

Arco Elétrico Cantante

Introdução

Este experimento produz um arco elétrico entre os terminais do circuito, gerando um ruído característico deste tipo de arco. Se o arco elétrico for ligado e desligado em uma determinada frequência, ouviremos o som característico desta frequência. Agora, se a sequência de arcos elétricos produzidos seguir um conjunto de frequências equivalente às notas musicais de uma canção conhecida, poderemos ouvi-la através deste arco voltaico. É justamente isso que este experimento nos possibilita realizar! A Figura 1 abaixo mostra uma possível configuração para a montagem do circuito.

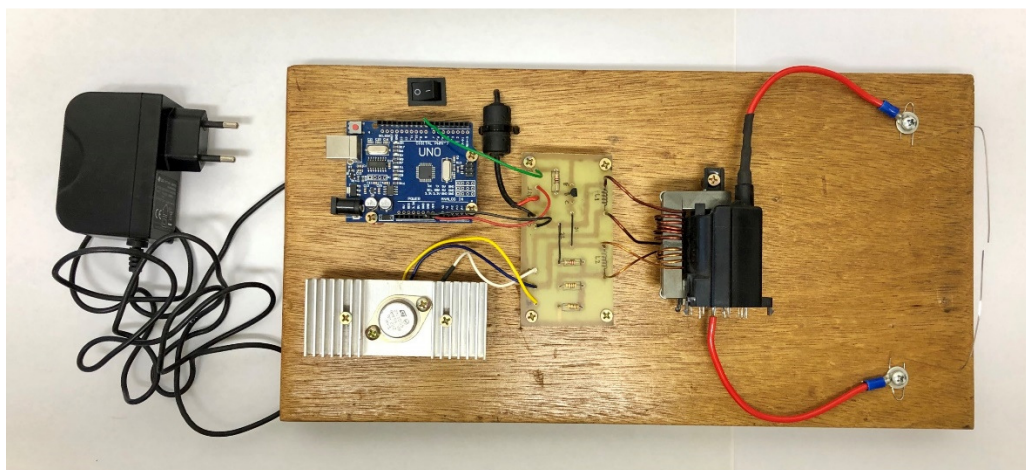


Figura 1: Uma possível configuração para a montagem do circuito de arco elétrico cantante.
Fonte: Projeto Show de Física da Ufes

Uma variante possível é fazer a montagem do circuito sem uso de placa de circuito impresso. É possível também montar os eletrodos utilizando diretamente os terminais do Flyback, sem a necessidade do arame galvanizado, que na figura aparece parafusado nos terminais do componente citado.

Materiais

Para construir seu próprio circuito para gerar o arco elétrico cantante, você vai precisar de:

- 1- Fly Back AT2079/B9 (Pode ser qualquer modelo)
- 1 Transistor NPN 2N3055-4D fashiba
- 1 dissipador de calor para o NPN 2N3055 (pode ser substituído por uma placa de alumínio de 15x8cm e espessura 3mm)
- 1 Transistor NPN BC548/b 011
- 2 Resistores de 15 Ω
- 1 Resistor de 1k Ω
- 1 Resistor de 220 Ω
- 1 Placa Arduíno
- Fio esmaltado ou encapado de 1,5 mm, 43,5cm de comprimento
- Fio esmaltado ou encapado de 0,5 mm, 48,5cm de comprimento
- Fio de 0,5 mm, de cores diferentes, de 25cm de comprimento
- 2 Jumper macho/macho
- 2 Pedacos de arame galvanizado
- 2 parafusos soberbos
- Base de madeira 35x10cm
- Cola quente

- Ferro de solda
- Alicate de bico
- Uma fonte cc de 12V, de até 3A
- Placa de fenolite para circuito impresso 10x5cm (opcional)
- Percloroeto de Ferro (opcional)
- Caneta para escrever em CD (opcional)

Montagem

A proposição deste projeto de arco elétrico cantante, adaptou o layout sugerido pelo Prof Marlon Nardi, no canal do YouTube <https://www.youtube.com/watch?v=tosK9r2StOk>. O passo a passo para uma montagem do referido professor pode ser conferida no intervalo de 6:30min até 22:00min do vídeo. O esquema do circuito elétrico é mostrado na Figura 2 abaixo.

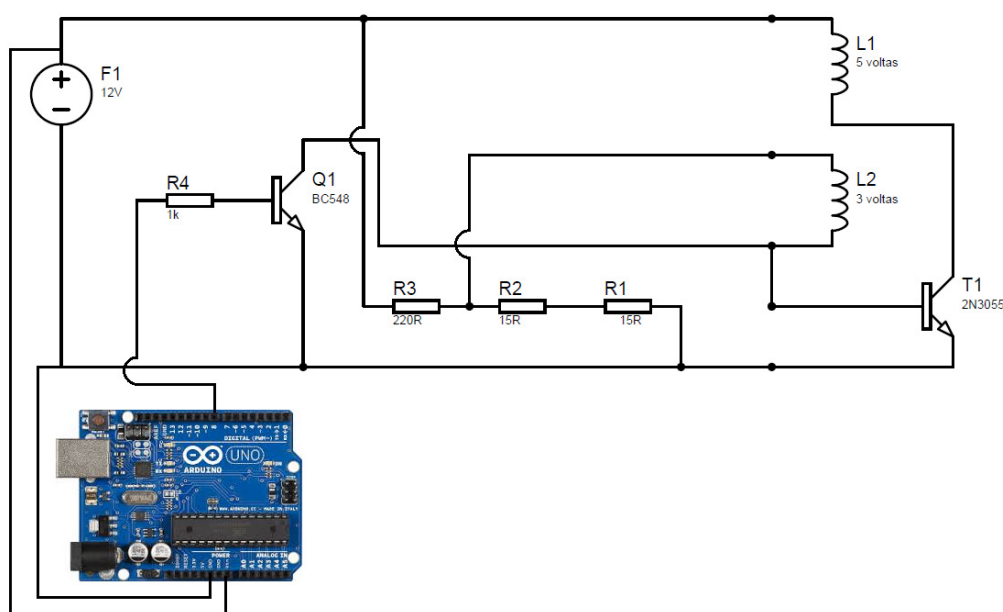


Figura 2: Circuito elétrico mostrando esquematicamente a ligação dos componentes.

Fonte: Projeto Show de Física da Ufes

Na Figura 2, As ligações na placa Arduino foram feitas da seguinte forma: o polo positivo da bateria está ligado no pino *Vin*, o negativo no pino *GND* e o resistor R4 (1k Ω) ao pino 8. Caso você decida também montar o experimento utilizando uma placa de circuito impresso, é disponibilizado abaixo a desenho com as trilhas que devem ser repassadas à placa de fenolite, para depois fazer a corrosão com percloroeto de ferro.

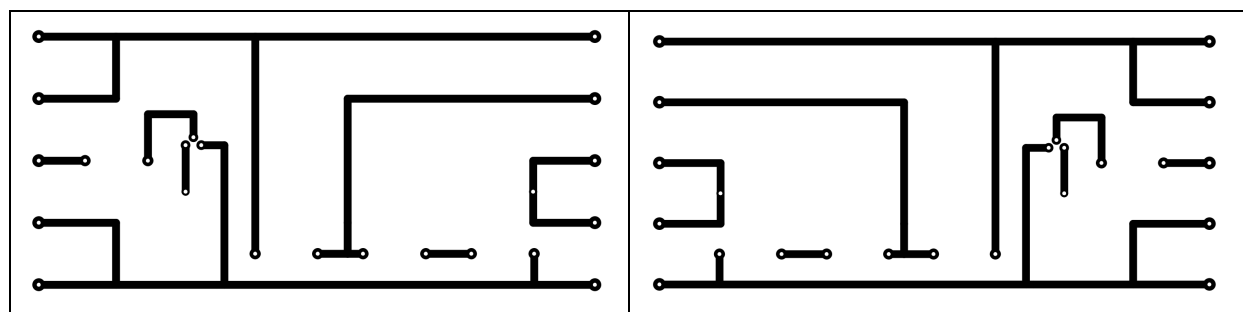


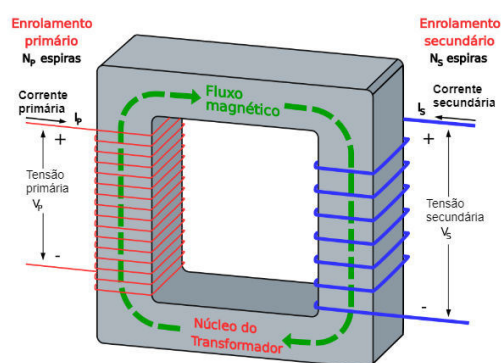
Figura 3: A parte da direita é a visão frontal do circuito, tal como ele deve ser visualizado após a corrosão na placa de fenolite. A parte da esquerda é o espelhamento do circuito, que é útil para repassar as trilhas à placa de fenolite por aquecimento, usando ferro de passar roupa, por exemplo.

Fonte: Projeto Show de Física da Ufes

Como o experimento funciona?

A forma mais fácil é entender o funcionamento das várias partes do circuito. O flyback é uma espécie de transformador, com conjuntos de bobinas internas que convertem 12 volts em corrente contínua, que é a tensão de entrada, em uma tensão aproximada de 10 a 30 mil volts, com uma corrente de saída de aproximadamente 4 a 15 mA. Ele é utilizado em televisores ou monitores antigos de computador. Quando ele está sendo utilizado pelos televisores ele opera numa frequência de 15 kHz, graças ao circuito da TV, e quando é utilizado em monitores de computador ele opera entre 30 e 150kHz.

A fonte de 12VDC está ligada em série à bobina primária L1 (5 voltas de fio 1,5mm² 14 AWG) e ao transistor de alta potência 2N3055. Quando a fonte é ligada, passa corrente na bobina L1, gerando um campo que irá induzir corrente na bobina secundária do flyback, mas também induzirá corrente na bobina L2. O esquema ao lado ilustra o esquema de indução entre o primário e o secundário de um transformador.



Fonte: <https://www.todamateria.com.br/inducaoeletroneutica/>

Como a base do transistor 2N3055 está ligada à bobina secundária L2 (3 voltas de fio 0,5mm² 20 AWG), ela vai polarizar o transistor, impedindo a passagem de corrente em L1. Portanto, ela constitui uma espécie de interruptor de liga e desliga da primeira bobina. Quando ele desliga, corta a corrente em L1, fazendo com a tensão em L2 também se anule, despolarizando a base de 2N3055.

A corrente volta a passar em L1, recomeçando o processo. Este processo cria uma tensão pulsante de frequência muito alta no primário do flyback. Esse oscilador é conhecido como oscilador *Hartley*. No secundário do flyback será produzida uma alta tensão em alta frequência. Se aproximarmos o terminal de alta tensão em alta frequência do flyback com outro terminal aterrado, vamos ver a formação de um arco elétrico.

Como o arco produz o som?

O Arduino exerce a função do modulador de áudio. Na Figura 2, observe que o pino 8 da placa de Arduino está ligada na base do transistor bc548. O coletor e emissor deste, por sua vez, está ligado em paralelo com o circuito que alimenta a base do transistor 2N3055. No pino 8 do Arduino é produzido um sinal de saída de tensão variando a intensidade na frequência de alguma música conhecida, por exemplo. Esse sinal, ligado em paralelo, acaba modulando a tensão na base do transistor 2N3055, provocando a modulação da frequência de geração do arco. Por isso, é possível escutar a música no arco elétrico produzido entre os dois terminais do flyback.

Aplicações

O funcionamento deste experimento tem similaridades com um central de transmissão de informações por ondas eletromagnéticas, tal como ocorre com o rádio, a televisão e o celular. Nestes casos, o circuito de transmissão produz um campo eletromagnético capaz de agitar as cargas elétricas ao seu redor. O alcance do sinal depende da potência da rádio. Os átomos da antena do aparelho receptor (rádio, celular, TV) sentem essa agitação, que é interpretada por um circuito receptor, que promove a decodificação da informação que foi transmitida na torre. Cada estação de rádio funciona em uma frequência específica e a utiliza para fazer a modulação e transmitir as informações. No caso do circuito aqui proposto, é o sinal de saída do Arduino que está gerando as informações que são reproduzidas no arco elétrico.

Como programar o Arduíno para esta modulação?

Quando um código é inserido no Arduíno, os pinos podem receber e processar sinais elétricos e também podem gerar sinais elétricos que podem controlar outros componentes. No caso deste experimento, o código foi desenvolvido para gerar pulsos elétricos no pino 8, com frequência e tempo de duração característico de alguma música escolhida. No site <https://www.marlonnardi.com/p/bobina-de-tesla-que-toca-muisecas-com.html> estão disponíveis três códigos para serem usados diretamente no Arduíno, para as músicas:

- ✓ Música 1: Sweet Child O Mine
- ✓ Música 2: Marcha Imperial - Star Wars
- ✓ Música 3: Melodia do Mario Theme

Para funcionar, basta copiar o código (tem que ser um de cada vez) para a IDE do Arduíno e em seguida compilar. Ao executar, a música já deverá ser produzida no arco elétrico. Os códigos também estão disponibilizados no arquivo .DOCX anexo a este PDF.

Fonte <https://www.marlonnardi.com/p/bobina-de-tesla-que-toca-miscas-com.html>

Música 1: Sweet Child O Mine

```
#include "pitches.h"

//Sweet Child O Mine- Guns N Roses
-----
int mainRiffD[] = {NOTE_D4 , NOTE_D5 , NOTE_A4, NOTE_G4, NOTE_G5, NOTE_A4, NOTE_FS5, NOTE_A4};
int mainRiffE[] = {NOTE_E4 , NOTE_D5 , NOTE_A4, NOTE_G4, NOTE_G5, NOTE_A4, NOTE_FS5, NOTE_A4};
int mainRiffG[] = {NOTE_G4 , NOTE_D5 , NOTE_A4, NOTE_G4, NOTE_G5, NOTE_A4, NOTE_FS5, NOTE_A4};

int mainRiffDurations[] = {
//d4 d5 a4 g4 g5 g4 fs5 a4
  6, 6, 6, 6, 6, 6, 6, 6};
//-----

void setup() {

for(int introTwoTimes = 0; introTwoTimes < 2; introTwoTimes++){
  for(int dTwice = 0; dTwice < 2; dTwice++){
    for (int thisNote = 0; thisNote < 8; thisNote++){
      int mainRiffDuration = 1000/mainRiffDurations[thisNote];
      tone(8, mainRiffD[thisNote], mainRiffDuration);
      int pauseBetweenNotes = mainRiffDuration * 1.30;
      delay(pauseBetweenNotes);
      noTone(8);
    }
  }
}

for(int eTwice = 0; eTwice < 2; eTwice++){
  for (int thisNote = 0; thisNote < 8; thisNote++){
    int mainRiffDuration = 1000/mainRiffDurations[thisNote];
    tone(8, mainRiffE[thisNote], mainRiffDuration);
    int pauseBetweenNotes = mainRiffDuration * 1.30;
    delay(pauseBetweenNotes);
    noTone(8);
  }
}

for(int gTwice = 0; gTwice < 2; gTwice++){
  for (int thisNote = 0; thisNote < 8; thisNote++){
    int mainRiffDuration = 1000/mainRiffDurations[thisNote];
    tone(8, mainRiffG[thisNote], mainRiffDuration);
    int pauseBetweenNotes = mainRiffDuration * 1.30;
    delay(pauseBetweenNotes);
    noTone(8);
  }
}

for(int dTwice = 0; dTwice < 2; dTwice++){
  for (int thisNote = 0; thisNote < 8; thisNote++){
    int mainRiffDuration = 1000/mainRiffDurations[thisNote];
    tone(8, mainRiffD[thisNote], mainRiffDuration);
    int pauseBetweenNotes = mainRiffDuration * 1.30;
    delay(pauseBetweenNotes);
    noTone(8);
  }
}
}

while(1);
}

void loop() {
```

```
}
```

Música 2: Marcha Imperial - Star Wars

```
int speakerPin = 8;

#define c 261
#define d 294
#define e 329
#define f 349
#define g 391
#define gS 415
#define a 440
#define aS 455
#define b 466
#define cH 523
#define cSH 554
#define dH 587
#define dSH 622
#define eH 659
#define fH 698
#define fSH 740
#define gH 784
#define gSH 830
#define aH 880

void setup()
{
    pinMode(speakerPin, OUTPUT);
}

void loop()
{
    march();
}

void beep (unsigned char speakerPin, int frequencyInHertz, long timeInMilliseconds)
{
    int x;
    long delayAmount = (long)(1000000/frequencyInHertz);
    long loopTime = (long)((timeInMilliseconds*1000)/(delayAmount*2));
    for (x=0;x<loopTime;x++)
    {
        digitalWrite(speakerPin,HIGH);
        delayMicroseconds(delayAmount);
        digitalWrite(speakerPin,LOW);
        delayMicroseconds(delayAmount);
    }

    delay(20);
}

void march()
{
    beep(speakerPin, a, 500);
    beep(speakerPin, a, 500);
    beep(speakerPin, a, 500);
    beep(speakerPin, f, 350);
    beep(speakerPin, cH, 150);
```

```
beep(speakerPin, a, 500);  
beep(speakerPin, f, 350);  
beep(speakerPin, cH, 150);  
beep(speakerPin, a, 1000);
```

```
beep(speakerPin, eH, 500);  
beep(speakerPin, eH, 500);  
beep(speakerPin, eH, 500);  
beep(speakerPin, fH, 350);  
beep(speakerPin, cH, 150);
```

```
beep(speakerPin, gS, 500);  
beep(speakerPin, f, 350);  
beep(speakerPin, cH, 150);  
beep(speakerPin, a, 1000);
```

```
beep(speakerPin, aH, 500);  
beep(speakerPin, a, 350);  
beep(speakerPin, a, 150);  
beep(speakerPin, aH, 500);  
beep(speakerPin, gSH, 250);  
beep(speakerPin, gH, 250);
```

```
beep(speakerPin, fSH, 125);  
beep(speakerPin, fH, 125);  
beep(speakerPin, fSH, 250);  
delay(250);  
beep(speakerPin, aS, 250);  
beep(speakerPin, dSH, 500);  
beep(speakerPin, dH, 250);  
beep(speakerPin, cSH, 250);
```

```
beep(speakerPin, cH, 125);  
beep(speakerPin, b, 125);  
beep(speakerPin, cH, 250);  
delay(250);  
beep(speakerPin, f, 125);  
beep(speakerPin, gS, 500);  
beep(speakerPin, f, 375);  
beep(speakerPin, a, 125);
```

```
beep(speakerPin, cH, 500);  
beep(speakerPin, a, 375);  
beep(speakerPin, cH, 125);  
beep(speakerPin, eH, 1000);
```

```
beep(speakerPin, aH, 500);  
beep(speakerPin, a, 350);  
beep(speakerPin, a, 150);  
beep(speakerPin, aH, 500);  
beep(speakerPin, gSH, 250);  
beep(speakerPin, gH, 250);
```

```
beep(speakerPin, fSH, 125);  
beep(speakerPin, fH, 125);  
beep(speakerPin, fSH, 250);  
delay(250);  
beep(speakerPin, aS, 250);  
beep(speakerPin, dSH, 500);  
beep(speakerPin, dH, 250);  
beep(speakerPin, cSH, 250);
```



```

beep(speakerPin, cH, 125);
beep(speakerPin, b, 125);
beep(speakerPin, cH, 250);
delay(250);
beep(speakerPin, f, 250);
beep(speakerPin, gS, 500);
beep(speakerPin, f, 375);
beep(speakerPin, cH, 125);

beep(speakerPin, a, 500);
beep(speakerPin, f, 375);
beep(speakerPin, c, 125);
beep(speakerPin, a, 1000);

}

```

Música 3: Melodia do MARIO THEME

```

int portSpeak(8);
//melodia do MARIO THEME
int melodia[] =
{660,660,660,510,660,770,380,510,380,320,440,480,450,430,380,660,760,860,700,760,660,520,580,480,500,760,720,680,620,650,380,430,500,430,500,570,500,760,7
20,680,620,650,1020,1020,1020,380,500,760,720,680,620,650,380,430,500,430,500,570,585,550,500,380,500,500,500,
500,760,720,680,620,650,380,430,500,430,500,570,500,760,720,680,620,650,1020,1020,1020,380,500,760,720,680,620
,650,380,430,500,430,500,570,585,550,500,380,500,500,500,500,500,500,580,660,500,430,380,500,500,500,500,58
0,660,870,760,500,500,500,500,580,660,500,430,380,660,660,660,510,660,770,380};

int duracaodasnotas[] =
{100,100,100,100,100,100,100,100,100,100,80,100,100,100,80,50,100,80,50,80,80,80,100,100,100,100,80,100,10
0,100,80,50,100,80,50,80,80,80,100,100,100,100,150,150,100,100,100,100,100,100,100,100,150,200,80,80,8
0,100,100,100,100,100,150,150,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,150,150,100,1
00,100,100,100,100,100,100,100,150,200,80,80,80,100,100,100,100,150,150,100,100,100,100,100,100,100,100,
100,100,100,100,100,60,80,60,80,80,80,80,80,60,80,60,80,80,80,80,60,80,60,80,80,80,80,100,100,100,100,1
00,100,100};

void setup() {

  for (int nota = 0; nota < 156; nota++) {

    int duracaodanota = duracaodasnotas[nota];
    tone(portSpeak, melodia[nota],duracaodanota);

    int pausadepoisdasnotas[]
    ={150,300,300,100,300,550,575,450,400,500,300,330,150,300,200,200,150,300,150,350,300,150,150,500,450,400,500,3
00,330,150,300,200,200,150,300,150,350,300,150,150,500,300,100,150,150,300,300,150,150,300,150,100,220,300,100,
150,150,300,300,150,300,300,150,300,300,100,150,150,300,300,150,150,300,150,100,420,450,420,360,300,300,150,300,30
0,150,300,300,150,300,150,150,300,150,100,220,300,100,150,150,300,300,300,150,300,300,300,100,150,150,300,300,1
50,150,300,150,100,420,450,420,360,300,300,150,300,150,300,350,150,350,150,300,150,600,150,300,350,150,150,550,
325,600,150,300,350,150,350,150,300,150,600,150,300,300,100,300,550,575};
    delay(pausadepoisdasnotas[nota]);}

    noTone(portSpeak);
  }

void loop() {

}

```