# Curso Arduíno – 07/2020

## 1. Informações Gerais:

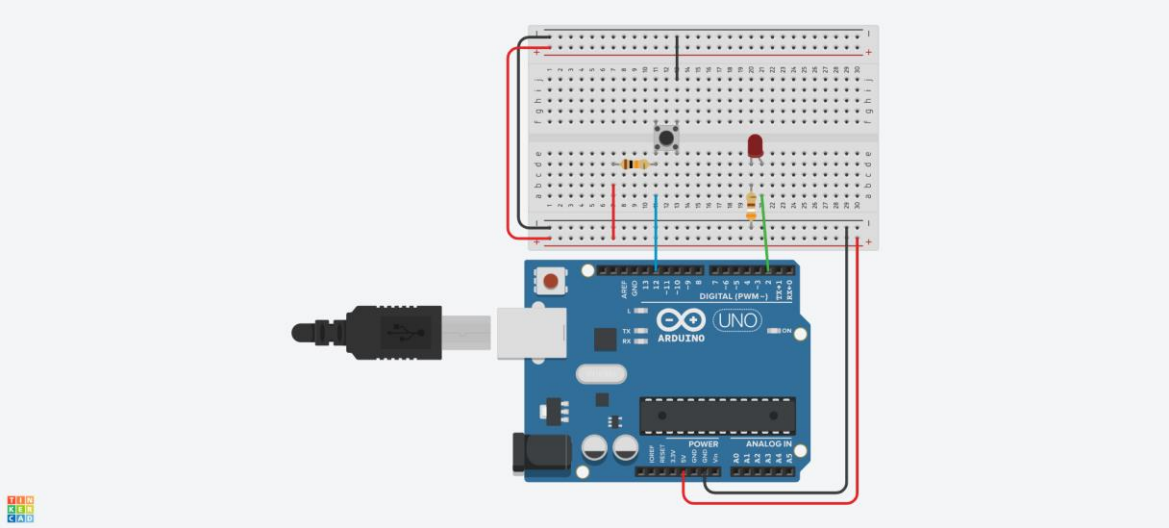
- Professor: Luís Gustavo Carvalho
- Email: ligu.carvalho@gmail.com
- Celular: (61) 99962-1845

- Site Arduíno: https://www.arduino.cc
- Download do ambiente de desenvolvimento: https://www.arduino.cc/en/Main/Software
- Referência da linguagem (comando, funções, variáveis): https://www.arduino.cc/reference/en/
- Tutoriais: https://www.arduino.cc/en/Tutorial/BuiltInExamples

- Simulador: https://www.tinkercad.com

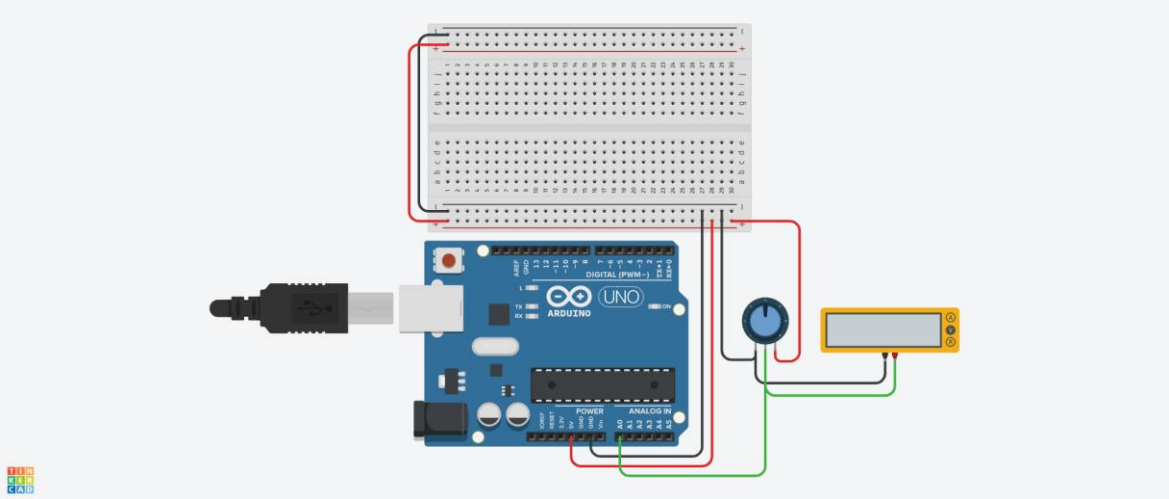
## 2. Projetos desenvolvidos

2.1. Leitura de chave e acionamento de um Led:



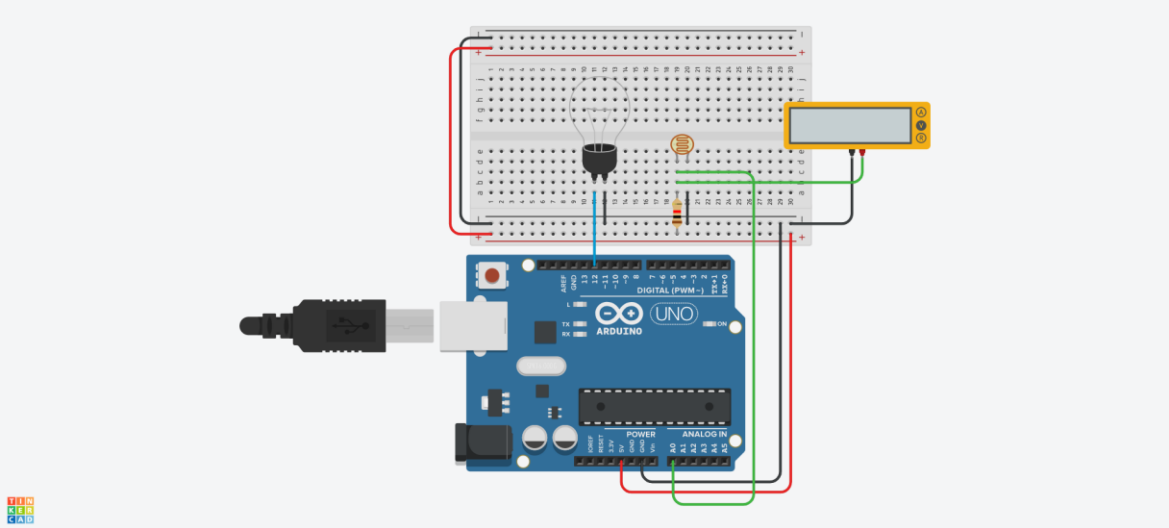

```
// Resistor do Led: 390 ohms / Resistor da Chave: 10 Kohms
void setup()
{
  // Porta 2 como saída (Led)
  pinMode(2, OUTPUT);
  // Porta 12 como entrada (Chave)
  pinMode(12, INPUT);
  // Led inicialmente apagado
  digitalWrite(2, LOW);
}

// Chave solta -> HIGH
// Chave apertada -> LOW
void loop()
{
  // Testa se a chave foi pressionada
  if (digitalRead(12) == LOW) {
    // Se sim, acende o led
    digitalWrite(2, HIGH);
  } else {
    // Caso contrário, apaga o led
    digitalWrite(2, LOW);
  }
  // Espera 100 milisegundos antes de efetuar um novo teste
  delay(100);
}
```

## 2.2. Leitura da porta analógica:



```
// Lembrar de ativar o Monitor para poder visualizar os dados transmitidos
void setup()
{
  // Ativa porta serial e configura velocidade para 9600 bps
  Serial.begin(9600);
}

void loop()
{
  // Efetua leitura da tensão do potenciômetro
  int pot = analogRead(A0);
  // Transmite leitura pela interface serial
  Serial.println(pot);
  // Aguarda 200 mseg
  delay(200);
}
```

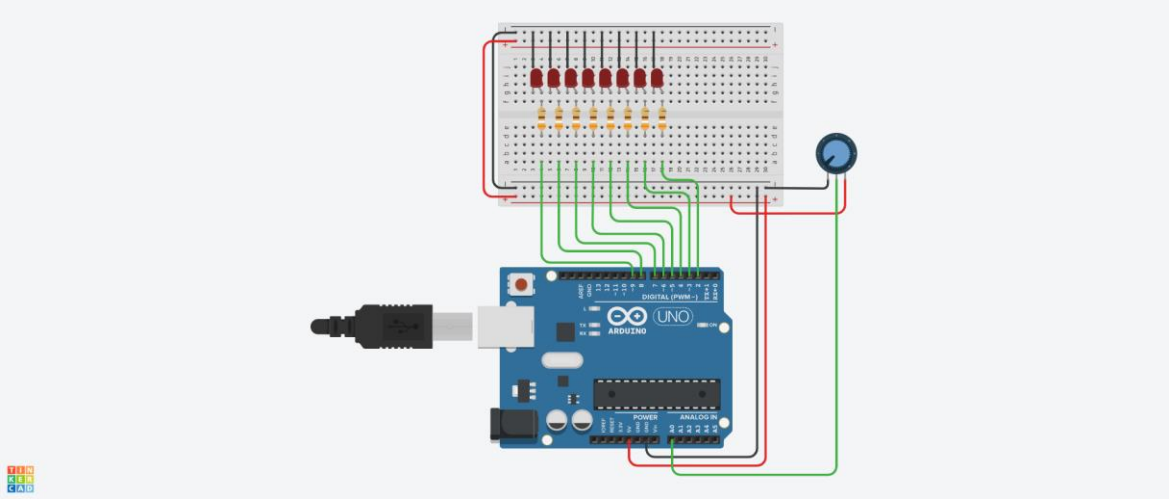
## 2.3. Lâmpada Noturna:



```
// Fotorresistor: varia sua resistência de acordo com a luminosidade
//   - Claro: baixa resistência
//   - Escuro: alta resistência
// Quando a luminosidade variar, a tensão na porta analógica irá variar
// Quanto mais escuro, maior será a tensão
// Resistor de 1 Kohms em série com o fotorresistor
// NUNCA ligar uma lâmpada diretamente na porta do Arduíno (aqui é simulador)
```

```
void setup()
{
  // Porta 12 como saída (Lâmpada)
  pinMode(12, OUTPUT);
  // Lâmpada inicialmente apagada
  digitalWrite(12, LOW);
}

void loop()
{
  // Verifica se já está escuro o suficiente (3,5 volts)
  if (analogRead(A0) > 716) {
    // Se sim, acende a lâmpada
    digitalWrite(12, HIGH);
  } else {
    // Se estiver claro, mantém a lâmpada apagada
    digitalWrite(12, LOW);
  }
  delay(200);
}
```

## 2.4. VU:

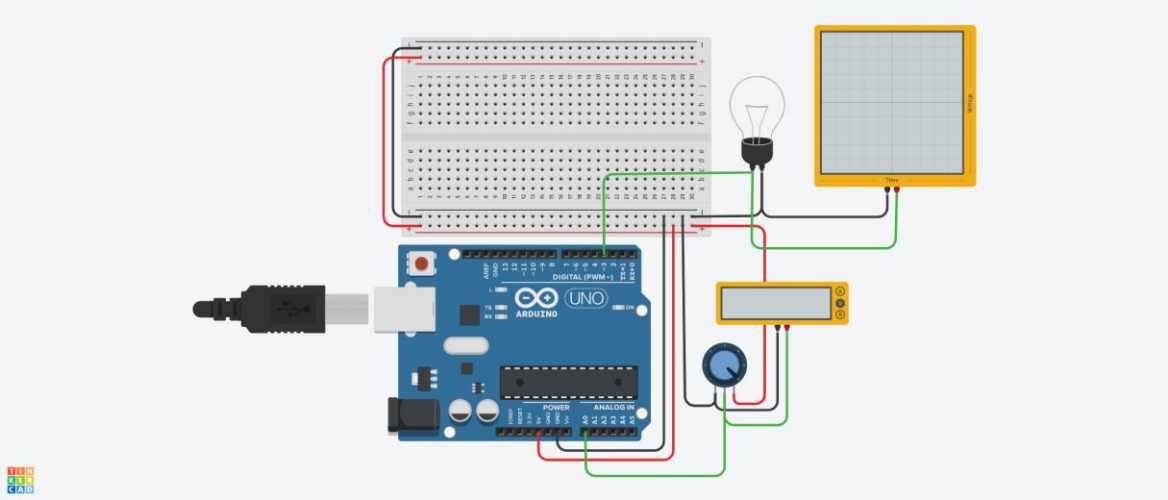


```
// Resistores de 390 ohms
void setup()
{
  // Portas utilizadas pelos Leds configuradas como saídas
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  // Leds inicialmente apagados
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
  Serial.begin(9600);
}
```

```
// Quando o potenciômetro varia de 0 a 5 volts, o valor lido na porta analógica
// varia de 0 a 1023. Dessa forma, essa faixa foi dividida em 9 faixas: de 0 a
// 113: nenhum led aceso, de 114 a 227: só um led aceso, de 228 a 341: dois leds
// acesos, e assim sucessivamente, até que todos os leds acendam
void loop()
{
  int pot = analogRead(A0);
  // Faixa 1: nenhum led aceso
  if (pot < 114) {
      digitalWrite(2, LOW);
      digitalWrite(3, LOW);
      digitalWrite(4, LOW);
      digitalWrite(5, LOW);
      digitalWrite(6, LOW);
      digitalWrite(7, LOW);
      digitalWrite(8, LOW);
      digitalWrite(9, LOW);
  // Faixa 2: só um led aceso
  } else if (pot < 228) {
      digitalWrite(2, HIGH);
      digitalWrite(3, LOW);
      digitalWrite(4, LOW);
      digitalWrite(5, LOW);
      digitalWrite(6, LOW);
      digitalWrite(7, LOW);
      digitalWrite(8, LOW);
      digitalWrite(9, LOW);
  // Faixa 3: dois leds acesos
  } else if (pot < 342){
      digitalWrite(2, HIGH);
      digitalWrite(3, HIGH);
      digitalWrite(4, LOW);
      digitalWrite(5, LOW);
      digitalWrite(6, LOW);
      digitalWrite(7, LOW);
      digitalWrite(8, LOW);
      digitalWrite(9, LOW);
  // Faixa 4: três leds acesos
  } else if (pot < 456){
      digitalWrite(2, HIGH);
      digitalWrite(3, HIGH);
      digitalWrite(4, HIGH);
      digitalWrite(5, LOW);
      digitalWrite(6, LOW);
      digitalWrite(7, LOW);
      digitalWrite(8, LOW);
      digitalWrite(9, LOW);
  // Faixa 5: quatro leds acesos
  } else if (pot < 570){
      digitalWrite(2, HIGH);
      digitalWrite(3, HIGH);
      digitalWrite(4, HIGH);
      digitalWrite(5, HIGH);
      digitalWrite(6, LOW);
      digitalWrite(7, LOW);
      digitalWrite(8, LOW);
      digitalWrite(9, LOW);
  // Faixa 6: cinco leds acesos
  } else if (pot < 684){
      digitalWrite(2, HIGH);
      digitalWrite(3, HIGH);
      digitalWrite(4, HIGH);
      digitalWrite(5, HIGH);
      digitalWrite(6, HIGH);
      digitalWrite(7, LOW);
```

```
      digitalWrite(8, LOW);
      digitalWrite(9, LOW);
  // Faixa 7: seis leds acesos
  } else if (pot < 798){
      digitalWrite(2, HIGH);
      digitalWrite(3, HIGH);
      digitalWrite(4, HIGH);
      digitalWrite(5, HIGH);
      digitalWrite(6, HIGH);
      digitalWrite(7, HIGH);
      digitalWrite(8, LOW);
      digitalWrite(9, LOW);
  // Faixa 8: sete leds acesos
  } else if (pot < 912){
      digitalWrite(2, HIGH);
      digitalWrite(3, HIGH);
      digitalWrite(4, HIGH);
      digitalWrite(5, HIGH);
      digitalWrite(6, HIGH);
      digitalWrite(7, HIGH);
      digitalWrite(8, HIGH);
      digitalWrite(9, LOW);
  // Faixa 9: todos os leds acesos
  } else {
      digitalWrite(2, HIGH);
      digitalWrite(3, HIGH);
      digitalWrite(4, HIGH);
      digitalWrite(5, HIGH);
      digitalWrite(6, HIGH);
      digitalWrite(7, HIGH);
      digitalWrite(8, HIGH);
      digitalWrite(9, HIGH);
  }
  // Serial.println(pot);
  // delay(500);
}
```

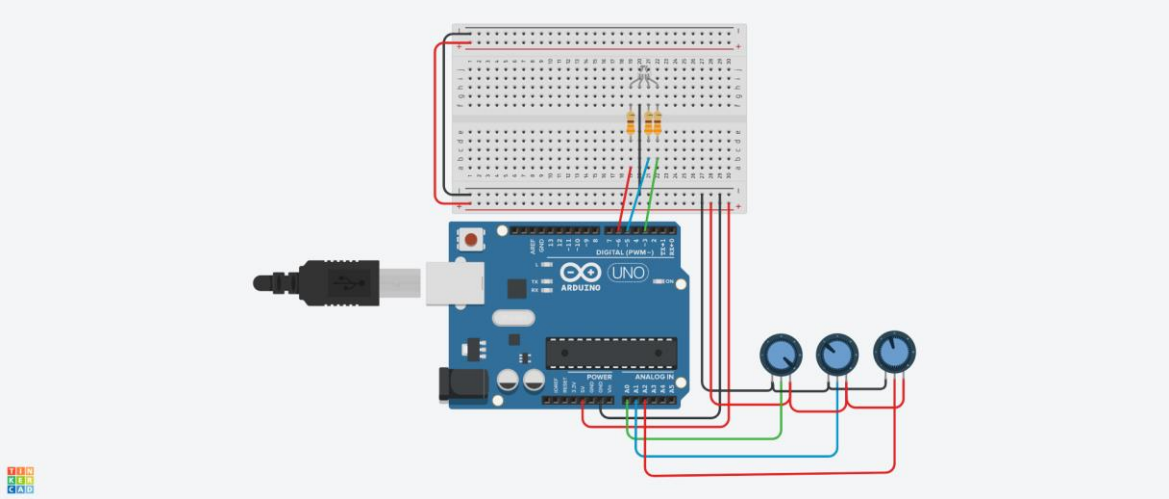
## 2.5. Controle do brilho de uma lâmpada por PWM:




```
// Configurar o osciloscópio para um TEMPO POR DIVISÃO de 500 useg
// As portas que podem gerar sinal PWM possuem um ~ na frente do seu número
void setup()
{
  // Porta 3 como saída (Lâmpada)
  pinMode(3, OUTPUT);
  // Lâmpada inicialmente apagada
  digitalWrite(3, LOW);
}
```

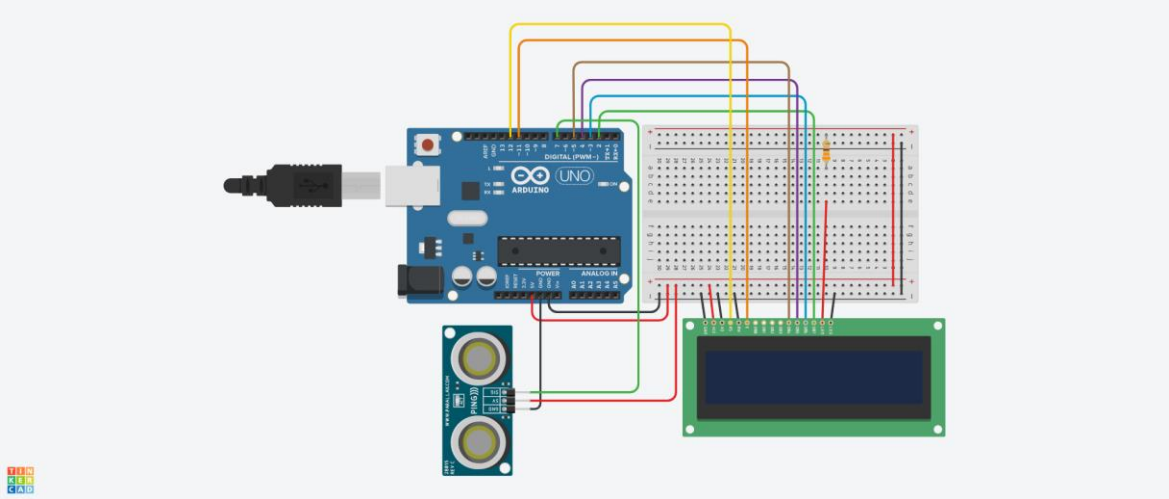
```
void loop()
{
  // Lê tensão do potenciômetro e divide por 4 para que
  // o valor fique ente 0 e 255
  int pwm = analogRead(A0) / 4;
  // Gera sinal PWM
  analogWrite(3, pwm);
  // Aguarda 500 mseg;
  delay(500);
}
```

## 2.6. Controle de um Led RGB por PWM:



```
// Resistores de 330 ohms
void setup()
{
  pinMode(3, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
}

void loop()
{
  // Lê potenciômetros e converte valores lidos para 8 bits
  int pot_vd = analogRead(A0)/4;
  int pot_az = analogRead(A1)/4;
  int pot_vm = analogRead(A2)/4;
  // Controla a intensidade de cada uma das cores do Led RGB
  analogWrite(3, pot_vd);
  analogWrite(5, pot_az);
  analogWrite(6, pot_vm);
  delay(200); // 200 mseg
}
```

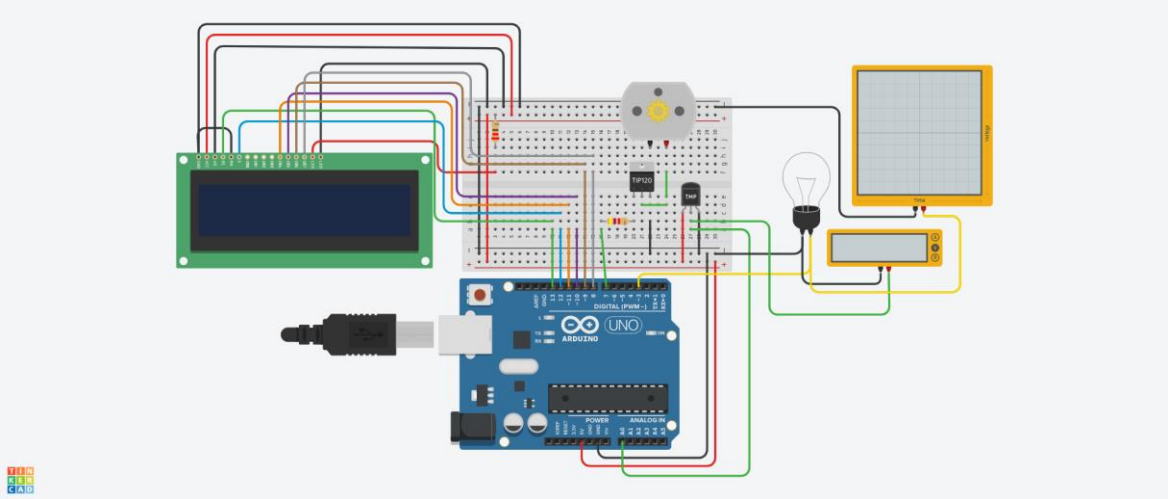
2.7. Trena Eletrônica:



```
#include <LiquidCrystal.h>

// LiquidCrystal lcd(RS, ENB, D4, D5, D6, D7);
// RS = Register Select (Seleção de registro)
// ENB = Enable (sinal de ativação)
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

byte ocirc[8] = {
  B00100,
  B01010,
  B01110,
  B10001,
  B10001,
  B10001,
  B01110,
  B00000
};

byte acirc[8] = {
  B00100,
  B01010,
  B01110,
  B00001,
  B01111,
  B10001,
  B01111,
  B00000
};

void setup()
{
  // Define o caracter com endereço 0
  lcd.createChar(0, ocirc);
  // Define o caracter com endereço 0
  lcd.createChar(1, acirc);
  lcd.begin(16, 2); // 16 colunas e 2 linhas
  lcd.print("Trena Eletr");
  lcd.write(byte(0));
  lcd.print("nica");
  lcd.setCursor(0, 1); // 1a coluna, 2a linha
  lcd.print("Dist");
  lcd.write(byte(1));
  lcd.print("ncia: ");
}
```

```
void loop()
{
  int pot = analogRead(A0);
  lcd.setCursor(11, 1); // 12a coluna, 2a linha
  // Lê o tempo gasto (em microssegundos) para o sinal bater em um anteparo
  // e voltar. A divisão por 58 converte o valor lido em cm
  long Dist = LeTempoUltrasom(7) / 58;
  lcd.print(Dist);
  lcd.print("  ");
  delay(400); // 400 mseg
}

long LeTempoUltrasom(int SignalPin)
{
  pinMode(SignalPin, OUTPUT);
  digitalWrite(SignalPin, LOW); // Garante saída LOW
  delayMicroseconds(2);
  // Gera pulso por 10 useg
  digitalWrite(SignalPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(SignalPin, LOW);
  pinMode(SignalPin, INPUT);
  // Lê o tempo gasto para o sinal retornar
  return pulseIn(SignalPin, HIGH);
}
```

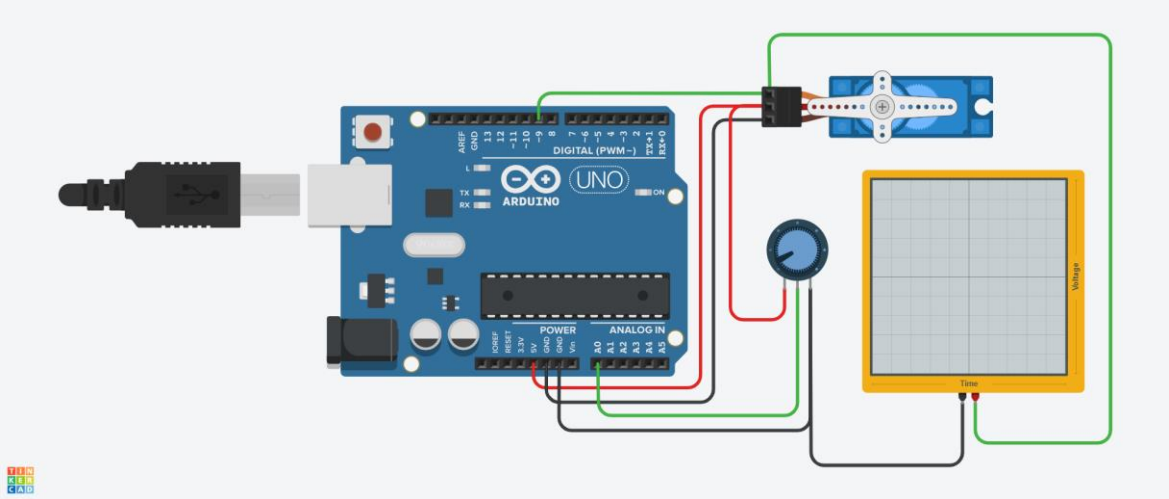
## 2.8. Chocadeira Eletrônica:




```
// Configurar o osciloscópio para um TEMPO POR DIVISÃO de 500 useg
// Resistor do display: 220 ohms
// Resistor do transistor TIP120: 4,7 Kohms

#include <LiquidCrystal.h>

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

byte graus[8] = {
  B00110,
  B01001,
  B01001,
  B00110,
  B00000,
  B00000,
  B00000,
  B00000
};
```

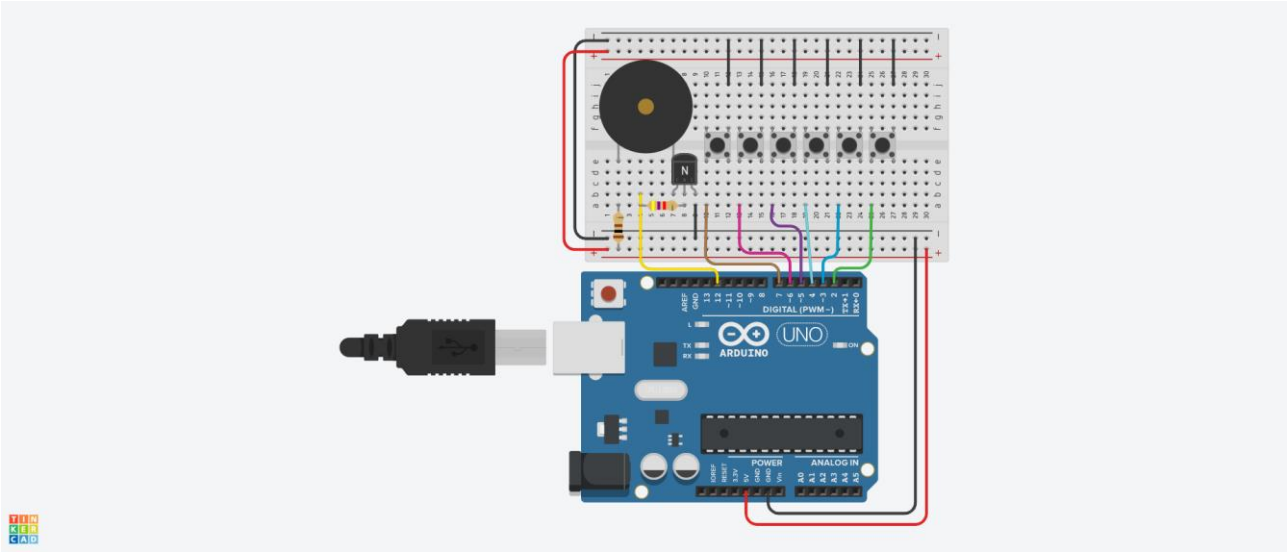
```
byte ecirc[8] = {
  B00100,
  B01010,
  B01110,
  B10001,
  B11111,
  B10000,
  B01111,
  B00000
};


void setup()
{
  pinMode(7, OUTPUT);
  pinMode(3, OUTPUT);
  digitalWrite(3, LOW);
  digitalWrite(7, LOW);
  lcd.createChar(0, graus);
  lcd.createChar(1, ecirc);
  lcd.begin(16, 2);
  lcd.print("Temp.: ");
  lcd.setCursor(0, 1);
  lcd.print("Pot");
  lcd.write(byte(1));
  lcd.print("ncia: ");
}

void loop()
{
  int temp = analogRead(A0);
  lcd.setCursor(7, 0);
  // Converte valor lido em Temperatura
  float val_temp = 0.5 * temp - 52;
  lcd.print(val_temp);
  lcd.write(byte(0));
  lcd.print("C   ");
  lcd.setCursor(10, 1);
  // Testa se a temperatura é menor do que -10°C
  if (temp < 82) {
      analogWrite(3, 255);
    digitalWrite(7, LOW);
    lcd.print("100%");
  // Testa se a temperatura está entre -10°C e 35°C
  } else if (temp >= 82 && temp <= 176) {
    int pwm = map(temp,82,176,255,0);
      analogWrite(3, pwm);
    digitalWrite(7, LOW);
    int pot = map(temp,82,176,100,0);
    lcd.print(String(pot) + "%  ");
  // Testa se a temperatura está entre 35°C e 38°C
  } else if (temp > 176 && temp < 182) {
      analogWrite(3, 0);
    digitalWrite(7, LOW);
    lcd.print("0%  ");
  // Temperatura acima de 38°C
  } else {
      analogWrite(3, 0);
    digitalWrite(7, HIGH);
    lcd.print("0%  ");
  }
  delay(100); // 100 mseg
}
```

## 2.9. Controle de um Servo-motor:




```
#include <Servo.h>

Servo Servo_M;

void setup() {
  // Indica em qual pino o servo está ligado
  Servo_M.attach(9);
}

void loop() {
  int val = analogRead(A0);
  // Converte valor lido em um ângulo de 0 a 180°
  val = map(val, 0, 1023, 0, 180);
  // Indica o ângulo para o servo
  Servo_M.write(val);
  delay(100);
}
```

## 2.10. Órgão Eletrônico:

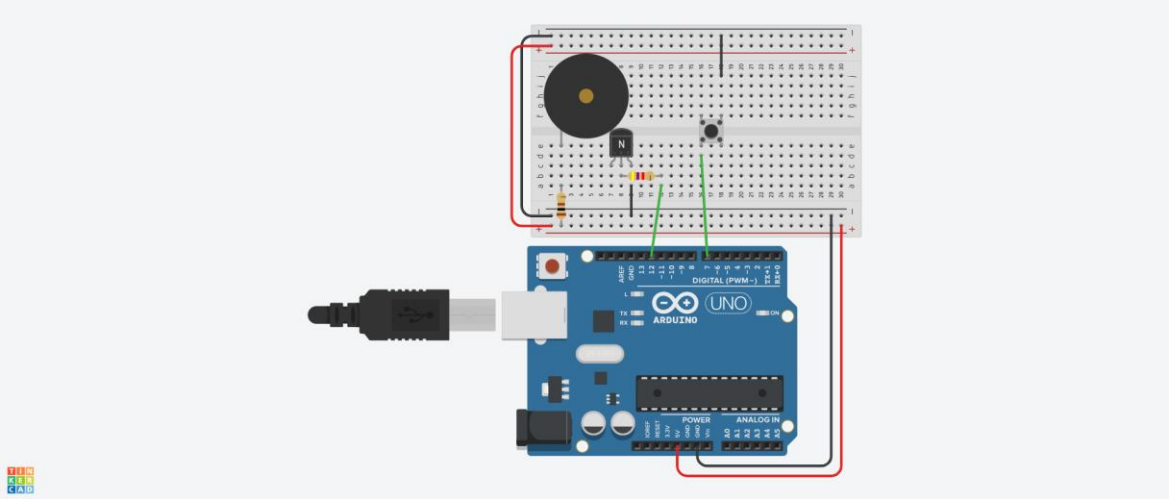


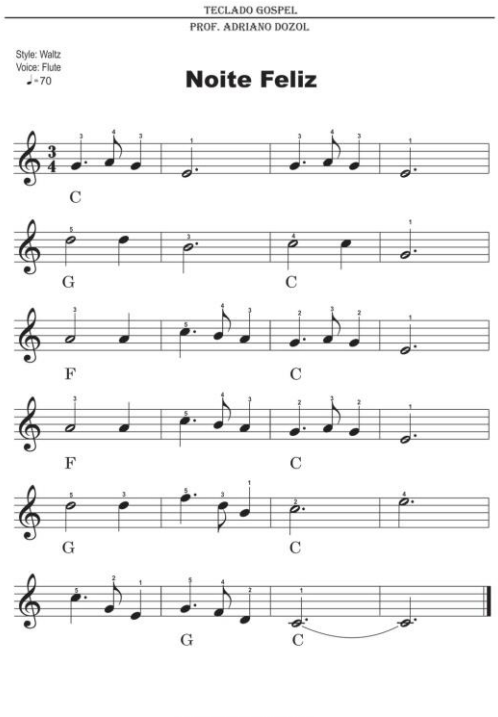
| NOTAS | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 1 | C | 32,703196 | 65,406391 | 130,81278 | 261,62557 | 523,25113 | 1046,5023 | 2093,0045 |
| 2 | C# | 34,647829 | 69,295658 | 138,59132 | 277,18263 | 554,36526 | 1108,7305 | 2217,461 |
| 3 | D | 36,708096 | 73,416192 | 146,83238 | 293,66477 | 587,32954 | 1174,6591 | 2349,3181 |
| 4 | D# | 38,890873 | 77,781746 | 155,56349 | 311,12698 | 622,25397 | 1244,5079 | 2489,0159 |
| 5 | E | 41,203445 | 82,406889 | 164,81378 | 329,62756 | 659,25511 | 1318,5102 | 2637,0205 |
| 6 | F | 43,653529 | 87,307058 | 174,61412 | 349,22823 | 698,45646 | 1396,9129 | 2793,8259 |
| 7 | F# | 46,249303 | 92,498606 | 184,99721 | 369,99442 | 739,98885 | 1479,9777 | 2959,9554 |
| 8 | G | 48,999429 | 97,998859 | 195,99772 | 391,99544 | 783,99087 | 1567,9817 | 3135,9635 |
| 9 | G# | 51,913087 | 103,82617 | 207,65235 | 415,3047 | 830,6094 | 1661,2188 | 3322,4376 |
| 10 | A | 55 | 110 | 220 | **440** | 880 | 1760 | 3520 |
| 11 | A# | 58,27047 | 116,54094 | 233,08188 | 466,16376 | 932,32752 | 1864,655 | 3729,3101 |
| 12 | B | 61,735413 | 123,47083 | 246,94165 | 493,8833 | 987,7666 | 1975,5332 | 3951,0664 |

```
// Tabela das frequências das oitavas 4 e 5
#define C4  262 //DÓ
#define CS4 277 //DÓ#
#define D4  294 //RÉ
#define DS4 311 //RÉ#
#define E4  330 //MI
#define F4  349 //FÁ
#define FS4 370 //FÁ#
#define G4  392 //SOL
#define GS4 415 //SOL#
#define A4  440 //LÁ
#define AS4 466 //LÁ#
#define B4  494 //SI
#define C5  523
#define CS5 554
#define D5  587
#define DS5 622
#define E5  659
#define F5  698
#define FS5 740
#define G5  784
#define GS5 831
#define A5  880
#define AS5 932
#define B5  988

// Resistor em série com o alto-falante: 100 ohms
// Resistor na base do transistor NPN: 4,7 Kohms
void setup() {
  pinMode(12, OUTPUT);
  // Portas das chaves como entradas com resistor de PULL-UP ativado
  pinMode(7, INPUT_PULLUP);
  pinMode(6, INPUT_PULLUP);
  pinMode(5, INPUT_PULLUP);
  pinMode(4, INPUT_PULLUP);
  pinMode(3, INPUT_PULLUP);
  pinMode(2, INPUT_PULLUP);
  digitalWrite(12, LOW);
}

void loop() {
  if (digitalRead(7) == LOW) {
    // Nota DÓ
    tone(12, C4) ;
```

```
  } else if (digitalRead(6) == LOW) {
    // Nota RÉ
    tone(12, D4) ;
  } else if (digitalRead(5) == LOW) {
    // Nota MI
    tone(12, E4) ;
  } else if (digitalRead(4) == LOW) {
    // Nota FÁ
    tone(12, F4) ;
  } else if (digitalRead(3) == LOW) {
    // Nota SOL
    tone(12, G4) ;
  } else if (digitalRead(2) == LOW) {
    // Nota LÁ
    tone(12, A4) ;
  } else {
    noTone(12);
  }
}
```

## 2.11. Caixa de Música:



```
// Tabela das frequências das oitavas 4 e 5
#define C4  262
#define CS4 277
#define D4  294
#define DS4 311
#define E4  330
#define F4  349
#define FS4 370
#define G4  392
#define GS4 415
#define A4  440
#define AS4 466
#define B4  494
#define C5  523
#define CS5 554
#define D5  587
#define DS5 622
#define E5  659
#define F5  698
#define FS5 740
#define G5  784
#define GS5 831
#define A5  880
#define AS5 932
#define B5  988

#define base 100

int melodia[] = {G4, A4, G4, E4, G4, A4, G4, E4, D5, D5, B4, C5, C5, G4};
int Duracao[] = {6, 2, 4, 12, 6, 2, 4, 12, 8, 4, 12, 8, 4, 12};

void setup() {
  pinMode(12, OUTPUT);
  pinMode(7, INPUT_PULLUP);
  digitalWrite(12, LOW);
}

void loop() {
  // Aguarda chave ser pressionada para iniciar a melodia
  if (digitalRead(7) == LOW) {
     for (int Nota = 0; Nota <= 13; Nota++) {
       int noteDuration = base * Duracao[Nota];
       tone(12, melodia[Nota], noteDuration);
       delay(noteDuration*1.3);
       noTone(12);
     }
  }
}
```