



Introdução à Robótica Arduino



Monitor serial do Arduino



O monitor serial é uma ferramenta presente na IDE do Arduino e por ela você pode acompanhar qualquer impressão ou transferência de dados via serial.

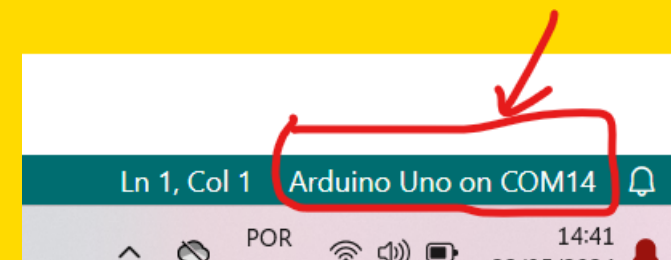
Monitor serial do Arduino



Abra a IDE do Arduino, conecte sua placa a USB e verifique se a porta de comunicação com o Arduino foi reconhecida.

A screenshot of the Arduino IDE 2.1.0 interface. The top bar shows the file name 'MonitorSerial-01 | Arduino IDE 2.1.0' and a menu bar with 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu bar is a toolbar with icons for a checkmark, a right arrow, and a play button. A dropdown menu shows 'Arduino Uno'. The main editor area displays the code for 'MonitorSerial-01.ino'. The code is as follows:

```
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8  
9 }  
10
```



Monitor serial do Arduino



Agora vamos aprender a ler dados enviados para a porta serial do Arduino e exibí-los.

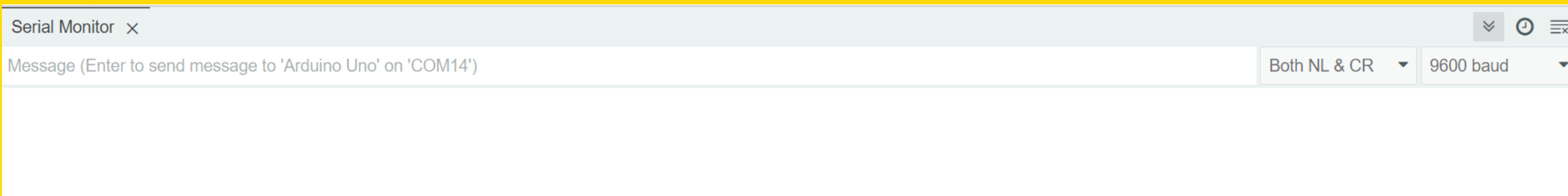
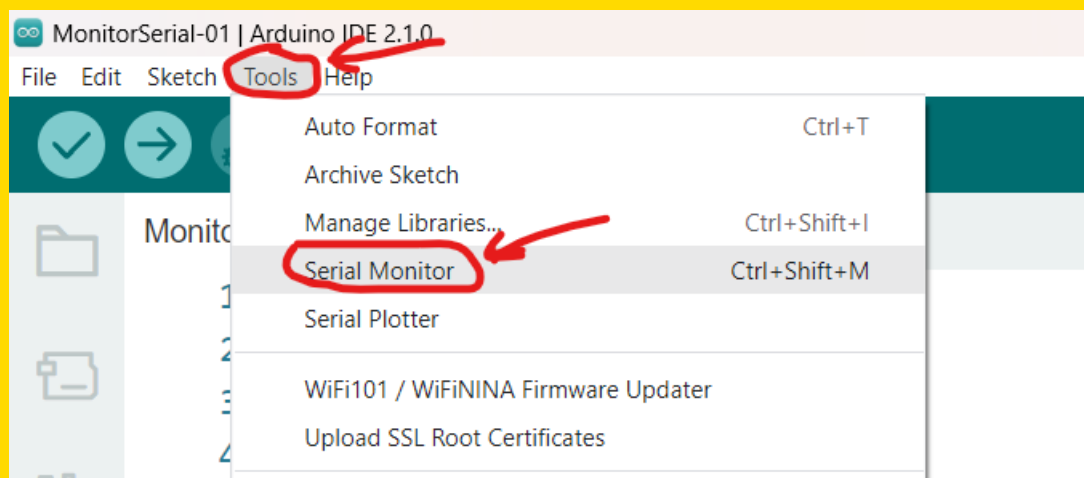
MonitorSerial-01.ino

```
1  void setup() {  
2      Serial.begin(9600);  
3  
4  }  
5  
6  void loop() {  
7      Serial.println("PEQUENOS TALENTOS");  
8      delay(1000);  
9  
10 }
```

Monitor serial do Arduino



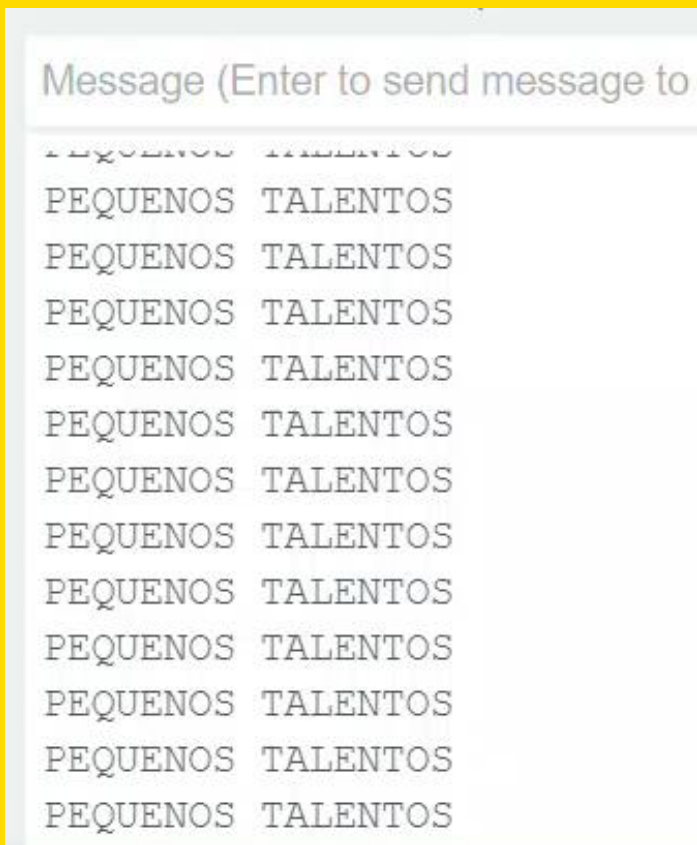
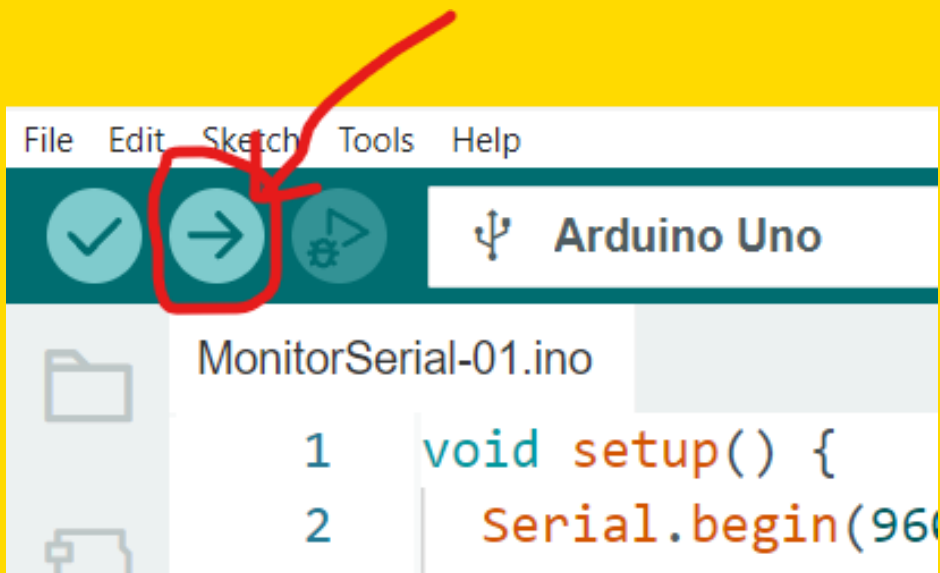
Abra a janela do Monitor Serial.





Monitor serial do Arduino

Envie o código para o Arduino



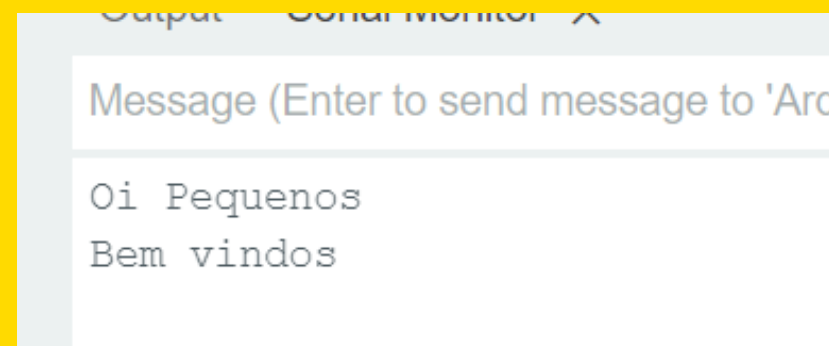
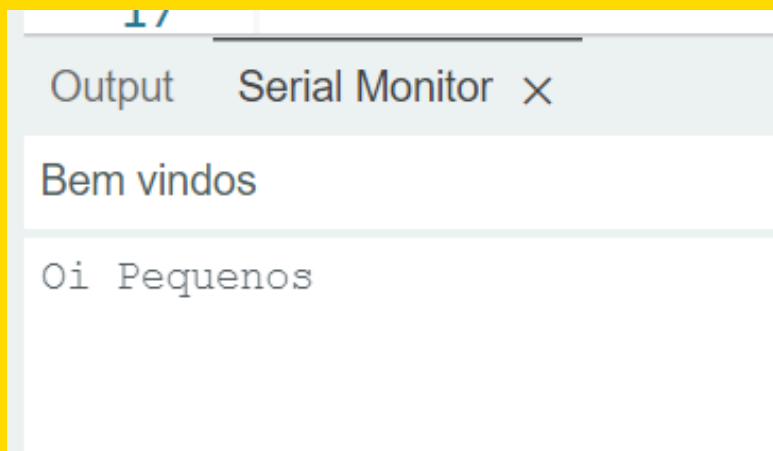


Monitor serial do Arduino

Agora vamos ver como enviar caracteres para a porta serial do Arduino

MonitorSerial-01.ino

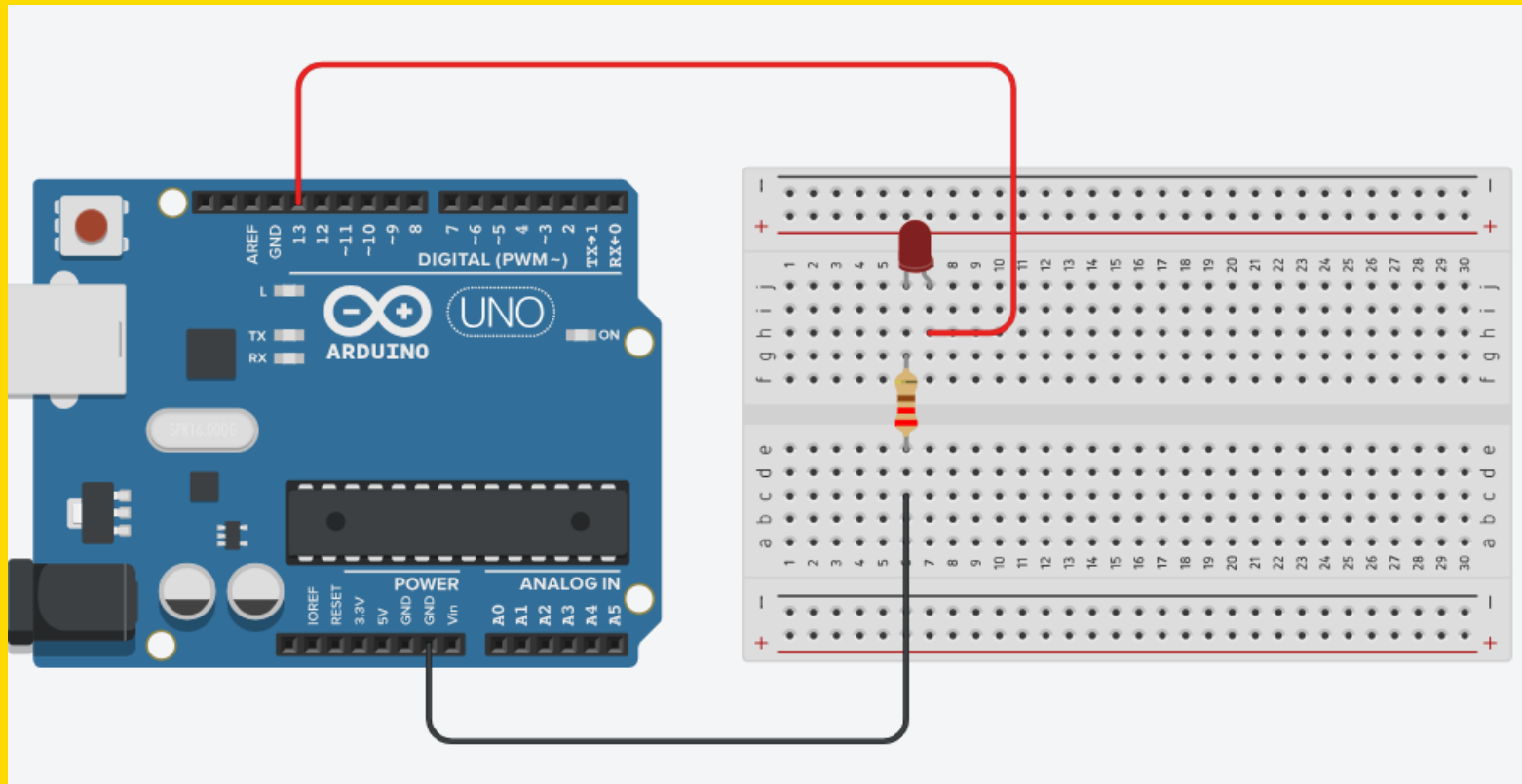
```
1 void setup() {  
2     Serial.begin(9600);  
3  
4 }  
5  
6 char c;  
7  
8 void loop() {  
9     if (Serial.available() > 0) {  
10         c = Serial.read();  
11         Serial.print(c);  
12     }  
13  
14 }
```

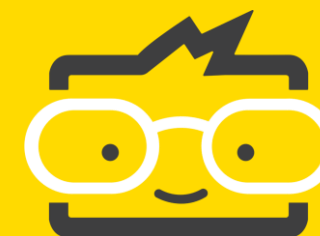


Monitor serial do Arduino



Para a parte prática vamos fazer com que um Led acenda ou apague, utilizando comandos enviados pela porta serial.





Monitor serial do Arduino

MonitorSerial-01.ino

```
1  int led = 13;
2  void setup() {
3      Serial.begin(9600);
4      pinMode(led,OUTPUT);
5
6  }
7  char c;
8  void loop() {
9      if (Serial.available()>0){
10         c = Serial.read();
11         if(c == 'L'){
12             digitalWrite(led, HIGH);
13         }
14         if(c == 'D'){
15             digitalWrite(led, LOW);
16         }
17     }
18 }
```

```
int led = 13;
void setup() {
    Serial.begin(9600);
    pinMode(led,OUTPUT);

}
char c;
void loop() {
    if (Serial.available()>0){
        c = Serial.read();
        if(c == 'L'){
            digitalWrite(led, HIGH);
        }
        if(c == 'D'){
            digitalWrite(led, LOW);
        }
    }
}
```



Monitor serial do Arduino

Depois de enviado o programa para o Arduino, abra o monitor serial e digite “L” e depois “D”.

Para pensar e testar

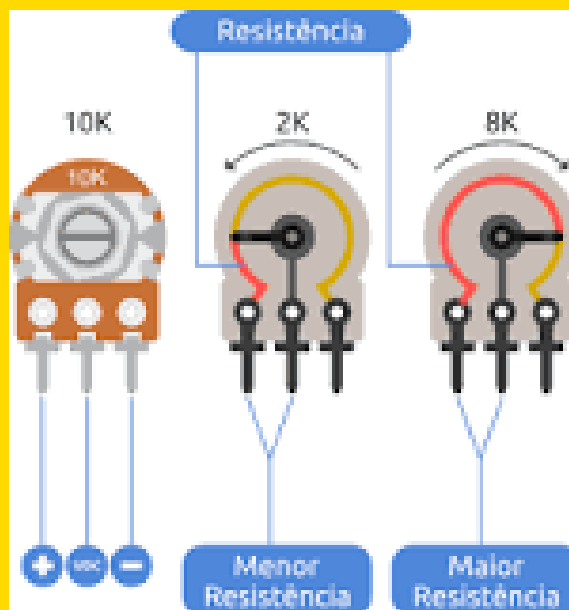
O que acontece se você digitar?

- l
- d
- ligado
- LIGADO
- LID

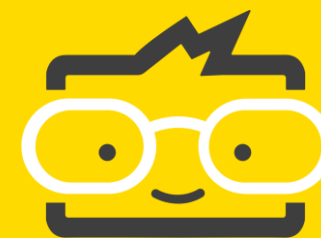


Potenciômetro

Um potenciômetro é um componente eletrônico que possui resistência elétrica ajustável (elemento resistivo), que possui normalmente três terminais onde a conexão central é deslizante e manipulável via cursor móvel

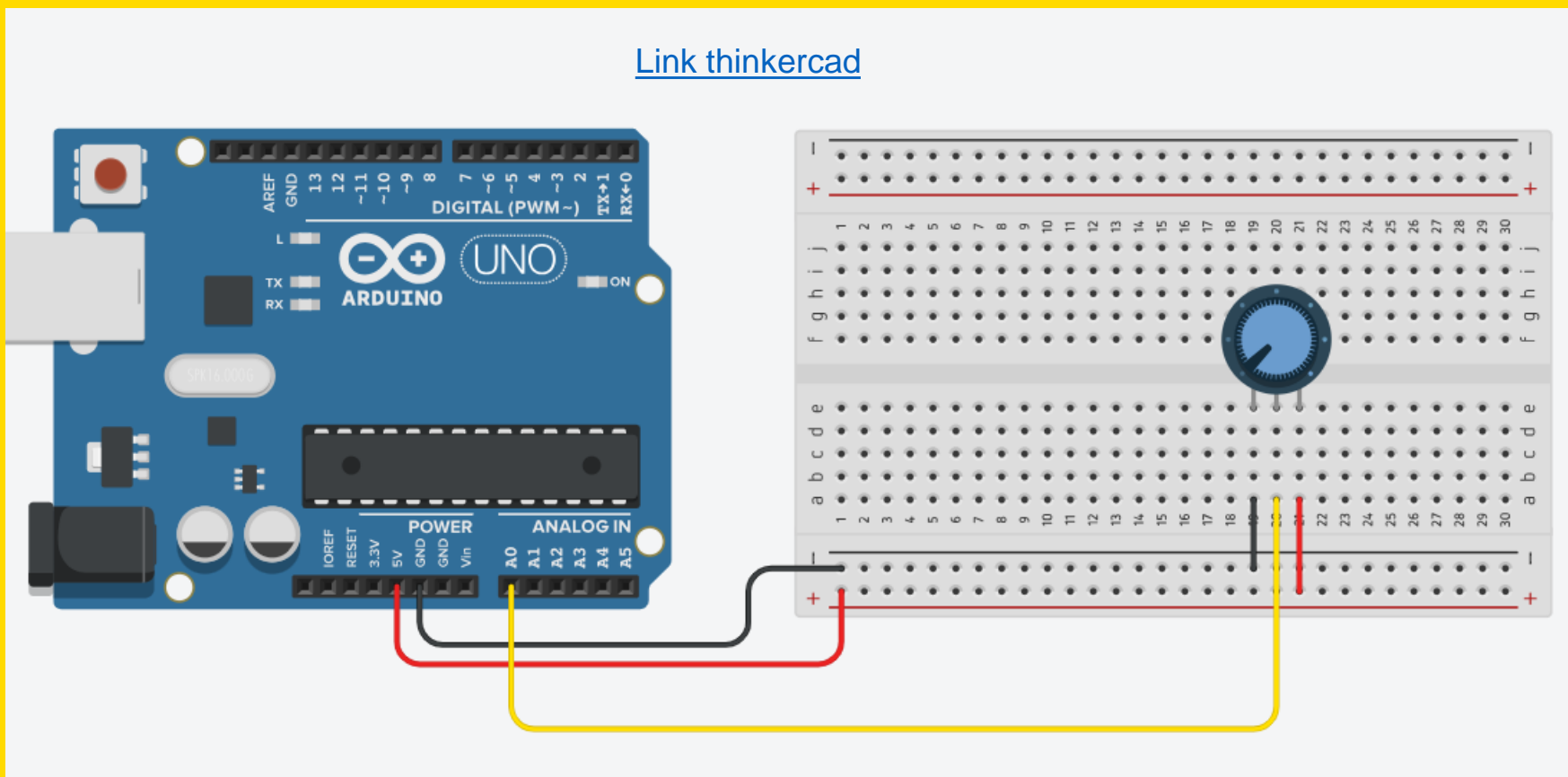


Potenciômetro



Monte o seguinte circuito

[Link thinkercad](#)



Potenciômetro



potenciometro.ino

```
1  int pinoPOT = A0; //PINO ANALÓGICO A0 UTILIZADO PELO POTENCIÔMETRO
2  int potvalor;
3  void setup()
4  {
5      Serial.begin(9600);
6      pinMode(pinoPOT, INPUT); //DEFINE O PINO COMO ENTRADA
7  }
8
9  void loop()
10 {
11     potvalor = map(analogRead(pinoPOT), 0, 1023, 0, 180);
12     Serial.print("Valor do Potenciometro = ");
13     Serial.println(potvalor);
14     delay(250);
15 }
16
```

```
int pinoPOT = A0; //PINO ANALÓGICO A0 UTILIZADO
PELO POTENCIÔMETRO
int potvalor;
void setup()
{
    Serial.begin(9600);
    pinMode(pinoPOT, INPUT); //DEFINE O PINO COMO
ENTRADA
}

void loop()
{
    potvalor = map(analogRead(pinoPOT), 0, 1023, 0,
180);
    Serial.print("Valor do Potenciometro = ");
    Serial.println(potvalor);
    delay(250);
}
```



Potenciômetro

Envie o programa para o Arduino, deixe a janela do monitor serial aberta e verifique as saídas ao movimentar o cursor do potenciômetro.

```
Valor do Potenciometro = 0  
Valor do Potenciometro = 0  
Valor do Potenciometro = 0  
Valor do Potenciometro = 0  
Valor do Potenciometro = 0  
Valor do Potenciometro = 0
```

```
Valor do Potenciometro = 64  
Valor do Potenciometro = 64  
Valor do Potenciometro = 64  
Valor do Potenciometro = 64  
Valor do Potenciometro = 64
```

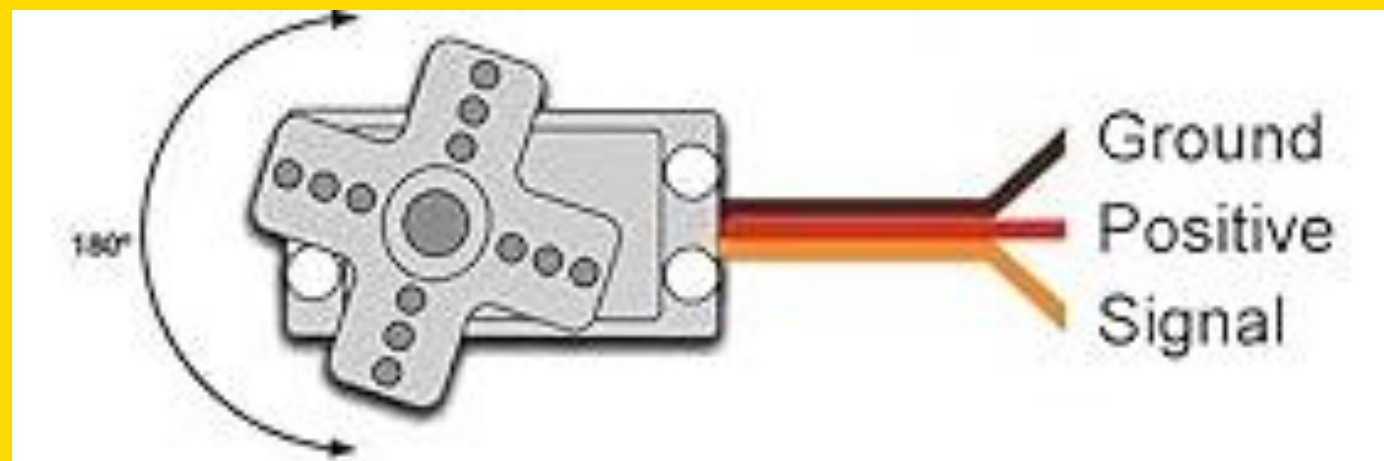
```
Valor do Potenciometro = 180  
Valor do Potenciometro = 180  
Valor do Potenciometro = 180  
Valor do Potenciometro = 180  
Valor do Potenciometro = 180
```

O programa faz com que o valor inicial seja 0 e o valor final 180. Isso é possível devido a utilização da função map que faz o mapeamento dos valores de entrada obtidos em volts e os converte para uma faixa de valores que definimos (de 0 a 180).

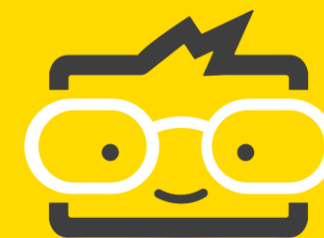
Servo Motor



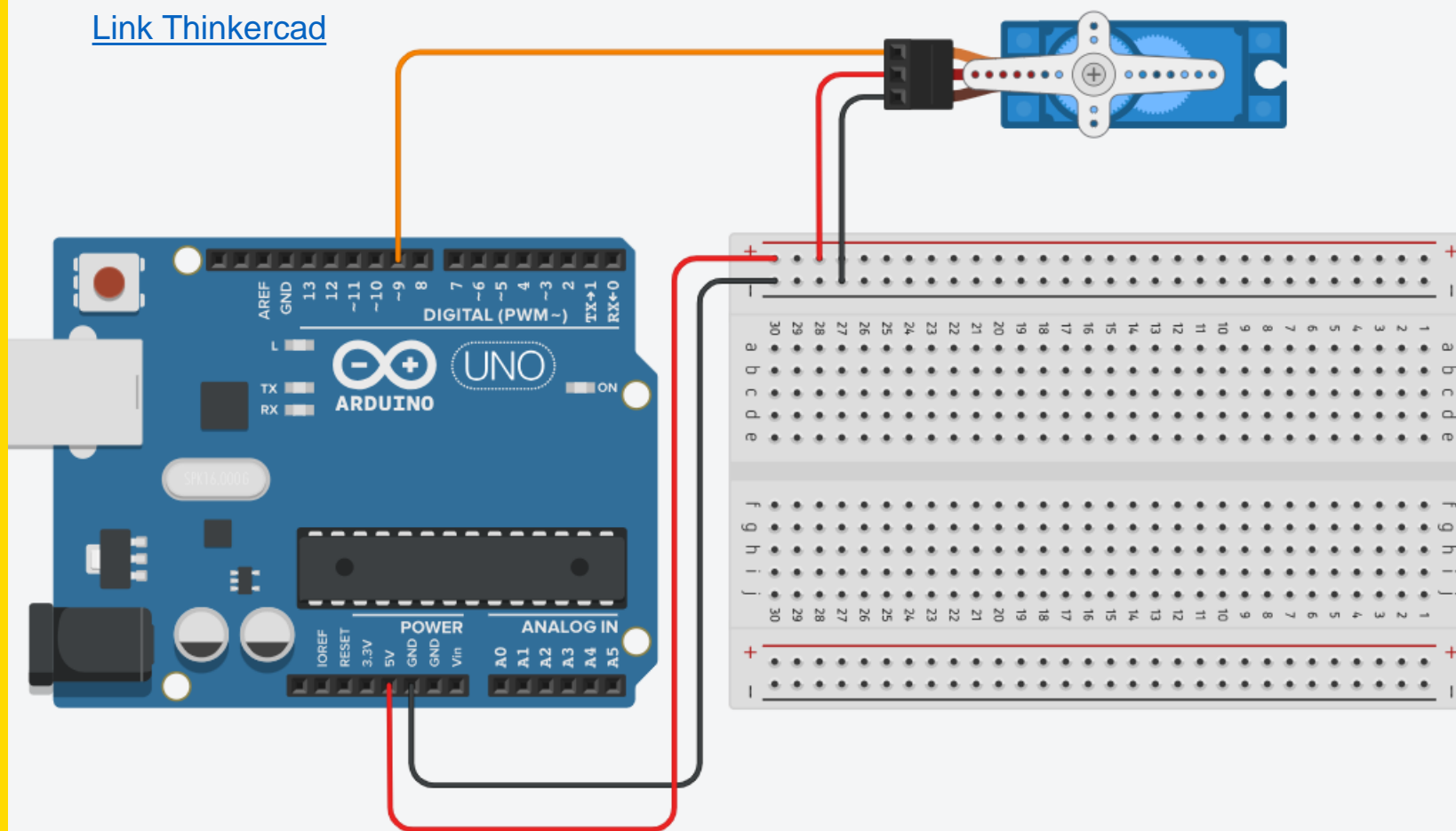
Um servo motor é um dispositivo eletromecânico que converte sinais de controle em movimento mecânico preciso. É composto por três principais componentes: um motor, um sistema de feedback e um controlador.



Servo Motor



[Link Thinkercad](#)



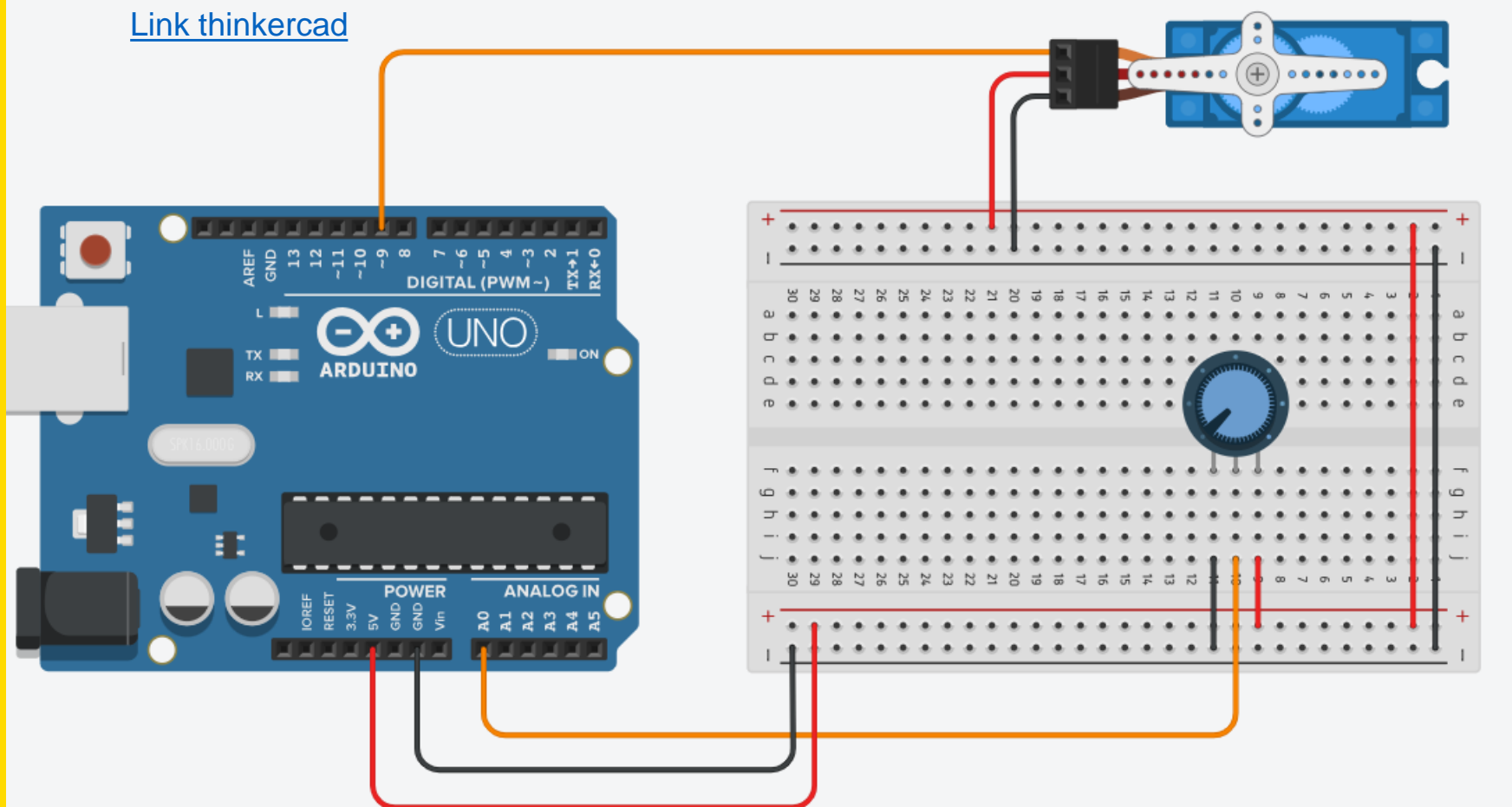
servo_motor.ino

```
1  #include <Servo.h>
2  Servo meuservo;
3
4  void setup()
5  {
6      meuservo.attach(9);
7  }
8
9  void loop()
10 {
11     meuservo.write(180);
12     delay(1800);
13     meuservo.write(0);
14     delay(1800);
15     meuservo.write(90);
16     delay(1800);
17     meuservo.write(0);
18     delay(1800);
19 }
```


Servo Motor e Potenciômetro



[Link tinkercad](#)



Servo Motor e Potenciômetro



servo_motor_potenciometro.ino

```
1  #include <Servo.h>
2  Servo meuservo;
3  int pinoPOT = A0;
4  int potvalor;
5
6  void setup()
7  {
8      Serial.begin(9600);
9      pinMode(pinoPOT, INPUT);
10     meuservo.attach(9);
11
12 }
13
14 void loop()
15 {
16     potvalor = map(analogRead(pinoPOT), 0, 1023, 0, 180);
17     Serial.print("Valor do Potenciometro = ");
18     Serial.println(potvalor);
19     meuservo.write(potvalor);
20
21 }
```