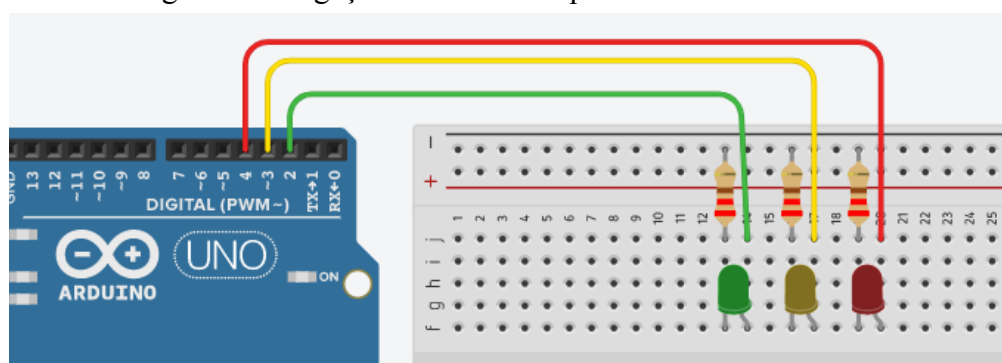
O presente projeto irá utilizar um buzzer para emitir sons quando um sensor de MQ-6 detectar a presença de gás próximo a ele. Além disso, será utilizado uma sequência de LEDs (verde, amarelo e vermelho) para indicar a concentração do gás, ou seja, se for muito forte o piezo é acionado junto com o LED vermelho. É importante destacar que o sensor identifica apenas gás de cozinha, porém, se ele tivesse sensibilidade a fumaça, ele seria um ótimo dispositivo para ser usado em um detector de incêndio. Por fim, cabe ressaltar que o circuito foi baseado no vídeo [Learn Top 5 Arduino Sensors Projects without any Hardware! Tinkercad](#) do canal THE ELECTRONIC GUY.

Circuito eletrônico:

- O circuito é composto:
 - 1 Arduino UNO;
 - 1 Sensor de gás MQ-6 ([Características](#));
 - 1 LED verde;
 - 1 LED amarelo;
 - 1 LED vermelho;
 - 1 Buzzer;
 - 1 Protoboard;
 - 4 Resistores de 220 Ω ;
- Montagem do circuito:

Etapa 1: Adicione os LED na protoboard na seguinte ordem mostrada na Fig. 20 e conecte o pino positivo deles às portas 4,3 e 2 do arduino. Em cada pino negativo dos LEDs, conecta-se os resistores de 220 ohms.

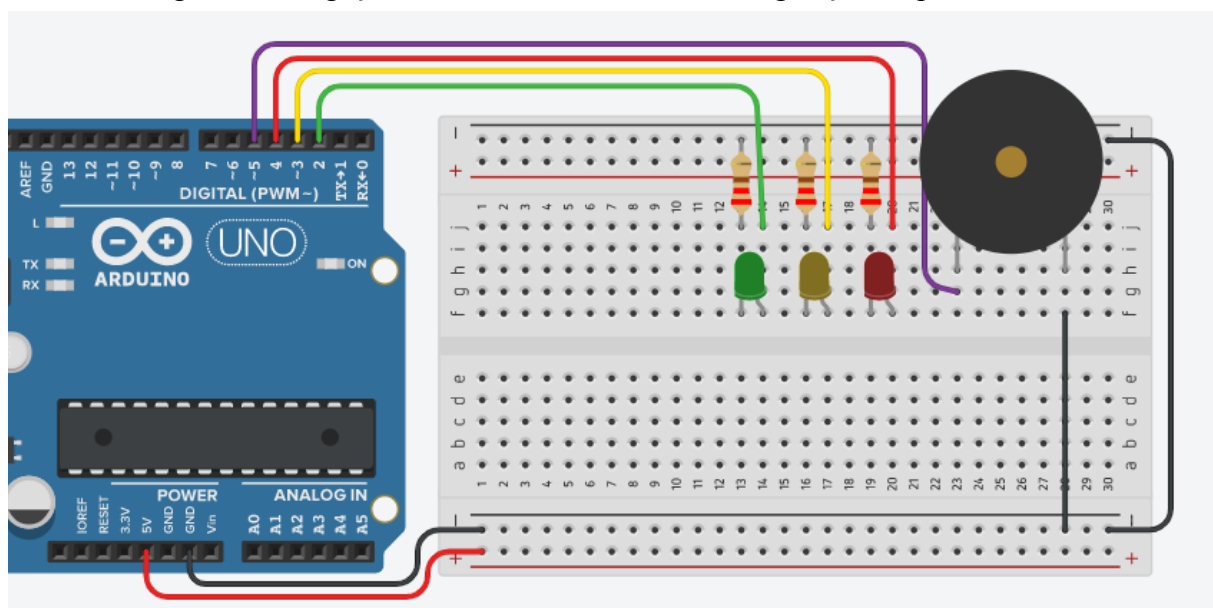
Figura 20 - Ligações dos LED na protoboard e no arduino.



Fonte: Autor.

Etapa 2: Adicione o Buzzer na protoboard e conecte o pino positivo dele na porta 5 do arduino. Após isso, energisa a protoboard conectando o pino 5 V do arduino ao + da protoboard e, conectando também, o pino GND do arduino no - da protoboard, como é mostrado na Fig. 21. Aqui, é válido ressaltar que foi alimentado primeiro duas linhas de baixo da protoboard, após isso, foram puxados dois fios para alimentar as duas linhas de cima e, consequentemente, fechar o circuito dos LEDs.

Figura 21 - Ligações do buzzer ao arduino e energização da protoboard.



Fonte: Autor.

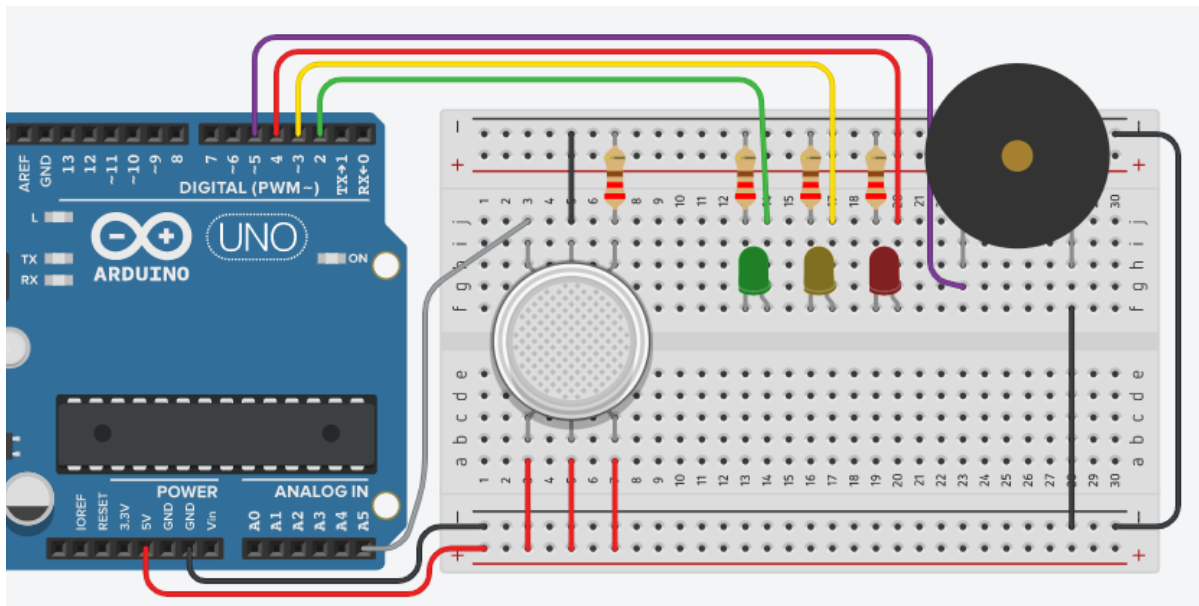
Etapa 3: Pesquisa “sensor de gás” na busca de componentes e o adicione ao circuito. Para isso, utilize a Tabela a seguir para identificar as ligações de cada pino do sensor. Além disso, as ligações podem ser visualizadas na Fig. 22.

Tabela 1 - Sequência de ligações do sensor de gás.

Pino no sensor	Ligado ao	Cor
A1	A5 do Arduino	Vermelho
H1	Negativo da protoboard	Preto
A2	Resistor de 220 Ω e negativo da protoboard	-
B1	Positivo da protoboard	Vermelho
H2	Positivo da protoboard	Vermelho
B2	Positivo da protoboard	Vermelho

Fonte: Autor.

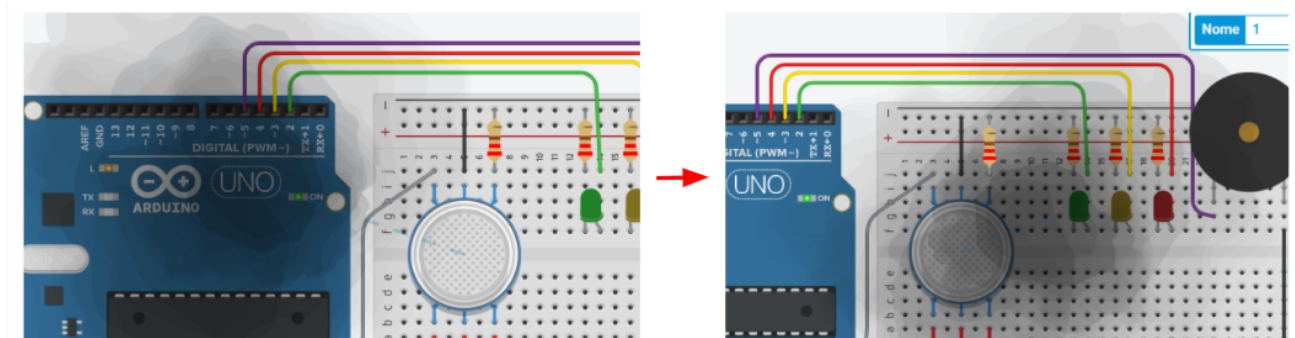
Figura 22 - Ligações do buzzer ao arduino e energização da protoboard.



Fonte: Autor.

Por fim, cabe ressaltar que, para modificar a posição do gás, é necessário clicar primeiro no sensor, ao iniciar a simulação, e movimentar a fumaça preta que aparece, como pode ser visto na Fig. 23.

Figura 23 - Ligações do buzzer ao arduino e energização da protoboard.



Fonte: Autor.

Programação:

Inicialmente, a Fig. 24 mostra as configurações iniciais do código, estabelecendo as portas dos LEDs, do buzzer e do sensor conforme a montagem do circuito. Além disso, também é possível visualizar duas formas de declarar os pinos como de saída, ou OUTPUT. Uma delas é fazendo a configuração e nomeando cada pino, conforme as linhas 11 e 14. A outra forma é utilizando um loop de repetição, o qual a variável *i* inicia no 2 e vai até 5 e, dentro do loop, utiliza-se o comando `pinMode` para declarar o pino *i*.

Figura 24 - Configurações iniciais e void setup () do código do sensor de gás.

```

1 //Autor: Thiago Victor A. de Freitas - Estudante de Engenharia Mecânica (UFC)
2 //Inicialmente, defini-se as portas
3 # define verde 2
4 # define amarelo 3
5 # define vermelho 4
6 # define buzzer 5
7 # define sen_gas A5
8
9 void setup() {
10 /*Pode-se definir os pinos assim:
11 pinMode(verde,OUTPUT);//defini a porta do LED verde como OUTPUT
12 pinMode(amarelo,OUTPUT);//defini a porta do LED amarelo como OUTPUT
13 pinMode(vermelho,OUTPUT);//defini a porta do LED vermelhocomo OUTPUT
14 pinMode(buzzer,OUTPUT);//defini a porta do buzzer como OUTPUT
15 Ou assim:*/
16 for (int i = 2; i < 6; i++)
17     pinMode(i, OUTPUT);//Define a porta i como OUTPUT
18
19 Serial.begin(9600);//Inicial a comunicação serial
20 }
21

```

Fonte: Autor.

Na Fig. 25, é possível visualizar a parte do loop no código, o qual, inicialmente, desliga os LEDs e o buzzer. Após isso é feito a leitura do pino o qual contém o sensor de gás e realiza-se o seguinte testes: Se a leitura for maior do que 40, então o LED vermelho é acesso e o buzzer começa a tocar; se não se ainda tiver acima de 25, apenas o LED vermelho e aceso; se não for nenhuma dessas duas opções, o LED verde é acesso. Por fim, o código espera 0,01 segundos antes de repetir o loop.

Figura 25 - Void loop do código do sensor de gás.

```

22 void loop() {
23     //Loop de repetição para desligar todos os LEDs e a emissão do buzzer
24     for (int i = 2; i < 6; i++)
25         digitalWrite(i, LOW);//Desliga a porta i
26
27     int leitura_do_gas = analogRead(sen_gas);//Cria uma variável para ler o pino do sensor
28     Serial.println(leitura_do_gas);//Imprime no serial monitor o valor lido
29     if ( leitura_do_gas >= 40 ) {
30         //Se o valor lido for maior ou igual a 40
31         digitalWrite(vermelho, HIGH);//Liga o LED vermelho
32         digitalWrite(buzzer,HIGH);//Emite som com o buzzer
33     }else if ( leitura_do_gas >= 25 ) {
34         //Se não tiver acima de 40 e se tiver acima de 25 o valor lido
35         digitalWrite(amarelo, HIGH);
36     } else { //Se não for nenhuma as duas condições anteriores, então
37         digitalWrite(verde, HIGH);//Liga o LED verde
38     }
39     delay(10);//Espera 0,01 segundos
40 }
41

```

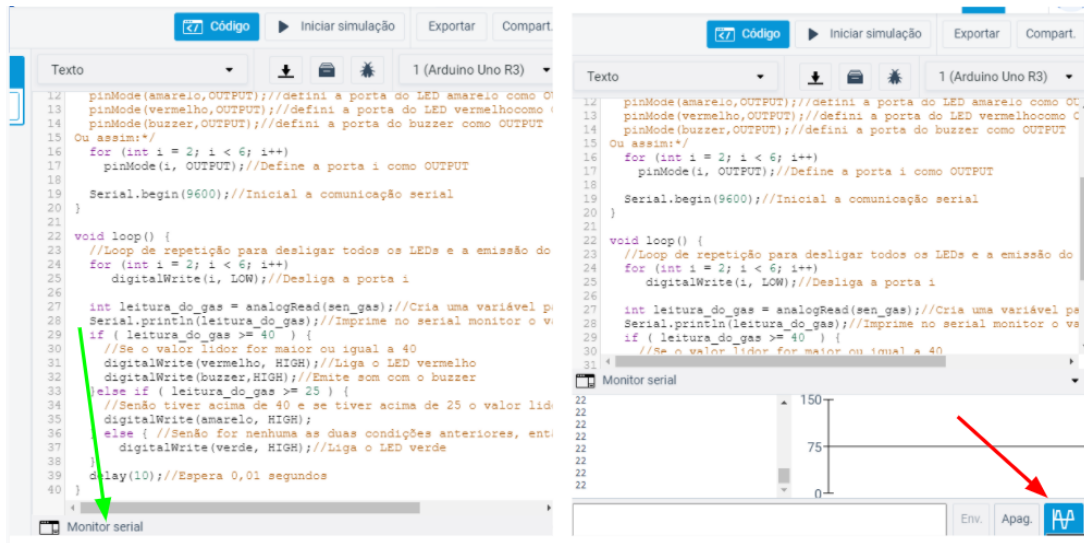
Fonte: Autor.

Comentário:

É importante destacar que foi escolhido esses valores para a estrutura de condição porque o sensor responde com um valor entre 0 e 100, sendo este último representando maior concentração do gás.

Além disso, é possível visualizar o valor lido no monitor serial de forma numérica clicando na opção “MONITOR SERIAL”, indicado pela seta verde na Fig. 26; e, também, é possível visualizar na forma de gráfico clicando na opção sinalizada pela seta vermelha.

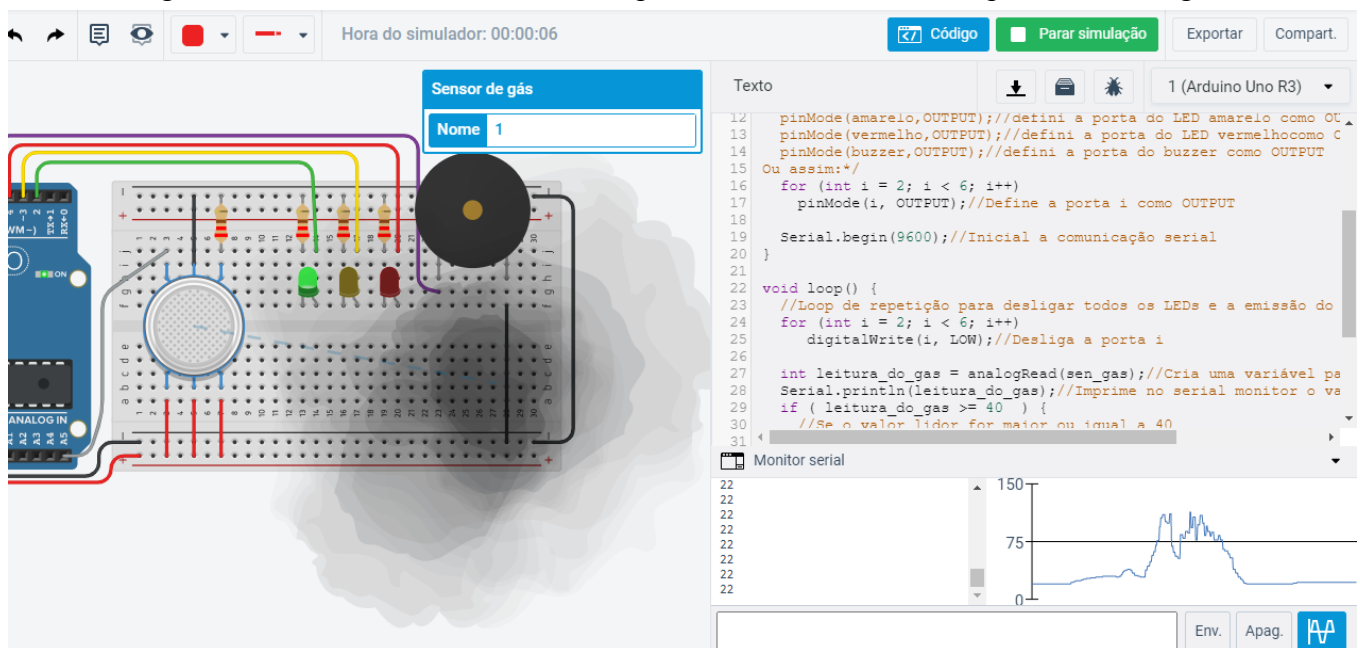
Figura 26 - Setas indicando as opções do monitor serial .



Fonte: Autor.

Com base nisso, é possível visualizar a concentração do gás no ambiente após um determinado tempo, como mostrado na Fig. 27.

Figura 27 - Utilizando o monitor serial para visualizar o valor lido pelo sensor de gás.



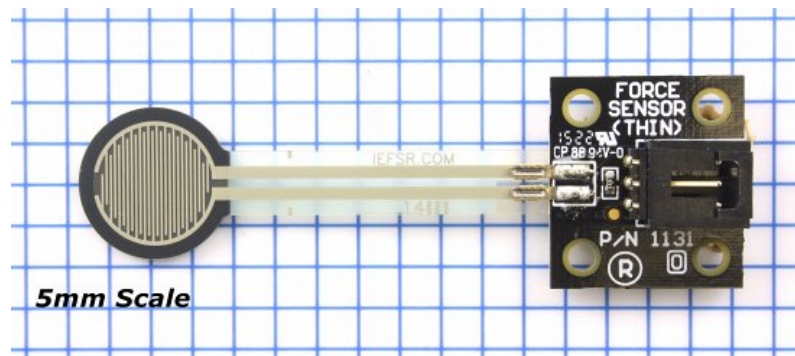
Fonte: Autor.

Projeto Sensor de Força

Descrição:

O seguinte projeto irá medir forças a partir do “Thin Force Sensor” presente no Tinkercad, o qual é mostrado pela Fig. 28, e emitir uma onda sonora com frequência proporcional a essa força.

Figura 28 - Thin Force Sensor.



Fonte: Phidgets.

Circuito:

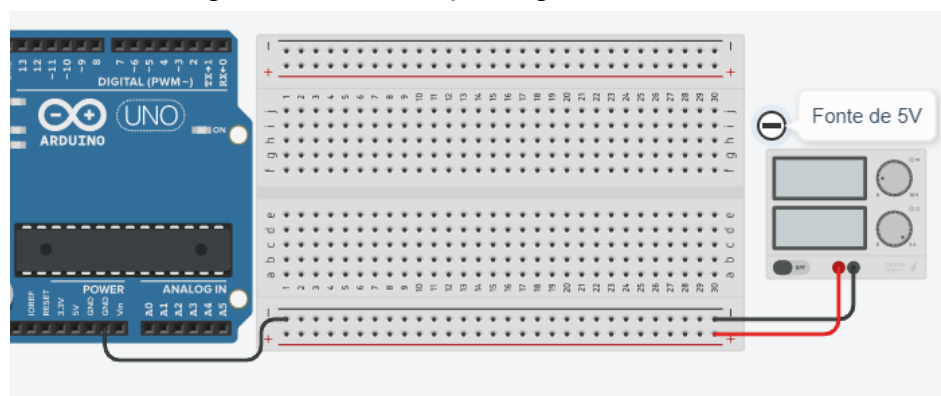
- O circuito é composto:
 - The Force Sensor ([características](#));
 - 1 Arduino;
 - 1 Protoboard;
 - 1 Fonte de energia de 5 V;
 - 1 Resistor de 10 Ω ;
 - 1 Buzzer;

- Montagem do circuito:

Etapa 1: Adicione o Arduino, a protoboard e 1 fonte ao circuito buscando “Arduino”, “Breadboard”, “Power supply”, respectivamente, nos componentes. É importante destacar que a fonte deve ser configurada para transmitir 5 V.

Etapa 2: Conecte a fonte no “+” e no “-” da protoboard. Por fim, conecte o GND do arduino no polo negativo da protoboard, como mostrado na Fig. 29

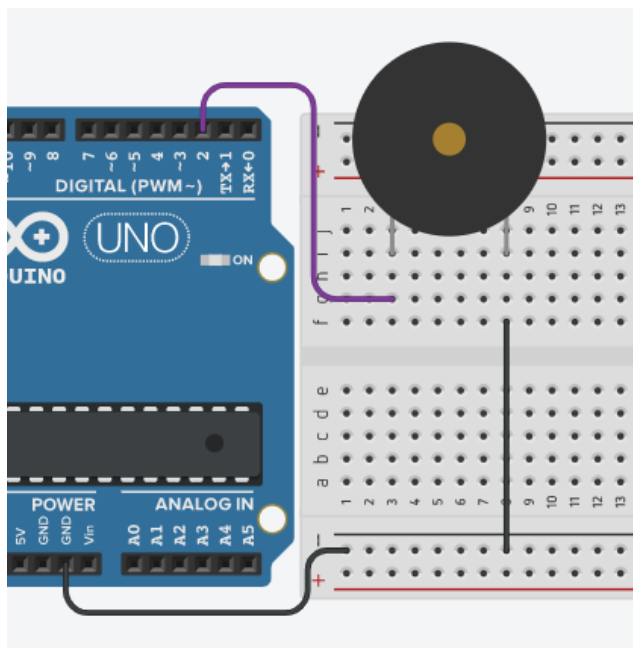
Figura 29 - Alimentação da protoboard.



Fonte: Autor.

Etapa 3: Adicione o buzzer buscando “Piezo” nos componentes e conecte um de seus pinos no polo negativo da protoboard, já o outro pino, conecte na porta 2 do Arduino, como mostrado na Fig. 30.

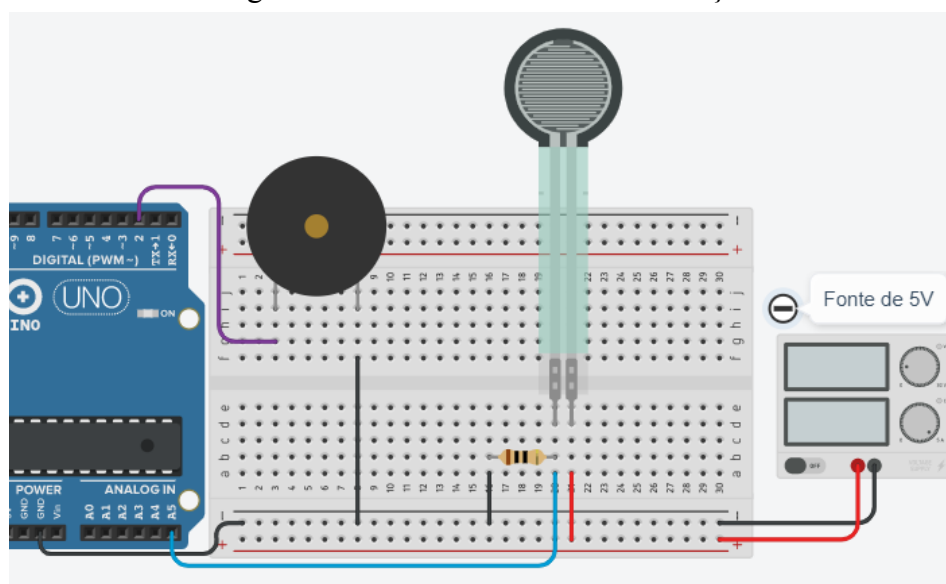
Figura 30 - Conexões do buzzer.



Fonte: Autor.

Etapa 4: Adicione o sensor de força buscando “Force sensor” nos componentes. Além disso, conecte o resistor de 10 Ω no GND e em um dos pinos do sensor, como é mostrado na Fig. 31. Além disso, conecte 5 V no outro pino do sensor. Por fim, para o Arduino ler os valores fornecidos pelo sensor, é necessário puxar um fio do mesmo pino em que se encontra o resistor e conectá-lo na porta A5 do Arduino.

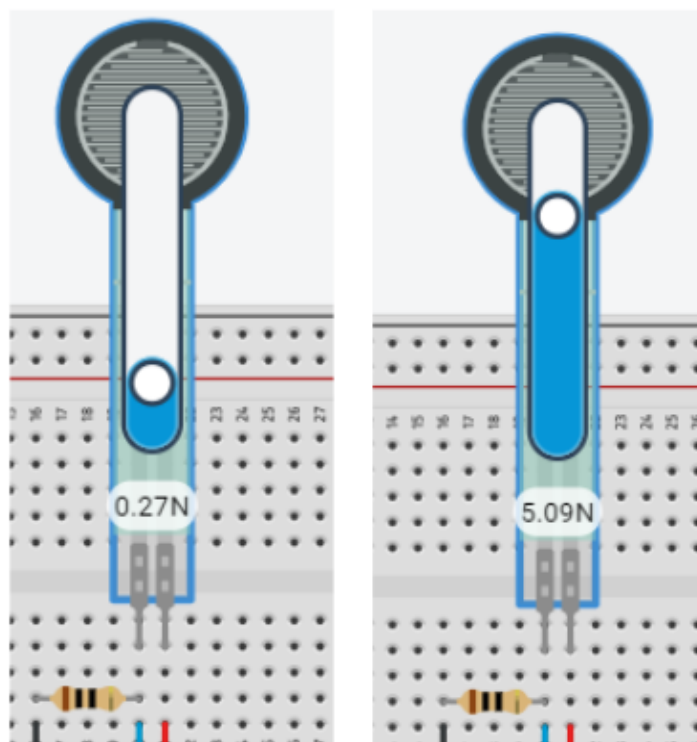
Figura 31 - Conexões do sensor de força.



Fonte: Autor.

Para finalizar, cabe ressaltar que é possível modificar o valor da força medida movimentando o círculo branco após clicar no sensor, como é mostrado na Fig. 32.

Figura 32 - Alterando o valor da força medida pelo sensor.



Fonte: Autor.

Programação:

Para iniciar o código, estabelece as variáveis que irão representar as portas do Arduino, como a que está conectada no buzzer ou no sensor de força. Além disso, é criado um vetor que irá armazenar na primeira posição o último valor lido e, na segunda posição, o penúltimo valor lido. Isso é necessário para a lógica que faz o buzzer emitir som proporcional ao valor da força. Após isso, configura-se essas portas de entrada e saída de sinal e inicia o serial monitor, como é mostrado na Fig. 33.

O loop do código se inicia armazenando o valor lido no vetor e, com base nisso, calcula-se o valor da frequência para o buzzer tocar. Como o buzzer dá problemas se for solicitado que ele toque várias frequências de uma vez, é utilizado uma estrutura condicional para ele parar de tocar se o último valor lido for diferente do penúltimo valor lido. Isto é, ele vai parar de tocar enquanto a medida do sensor estiver variando e, quando o valor lido for estabilizado em um determinado valor, o buzzer irá tocar.

Figura 33 - Código para realizar a leitura do sensor e emitir som com o buzzer.

```

1 //Autor: Thiago Victor A. de Freitas - Estudante de Engenharia Mecânica (UFC)
2 //Inicialmente, devemos nomear quem vai está conectado a cada porta
3 #define sen_forca A5 //Estabelece o pino A5 como a porta do sensor de força
4 #define buzzer 2 //Estabelece o pino 2 como a porta do buzzer
5 float valor_da_forca[] = {0,0}; //Vetor que armazena o valor da força
6 float frequencia; //Variável que irá armazenar o valor da frequencia
7 void setup() {
8     pinMode(sen_forca , INPUT); //Configura sen_force como porta de entrada de dados.
9     pinMode(buzzer , OUTPUT); //Configura buzzer como porta de saída de dados.
10    Serial.begin(9600); //Iniciando a comunicação serial do arduino
11 }
12 void loop() {
13     valor_da_forca[0] = analogRead(sen_forca); //Ler o valor do sensor
14     frequencia = valor_da_forca[0]*1000*sin(valor_da_forca[0]*3.1416/180);
15     tone(buzzer,frequencia); //Faz o buzzer tocar em determinada frequencia
16     if (valor_da_forca[1] != valor_da_forca[0]){
17 //Se o valor lido pela força for diferente da ultima leitura
18         noTone(buzzer); //Encerra o toque do buzzer
19     }
20     valor_da_forca[1] = valor_da_forca[0]; //O segundo espaço do vetor recebe
21 //o novo valor lido
22     Serial.print("Valor da força:"); //Escreve no monitor serial
23     Serial.print(valor_da_forca[0]); //Mostra o valor lido no serial monitor
24     Serial.print(";Valor da frequencia:"); //Escreve no monitor serial
25     Serial.println(frequencia); //Mostra a frequencia do buzzer no serial monitor
26     delay(10); //Espera 0,01s
27 }
28

```

Fonte: Autor.

Para finalizar, o valor da força e da frequência são mostrados no monitor serial, como é possível visualizar na Fig. 34.

Figura 34 - Valores mostrados no monitor serial do Arduino.



Serial Monitor

```

Valor da força:6.00;Valor da frequencia:627.17
Valor da força:6.00;Valor da frequencia:627.17
Valor da força:7.00;Valor da frequencia:853.09
Valor da força:7.00;Valor da frequencia:853.09
Valor da força:7.00;Valor da frequencia:853.09
Valor da força:7.00;Valor da frequencia:853.09
Valor da força:8.00;Valor da frequencia:1113.39
Valor da força:8.00;Valor da frequencia:1113.39

```

Fonte: Autor.

Referências

Sites:

- [Site Ctisystems descrevendo o que é um sensor](#)
- [Livro do autor França sobre instrumentação](#)
- [Mais sobre sensores e transdutores.](#)
- [Circuito do projeto Sinal Analógico e Digital](#)
- [Circuito do projeto Buzzer Como um Transdutor Piezo](#)
- [Tocando música com arduino.](#)
- [Guia do uso do sensor TMP36.](#)
- [Circuito do projeto Calibrando Sensor de Temperatura](#)
- [Explicação básica sobre o LCD.](#)
- [Circuito do projeto Display Como um Transdutor.](#)
- [IMPLEMENTAÇÃO DE UM SENSOR DE DIÓXIDO DE CARBONO NO AR
CONDICIONADO DE VEÍCULOS POPULARES](#)
- [Exemplo de circuito utilizado para medir a concentração de gás](#)
- [Vídeo explicativo construindo o circuito do sensor de gás](#)
- [Circuito do projeto Sensor de Gás.](#)
- [Circuito do projeto Sensor de Força](#)