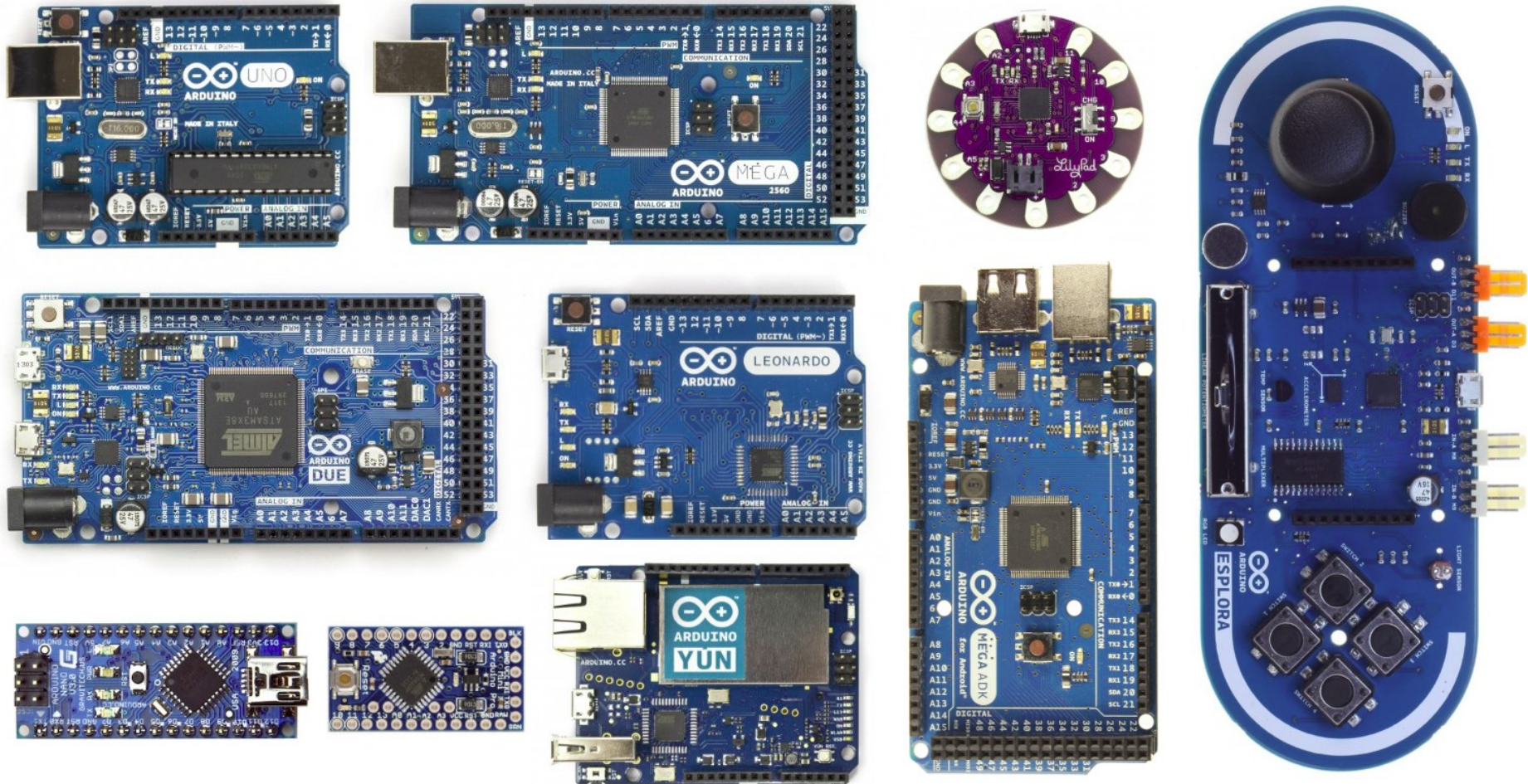




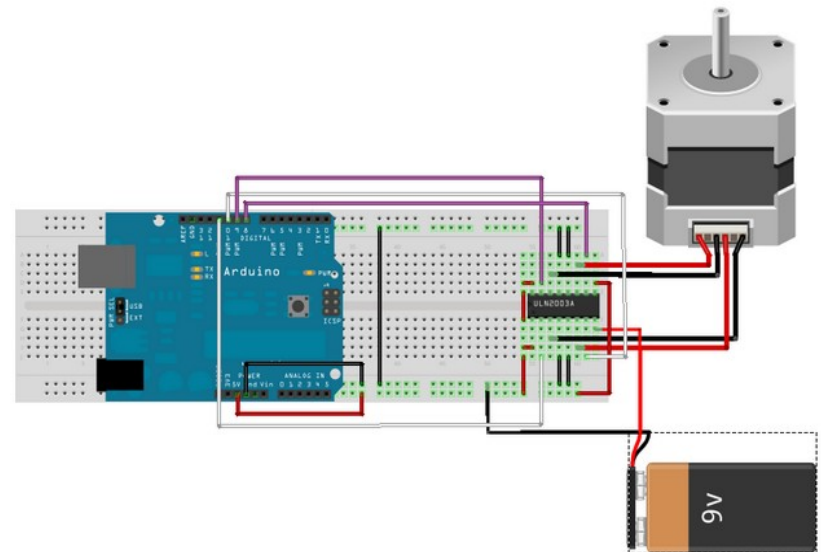
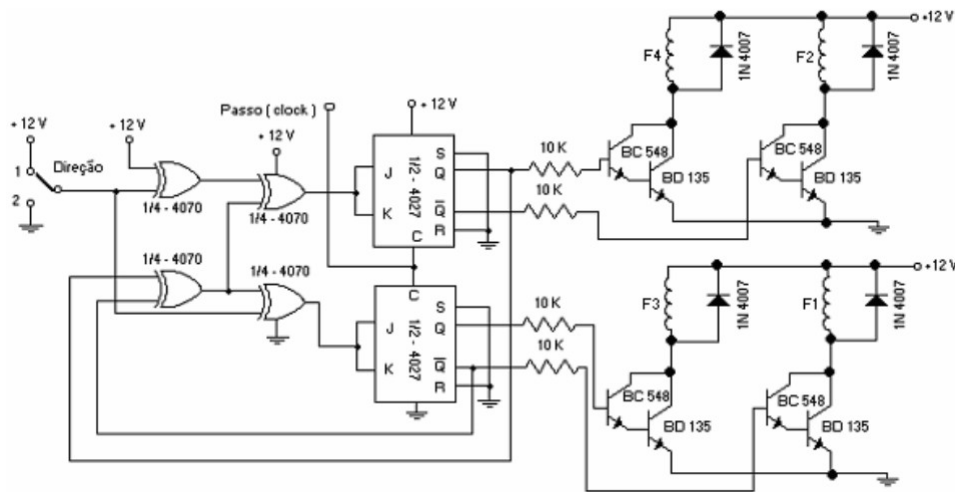
Curso Arduino – Aula 6





Motor de Passo

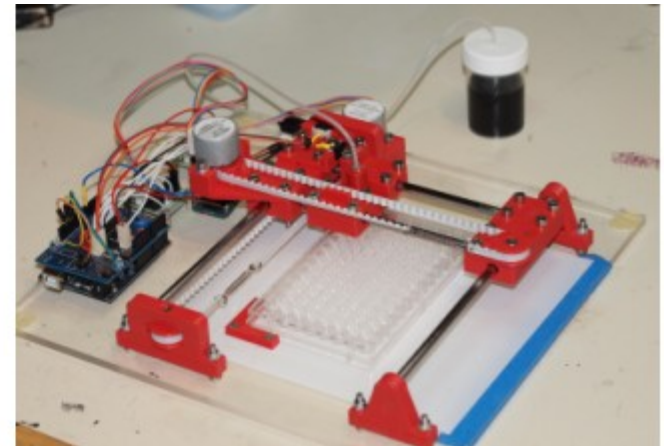
Motores de passos são dispositivos eletromagnéticos que podem ser controlados digitalmente através de um hardware específico ou através de softwares.





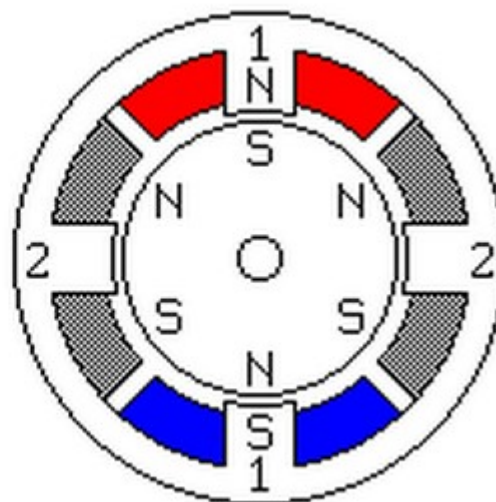
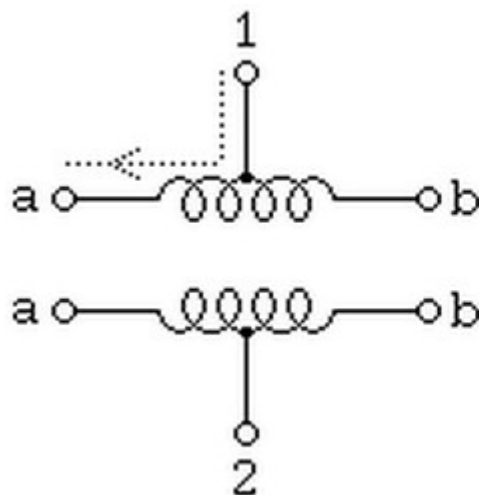
Motor de Passo

Motores de passos são encontrados em aparelhos onde a precisão é um fator muito importante. São usados em larga escala em impressoras, plotters, scanners, drivers de disquetes, discos rígidos e muitos outros aparelhos.



Motor de Passo Unipolar

O motor de passo unipolar apresenta uma derivação central (center tape) entre o enrolamento de duas bobinas.





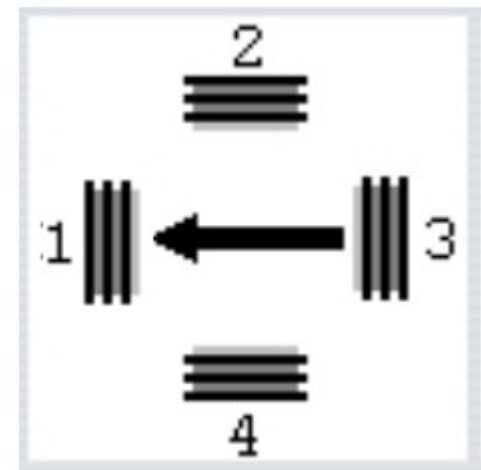
Estados dos Motores de Passo

Desligado:

Não há alimentação no motor.

Não existe consumo de energia, e todas as bobinas estão desligadas.

Na maioria dos circuitos este estado ocorre quando a fonte de alimentação é desligada.

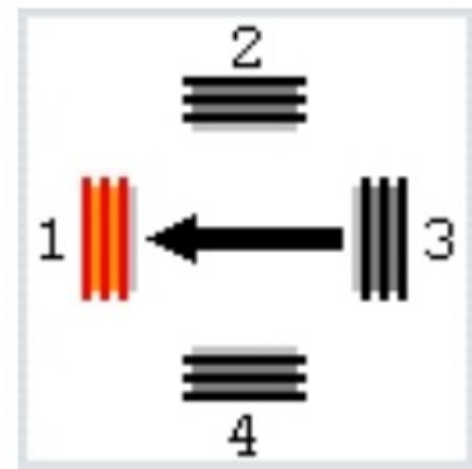


Estados dos Motores de Passo

Parado:

Pelo menos uma das bobinas fica energizada e o motor permanece estático num determinado sentido.

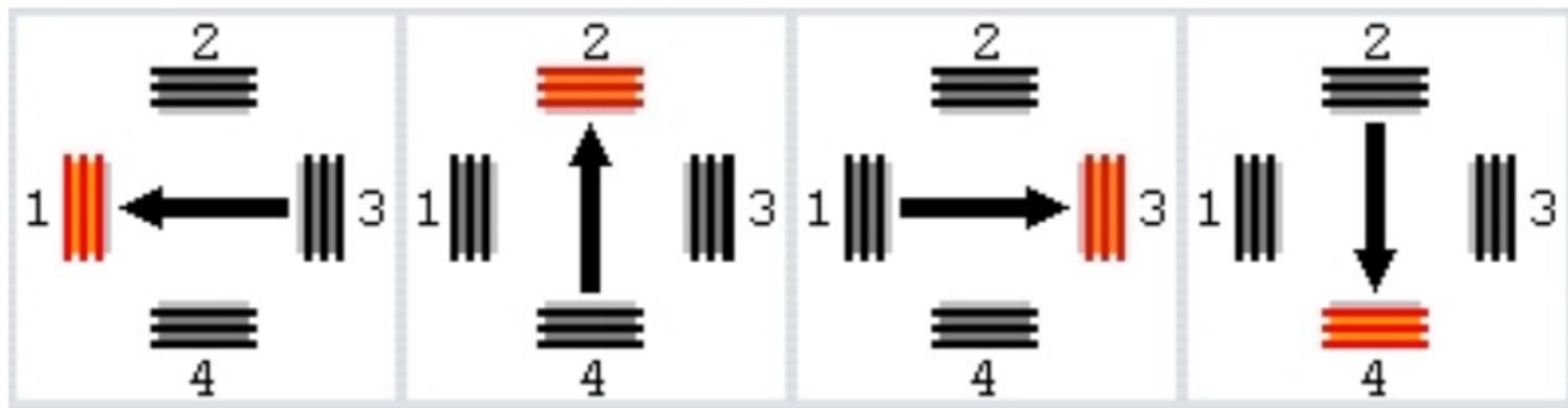
Nesse caso há consumo de energia, mas em compensação o motor mantém-se alinhado numa posição fixa.



Estados dos Motores de Passo

Rodando:

As bobinas são eletrizadas em intervalos de tempos determinados, impulsionando o motor a girar numa direção

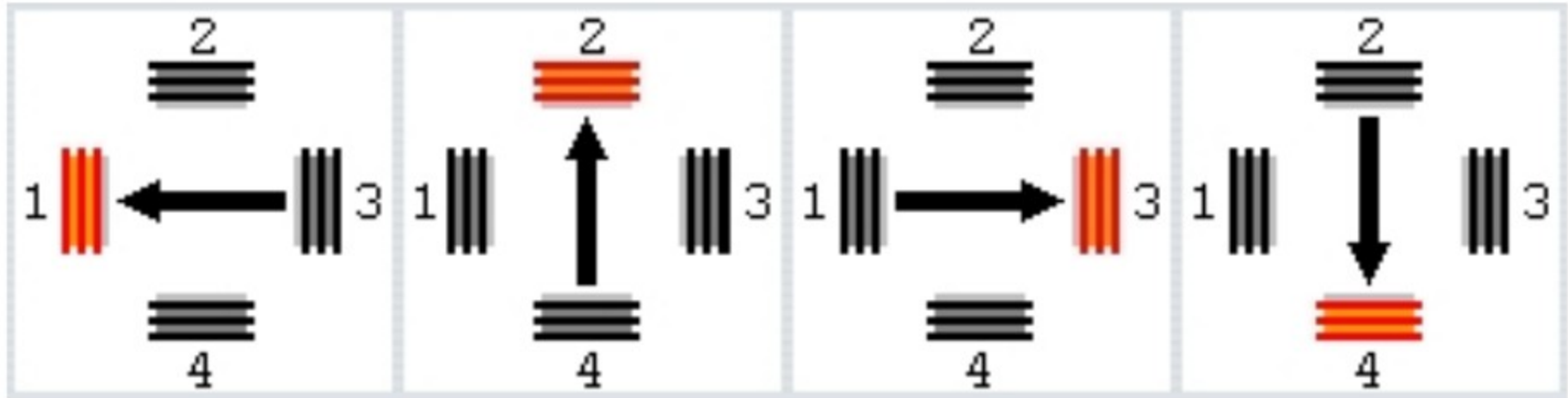




Modos de Operação

Passo completo 1 (Full-step) ou modo simples

- Apenas uma bobina é eletrizada a cada passo;
- Menor torque;
- Pouco consumo de energia;
- Maior velocidade.

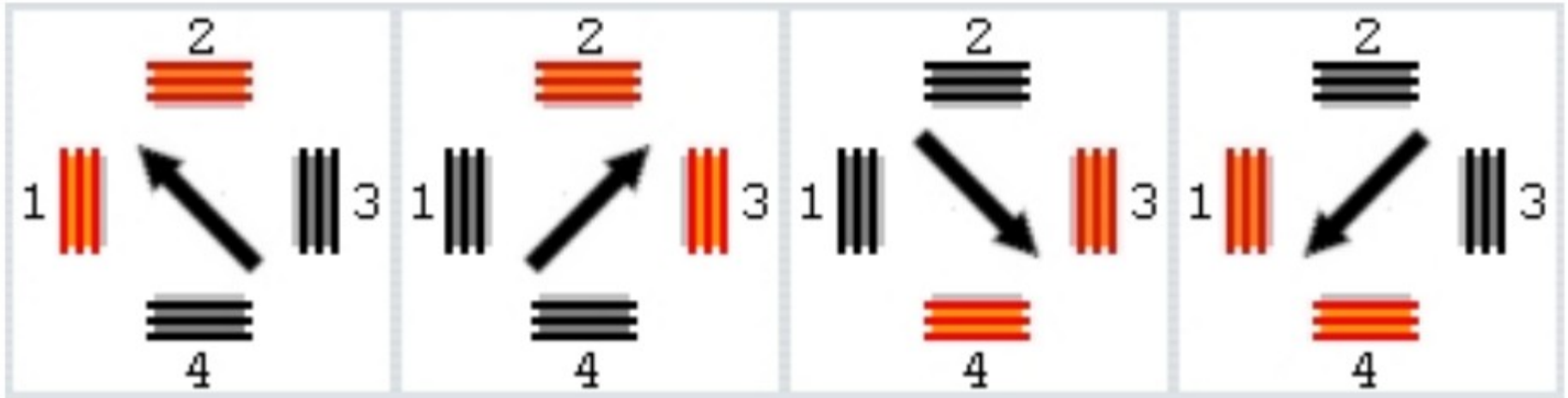




Modos de Operação

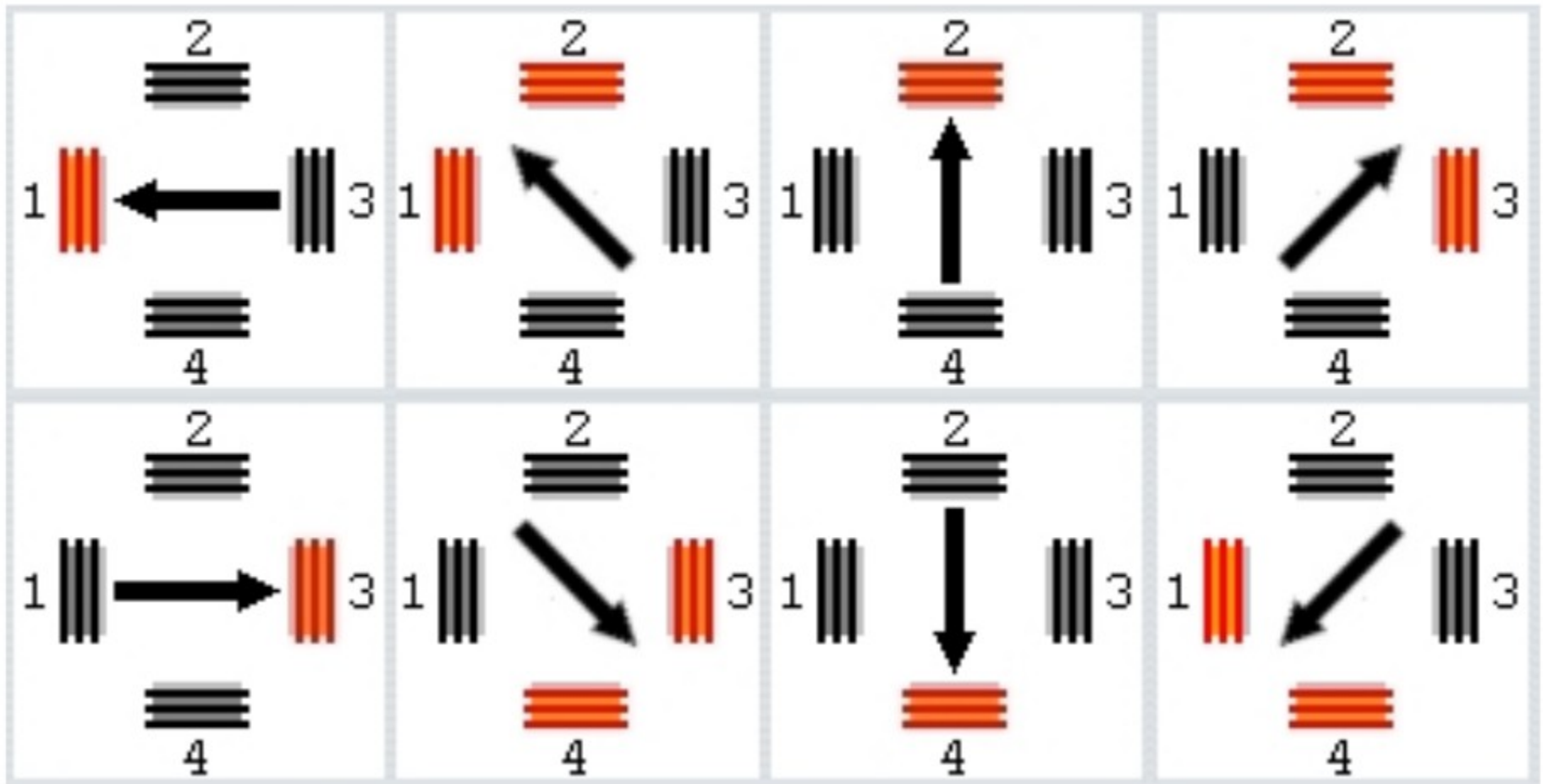
Passo completo 2 (Full-step) com alto torque

- Duas bobinas são eletrizadas a cada passo;
- Maior torque;
- Consome mais energia que o Passo completo 1;
- Maior velocidade.



Modos de Operação

Meio passo (Half-step)





Modos de Operação

Meio passo (Half-step)

- A combinação do passo completo 1 e do passo completo 2 gera um efeito de meio passo;
- Consome mais energia que os passos anteriores;
- É muito mais preciso que os passos anteriores;
- O torque é próximo ao do Passo completo 2;
- A velocidade é menor que as dos passos anteriores.



Modos de Operação

Tabelas com sequências para controle dos Motores de Passo.

Tabela 1 - Passo Completo 1 (Full-step)

Nº do passo	B3	B2	B1	B0	Decimal
1-->	1	0	0	0	8
2-->	0	1	0	0	4
3-->	0	0	1	0	2
4-->	0	0	0	1	1

Tabela 2 - Passo Completo 2 (Full-step)

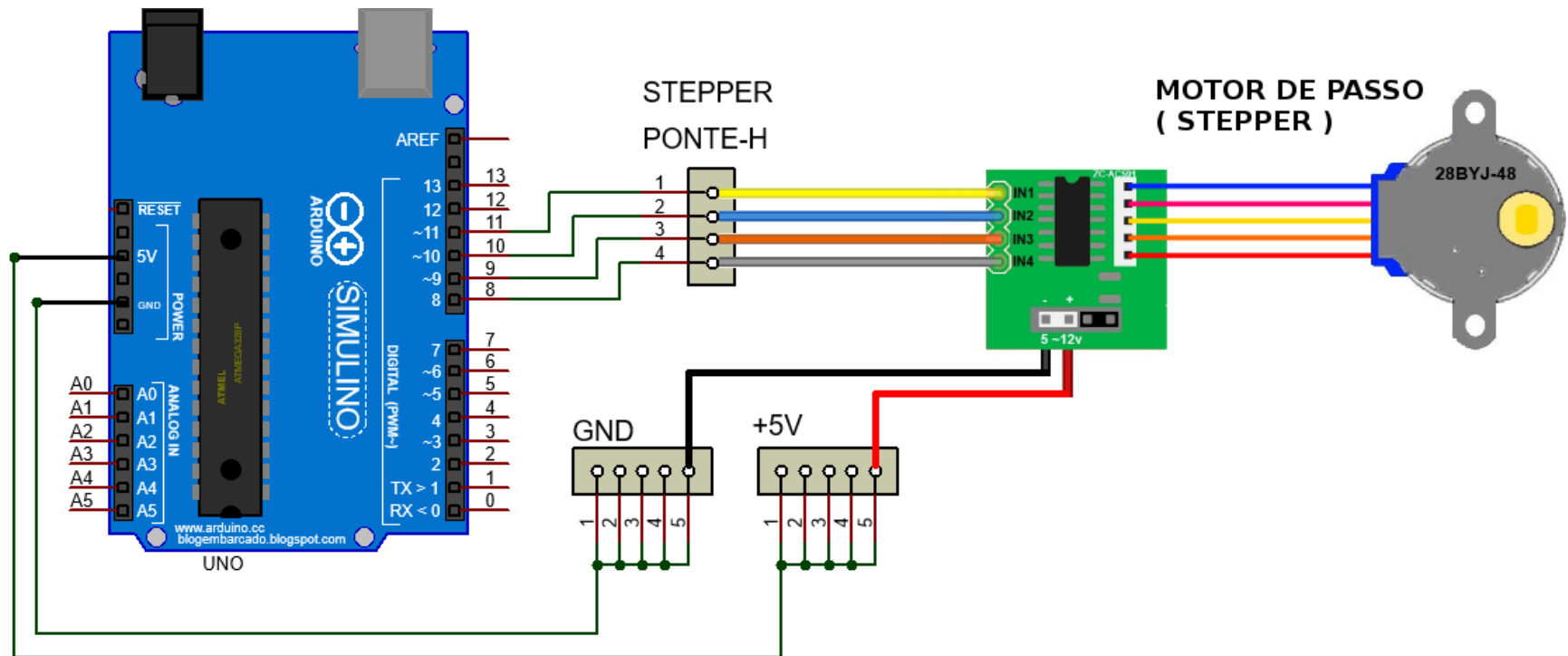
Nº do passo	B3	B2	B1	B0	Decimal
1-->	1	1	0	0	12
2-->	0	1	1	0	6
3-->	0	0	1	1	3
4-->	1	0	0	1	9

Tabela 3 - Meio passo (Half-step)

Nº do passo	B3	B2	B1	B0	Decimal
1-->	1	0	0	0	8
2-->	1	1	0	0	12
3-->	0	1	0	0	4
4-->	0	1	1	0	6
5-->	0	0	1	0	2
6-->	0	0	1	1	3
7-->	0	0	0	1	1
8-->	1	0	0	1	9

Motor de Passo com Drive de Potência

Na placa didática do curso, ligaremos o drive do motor de passo na saída STEPPER e na alimentação de +5V.



Prática 21 - Controle básico de Motor de Passo



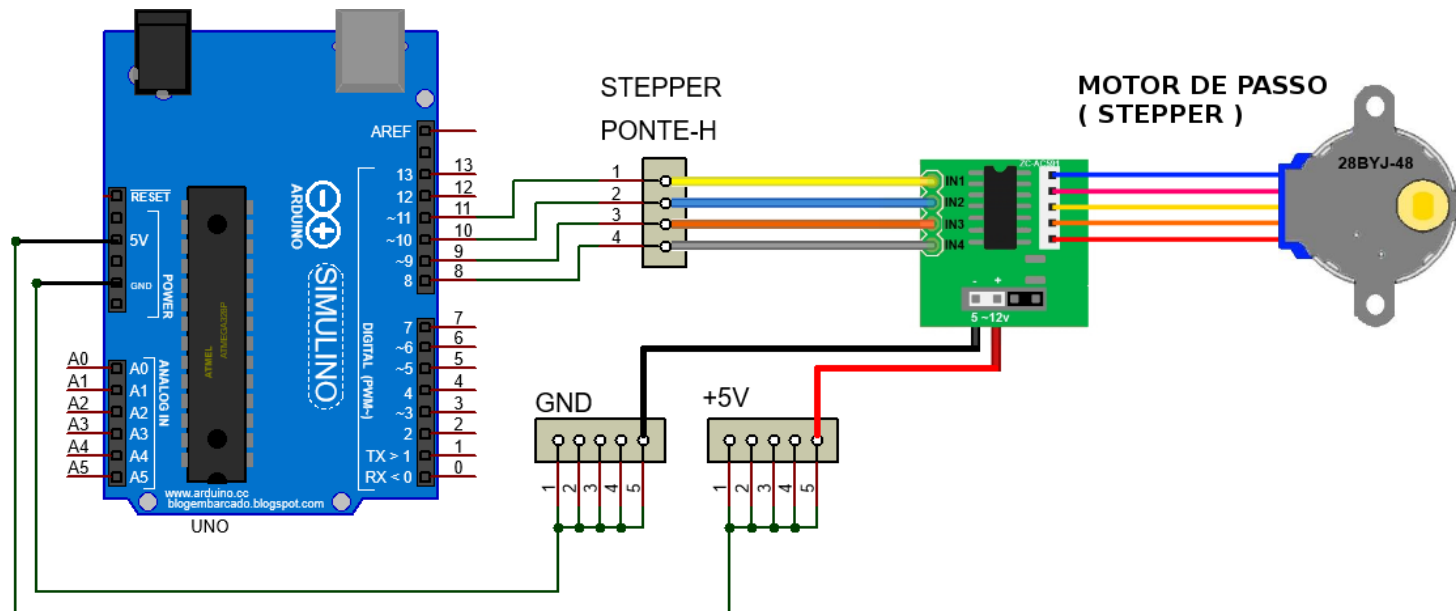
Monte o circuito com Arduíno.

IN1 ao pino 8 do Arduino

IN2 ao pino 9 do Arduino

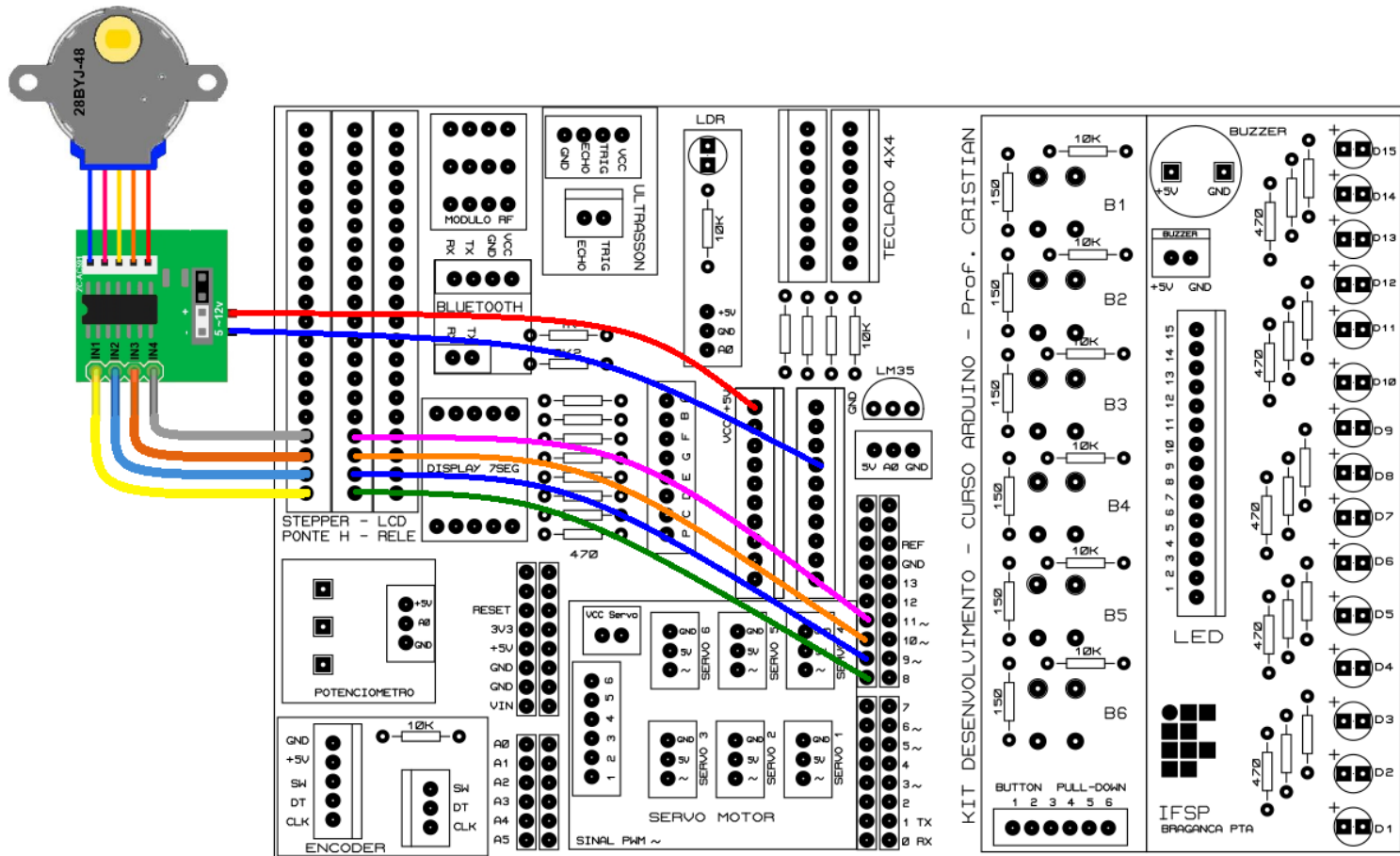
IN3 ao pino 10 do Arduino

IN4 ao pino 11 do Arduino

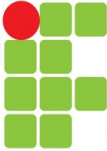


Prática 21 - Controle básico de Motor de Passo

A ligação do circuito deve ser realizada como visto a seguir:



Prática 21 - Controle básico de Motor de Passo



```
#include<Stepper.h> //Inclusão da biblioteca

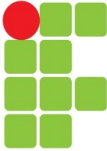
#define GIRO 500    // Definir o numero de passos por volta
                  // Para Motor de Passo Unipolar 28BYJ-48
                  // de 64 passos/volta utilizaremos
                  // 500 passos/volta para suavizar a rotação

Stepper passos(GIRO,8,10,9,11); // Motor de passo ligados pinos 8 a 11;

void setup(){
    passos.setSpeed(60); // Velocidade do motor em pwm:
}

void loop(){
    passos.step(-1024); // meia volta no sentido anti-horário:
    delay(1000);

    passos.step(2048); // uma volta completa no sentido horário:
    delay(1000);
}
```



Prática 21 - Controle básico de Motor de Passo

Para diminuir a velocidade ou o passo do motor, basta alterar nas funções **.setSpeed** e **.step**

```
#include<Stepper.h> //Inclusão da biblioteca

#define GIRO 500 // Definir o numero de passos por volta

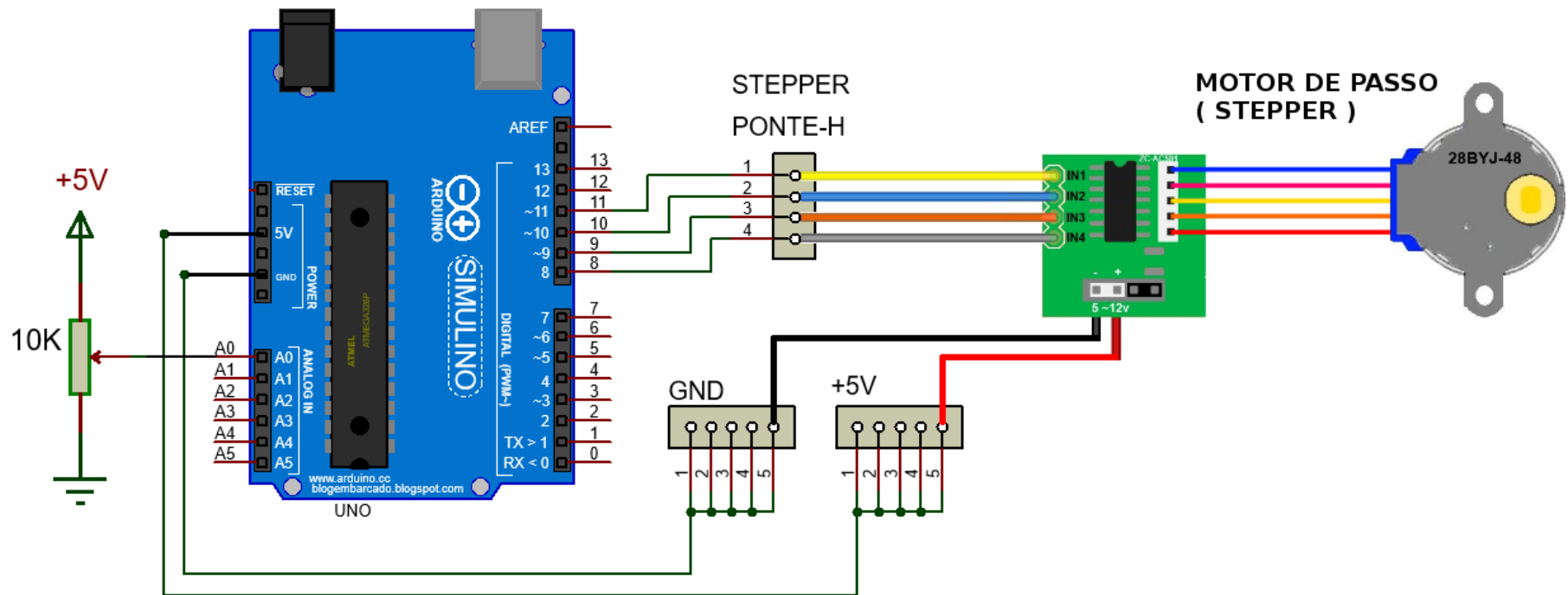
Stepper passos(GIRO,8,10,9,11); // Motor de passo ligados pinos 8 a 11

void setup(){
    passos.setSpeed(10); // Velocidade do motor
}

void loop(){
    passos.step(10); // giro no sentido horário:
    delay(1000);
}
```

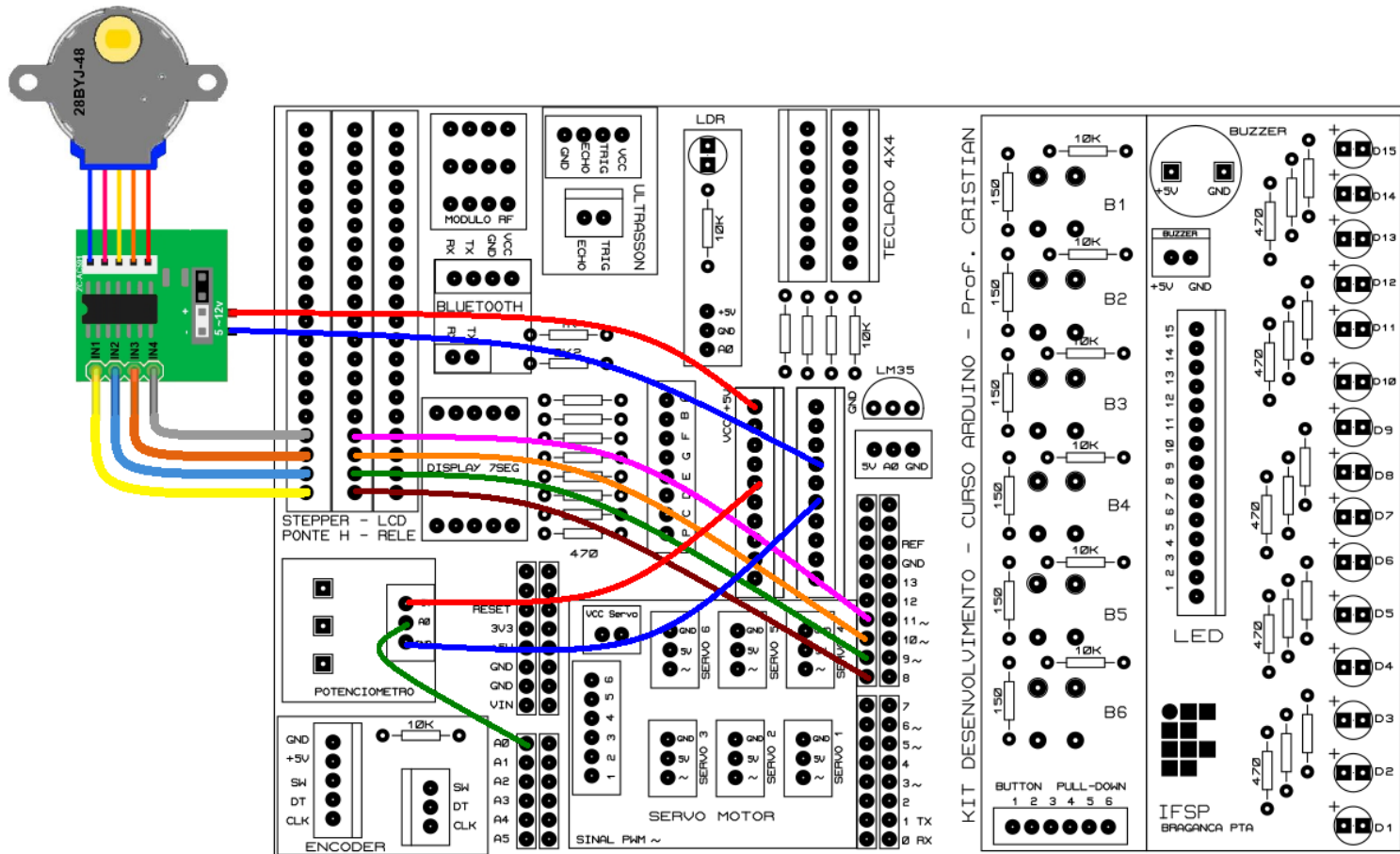
Prática 22 – Controle do Motor de Passo pelo Potenciômetro

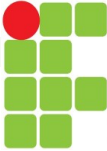
Agora com potenciômetro controlaremos a movimento do motor de passo.



Prática 22 – Controle do Motor de Passo pelo Potenciômetro

A ligação do circuito deve ser realizada como visto a seguir:





Prática 22 – Controle do Motor de Passo pelo Potenciômetro

Agora com potenciômetro controlaremos a movimento do motor de passo.

```
Stepper passos(STEPS, 8, 10, 9, 11); // Motor de passo ligados pinos 8 a 11:

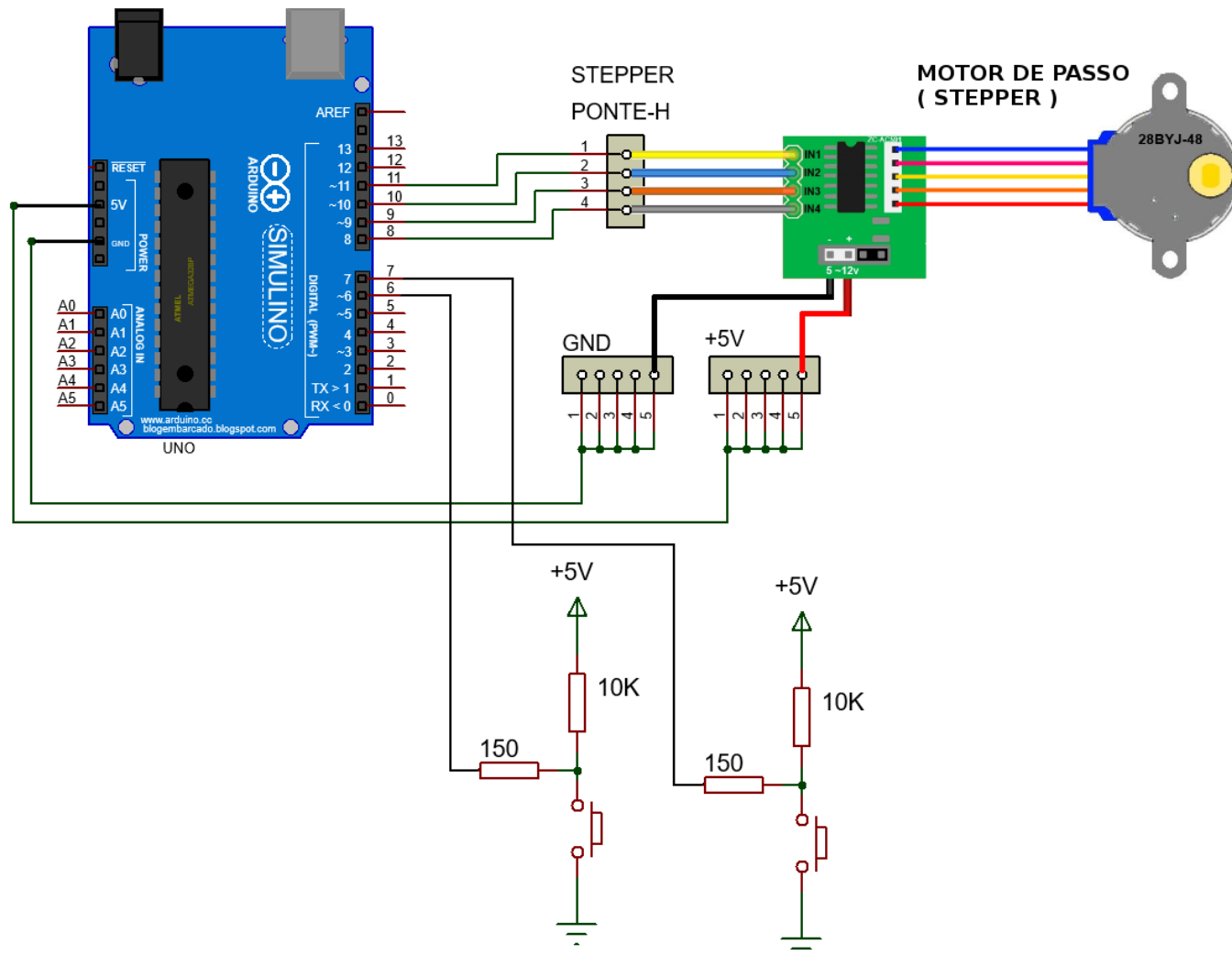
int previsto = 0; // Declarado variavel como inteira

void setup()
{
    passos.setSpeed(60); // Velocidade do motor
}

void loop()
{
    int valor = analogRead(0); // Pega o valor do sensor na entrada analogica A0
    passos.step(valor - previsto); // pega o valor lido e subtrai da posição anterior
                                   // com isso, o motor gira so o valor da diferença

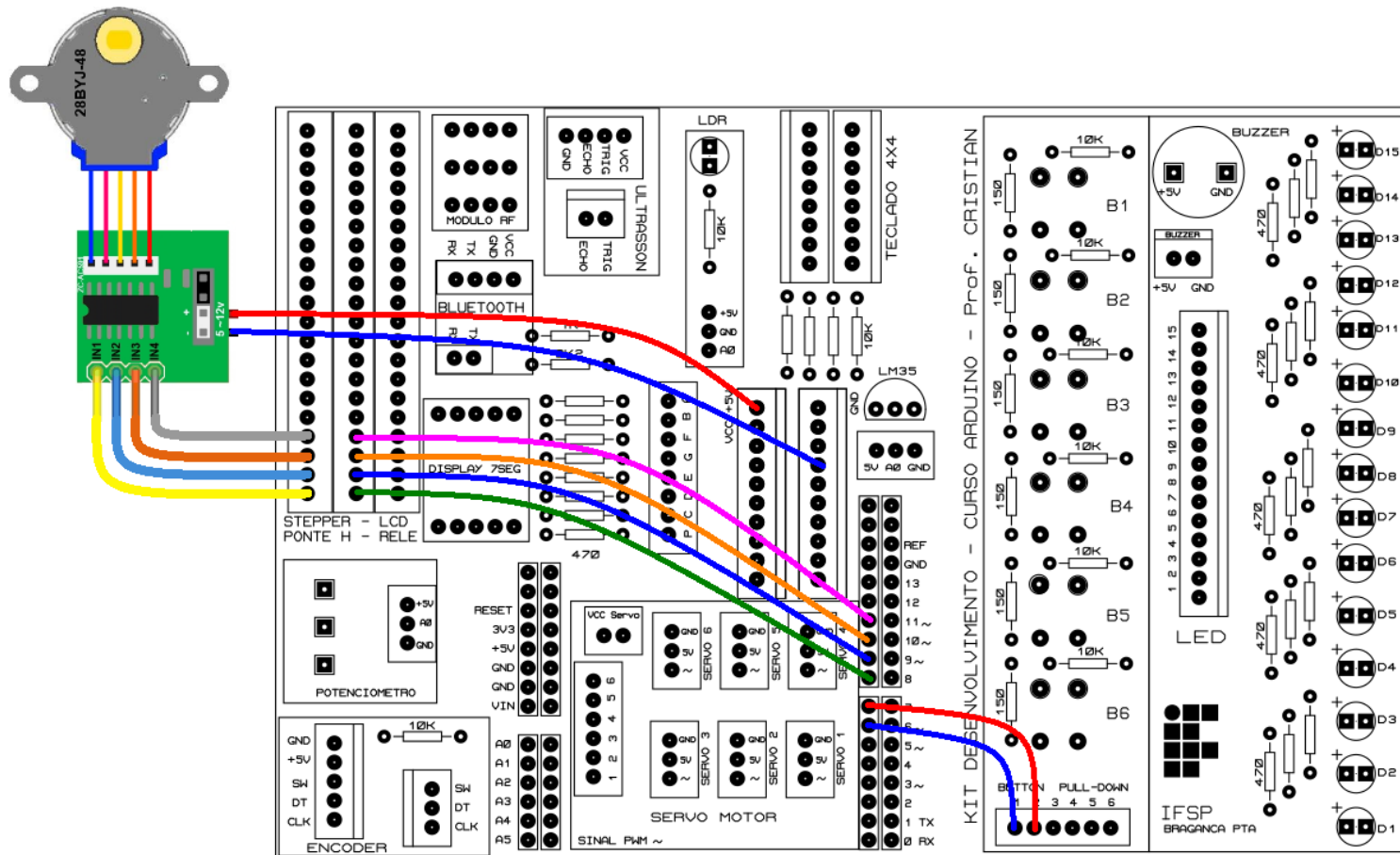
    previsto = valor; // guarda o valor lido na variavel previsto
}
```

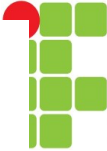

Prática 23 - Controle de Motor de Passo por botões



Prática 23 - Controle de Motor de Passo por botões

A ligação do circuito deve ser realizada como visto a seguir:





```
#include <Stepper.h> //Inclusão da biblioteca

#define STEPS 500 // Definir o numero de passos por volta
int button1 = 7;  // pino 7 atribuido ao botao 1
int button2 = 6;  // pino 4 atribuido ao botao 2
int buttonState1 = 0; // Estado logico do botao 1
int buttonState2 = 0; // Estado logico do botao 2
Stepper passos(STEPS, 8, 10, 9, 11); // Motor de passo ligados pinos 8 a 11:

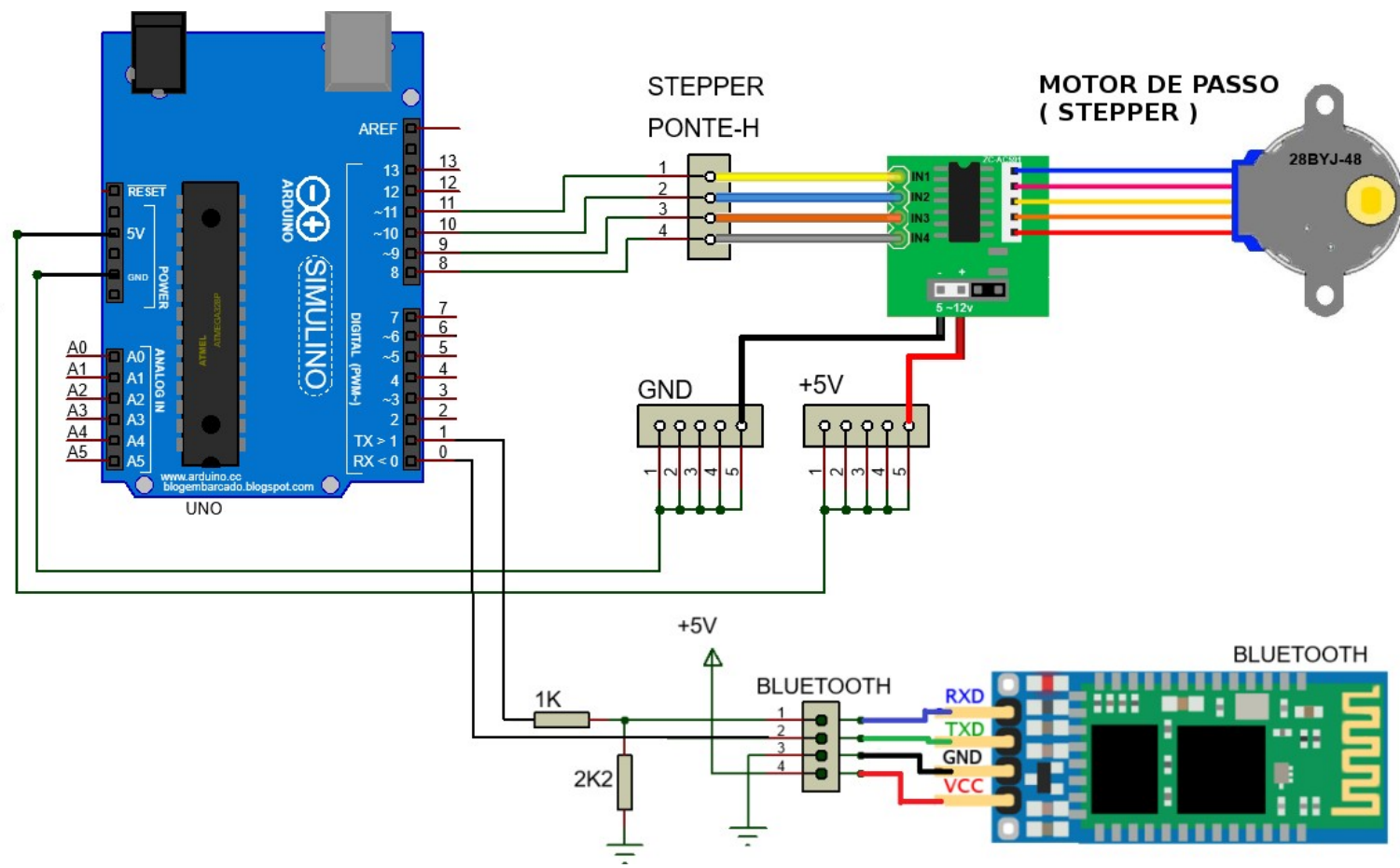
void setup(){
    pinMode(button1, INPUT); //declarado pino 7 como entrada
    pinMode(button2, INPUT); //declarado pino 4 como entrada
    passos.setSpeed(60); // Velocidade do motor
}

void loop() {
    buttonState1=digitalRead(button1); // verifica se o botão foi pressionado
    buttonState2=digitalRead(button2); // e armazena resultado na variavel state

    if(buttonState1 == HIGH){          // verifica se o botão foi pressionado
        passos.step(10);              // e armazena resultado na variavel buttonState
        if(buttonState1 == LOW){
            return;                   // quando o boto for solto a função return
        }                             // força a sair dos if's para o inicio do loop
    }
    if(buttonState2 == HIGH){
        passos.step(-10);
        if(buttonState2 == LOW){
            return;
        }
    }
}
```

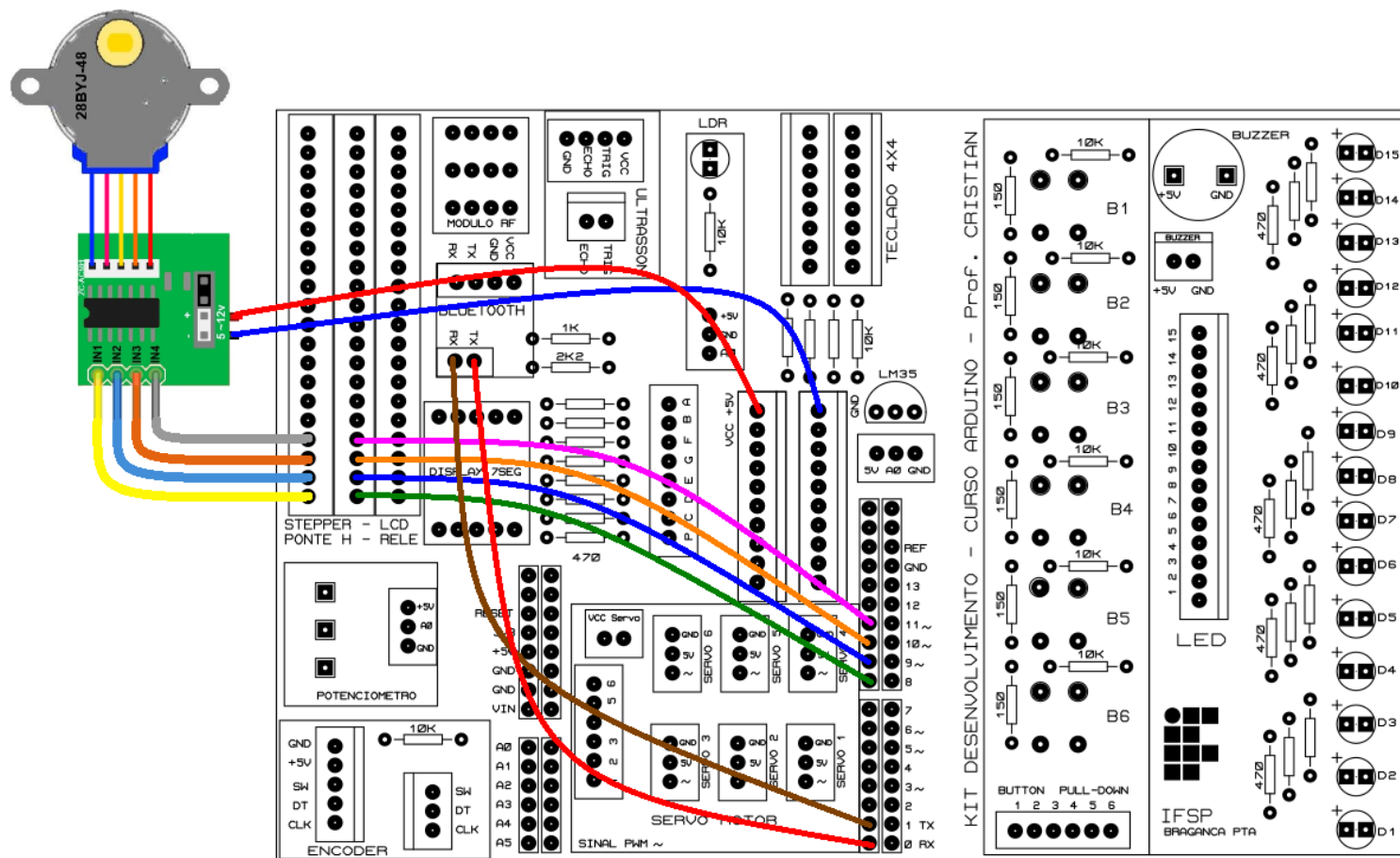
Prática 24 – Controle do Motor de Passo via Bluetooth

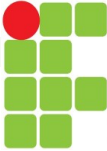
Agora veremos um programa para acionar o motor de passo, via bluetooth, no sentido horário e anti-horário



Prática 24 – Controle do Motor de Passo via Bluetooth

A ligação do circuito deve ser realizada como visto a seguir:





```
#include <Stepper.h>

#define STEPS 500

char comando;

Stepper passos(STEPS, 8, 10, 9, 11);

void setup() {
    Serial.begin(9600);
    passos.setSpeed(60);
}

void loop() {
    while(Serial.available() > 0) {
        comando=Serial.read();
    }
    if(comando=='L'){
        passos.step(-10); //L = Left
    }

    if(comando=='R'){
        passos.step(10); //R = Righth
    }

    if(comando=='S'){ //S = Stop
    }
}
```


Prática 25 – Controle do Motor de Passo via BlueTooth por Ângulo

Desenvolva um programa para acionar o motor de passo, via bluetooth, enviando o ângulo desejado.

