



Laboratório de Materiais e Equipamentos Elétricos

Roteiro 12: Motor de Passo

1. Equipe

Nome	RA
Gabriel Felipe Ferdinandi de Souza	2669480
Gustavo Ferreira Fonseca	2669510
Gustavo Henrique Gonçalves	2669528

2. Desenvolvimento da prática

A:

Código

```
#include <Stepper.h>
//Declarando variaveis:
const int steps_per_revolution = 2038;

Stepper motor_passos = Stepper(steps_per_revolution, 7, 5, 6, 4);
//Criando Objeto Stepper - Motor de Passos

void setup() {
```

```

Serial.begin(9600); //Iniciando a comunicação serial

motor_passos.setSpeed(3); //Define a velocidade inicial de giro do
motor
} //Fim SETUP

void loop() {

motor_passos.setSpeed(3);
motor_passos.step(steps_per_revolution);
delay(1000);
motor_passos.setSpeed(10);
motor_passos.step(-steps_per_revolution);
delay(1000);

} //Fim LOOP

```

B:

Código

```

#include <Stepper.h>
//Declarando variaveis:
const int steps_per_revolution = 2038;

Stepper motor_passos = Stepper(steps_per_revolution, 7, 5, 6, 4);
//Criando Objeto Stepper - Motor de Passos

void setup() {
    Serial.begin(9600); //Iniciando a comunicação serial

    motor_passos.setSpeed(5); //Define a velocidade inicial de giro do
motor
} //Fim SETUP

void loop() {

motor_passos.step(steps_per_revolution/2);
}

```

```
motor_passos.step(-steps_per_revolution/4);
} //Fim LOOP
```

C:

Código

```
#include <Stepper.h>
//Declarando variaveis:
const int steps_per_revolution = 2038;
#define pin_pot A0
int valor_pot;

Stepper motor_passos = Stepper(steps_per_revolution, 7, 5, 6, 4);
//Criando Objeto Stepper - Motor de Passos

void setup() {
    Serial.begin(9600); //Iniciando a comunicação serial

} //Fim SETUP

void loop() {

valor_pot = analogRead(pin_pot);

int velocidade = map(valor_pot, -200, 1000, 0, 10); //Mapeia a
velocidade do motor, usando parametros afim de evitar v = 0
motor_passos.setSpeed(velocidade); //Define velocidade do motor já
mapeada
Serial.println(velocidade);

motor_passos.step(1); //Faz apenas 1 step por ciclo para evitar
travamentos

} //Fim LOOP
```

D:

Código

```
#include <Stepper.h>
//Declarando variaveis:
const int steps_per_revolution = 2038;
#define pin_pot A0
#define pin_bot1 13
#define pin_bot2 12
#define pin_beep 10
int valor_pot;
long unsigned int tempo, tempo_anterior;
bool valor_bot1, valor_bot2, pre1, pre2, girar_teste = 0;

Stepper motor_passos = Stepper(steps_per_revolution, 7, 5, 6, 4);
//Criando Objeto Stepper - Motor de Passos

void setup(){
    Serial.begin(9600); //Iniciando a comunicação serial
    motor_passos.setSpeed(10); //Setando inicialmente velocidade para 5
RPM
    pinMode(pin_bot1, INPUT);
    pinMode(pin_bot2, INPUT);
    pinMode(pin_beep, OUTPUT);
    tempo_anterior = millis(); //Inicio dos tempos
} //Fim SETUP

void altera_velocidade(){

valor_pot = analogRead(pin_pot);
int velocidade = map(valor_pot, -200, 1075, 0, 10); //Mapeia a
velocidade do motor, usando parametros afim de evitar v = 0
motor_passos.setSpeed(velocidade);
Serial.println(velocidade);
} //Altera a velocidade pela leitura do potenciometro após chamada
várias vezes (cada 10 passos)

void beep(){
digitalWrite(pin_beep, 1);
delay(150);
```

```

digitalWrite(pin_beep, 0);
} //Fim BEEP

void girar(){
    while(girar_teste == 0){
        valor_bot1 = digitalRead(pin_bot1);
        valor_bot2 = digitalRead(pin_bot2);
        Serial.println("HAB");

        if(valor_bot1 == 1 && valor_bot2 == 1){
            tempo = millis();
            if(valor_bot1 == 1 && valor_bot2 == 1 && (tempo - tempo_anterior >
2000)){
                tempo_anterior = millis();
                girar_teste = 1;
                beep();
            }
        } else
            tempo_anterior = millis(); //Fim if-else checagem para sair após
2 segundos

        motor_passos.step(10);
        altera_velocidade();
    } //While para controlar a função em loop

} //Função para girar continuo até sair dela

void loop(){
Serial.println("MAIN");
altera_velocidade();
tempo = millis();
//valor_pot = analogRead(pin_pot);

delay(50);
valor_bot1 = digitalRead(pin_bot1);
valor_bot2 = digitalRead(pin_bot2);

if((valor_bot1 == 1 && valor_bot2 == 0) && pre1 == 1){

    motor_passos.step(steps_per_revolution/8);
}

```

```
if((valor_bot2 == 1 && valor_bot1 == 0) && pre2 == 1){

    motor_passos.step(-steps_per_revolution/8);
}

//Exutura de if-else que previnem o clique junto executar algo:
if((valor_bot1 == 1 && valor_bot2 == 0) && pre1 == 0)
    pre1 = 1;
else
    pre1 = 0;

if((valor_bot1 == 0 && valor_bot2 == 1) && pre2 == 0)
    pre2 = 1;
else
    pre2 = 0;

//Exutura de identificação para botões pressionado 2 segundos
if(valor_bot1 == 1 && valor_bot2 == 1){
    if(valor_bot1 == 1 && valor_bot2 == 1 && (tempo - tempo_anterior > 2000)){
        tempo_anterior = millis();
        beep();
        girar_teste = 0;
        girar();
    }
} else
    tempo_anterior = millis();

} //Fim LOOP
```