

amazing
robots

Programa

2 SETEMBRO

9H30 – 12H30

O que é um robô?

12H30 – 14H00

Almoço

14H00 – 17H00

Sensores

3 SETEMBRO

9H30 – 12H30

Actuadores

12H30 – 14H00

Almoço

14H00 – 17H00

Desenho e
Impressão 3D

4 + 5 SETEMBRO

TODO O DIA

Apoio ao
desenvolvimento
dos robôs

12H30 – 14H00

Almoço

6 SETEMBRO

9H30 – 12H30

Desafio

12H30 – 14H00

Almoço

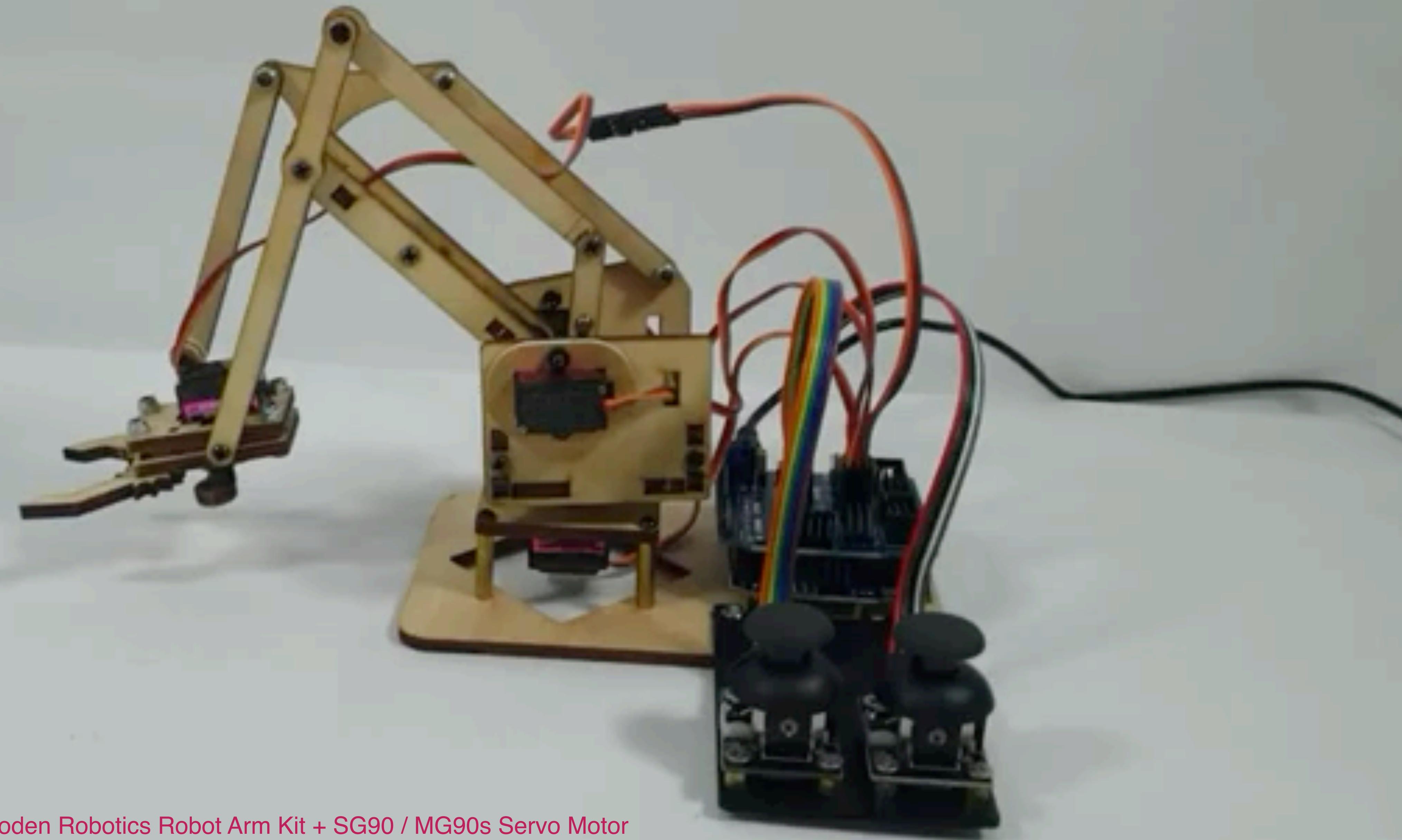
14H00 – 15H00

Entrega de Prémios

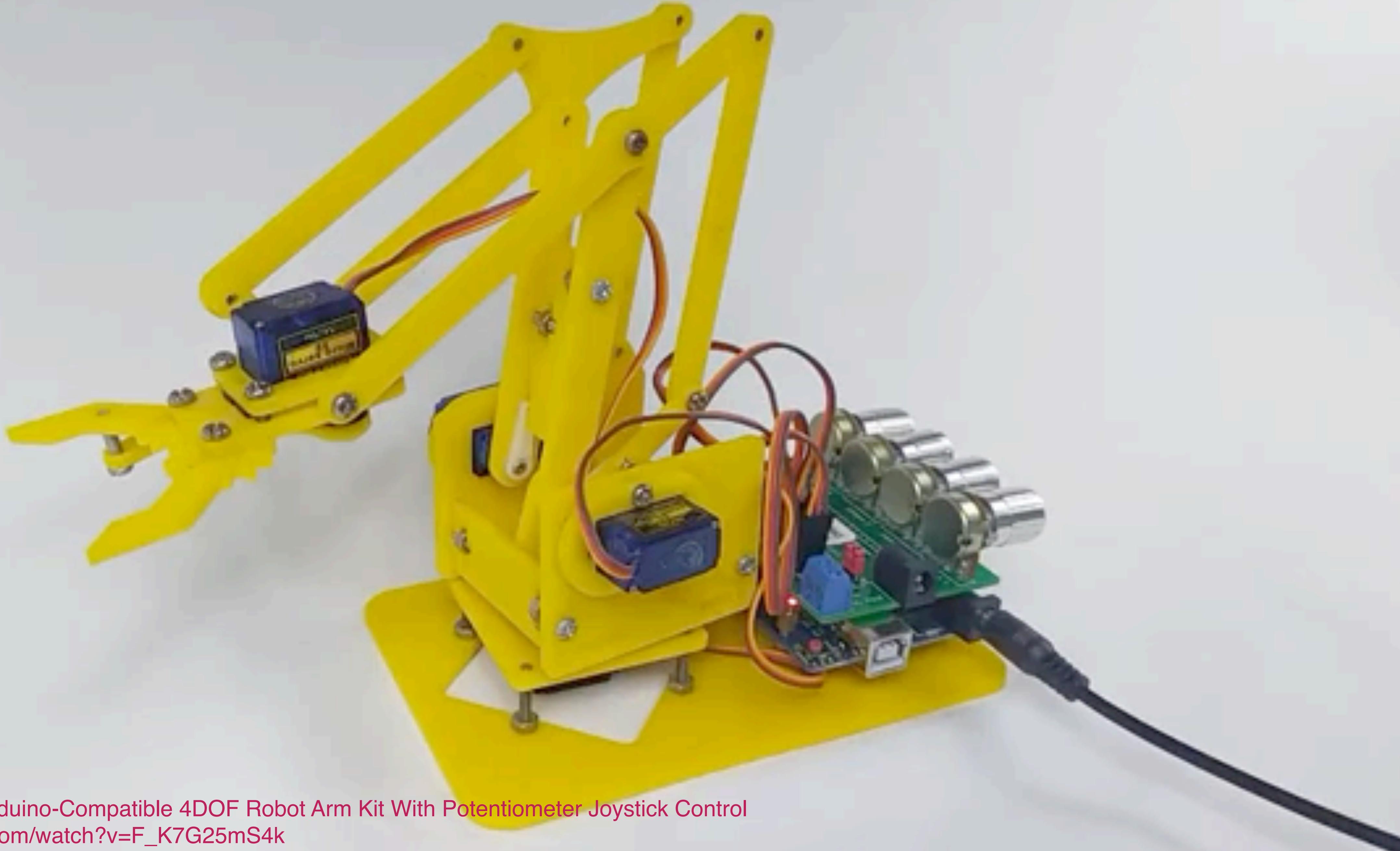
Atuadores

Para que servem os atuadores?





Arduino DIY MeArm 4DOF Wooden Robotics Robot Arm Kit + SG90 / MG90s Servo Motor
https://www.youtube.com/watch?v=N-_1nxhkiKQ



BARNABAS
ROBOTICS

Barnabas Robotics Arduino-Compatible 4DOF Robot Arm Kit With Potentiometer Joystick Control
https://www.youtube.com/watch?v=F_K7G25mS4k

Atuadores Elétricos

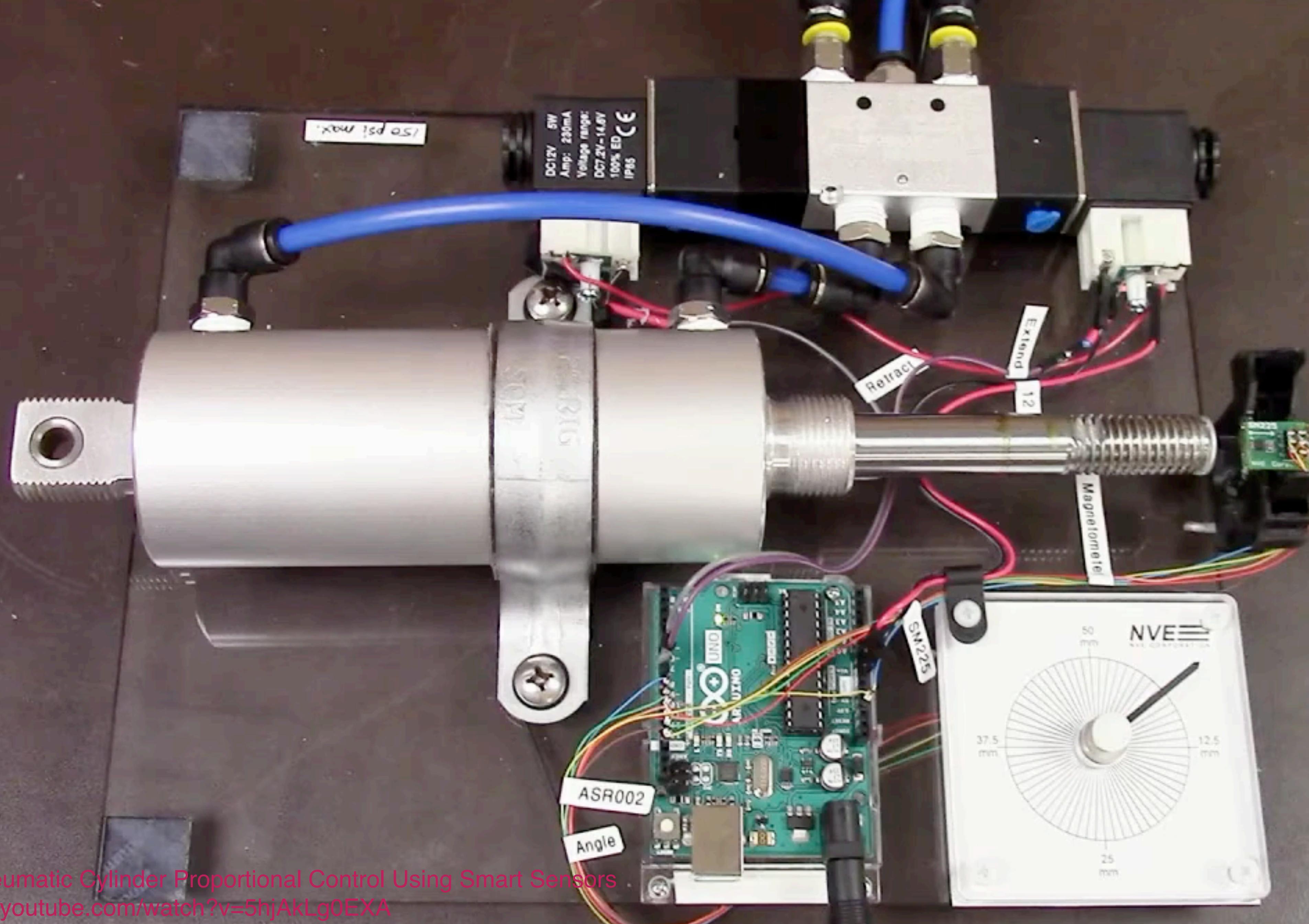
Os atuadores elétricos convertem energia elétrica em movimento mecânico. Eles são amplamente utilizados numa variedade de aplicações, desde equipamentos industriais até pequenos dispositivos domésticos.



Atuadores Pneumáticos

Os atuadores pneumáticos utilizam ar comprimido para gerar movimento mecânico. São conhecidos pela sua capacidade de fornecer movimentos rápidos e repetitivos.





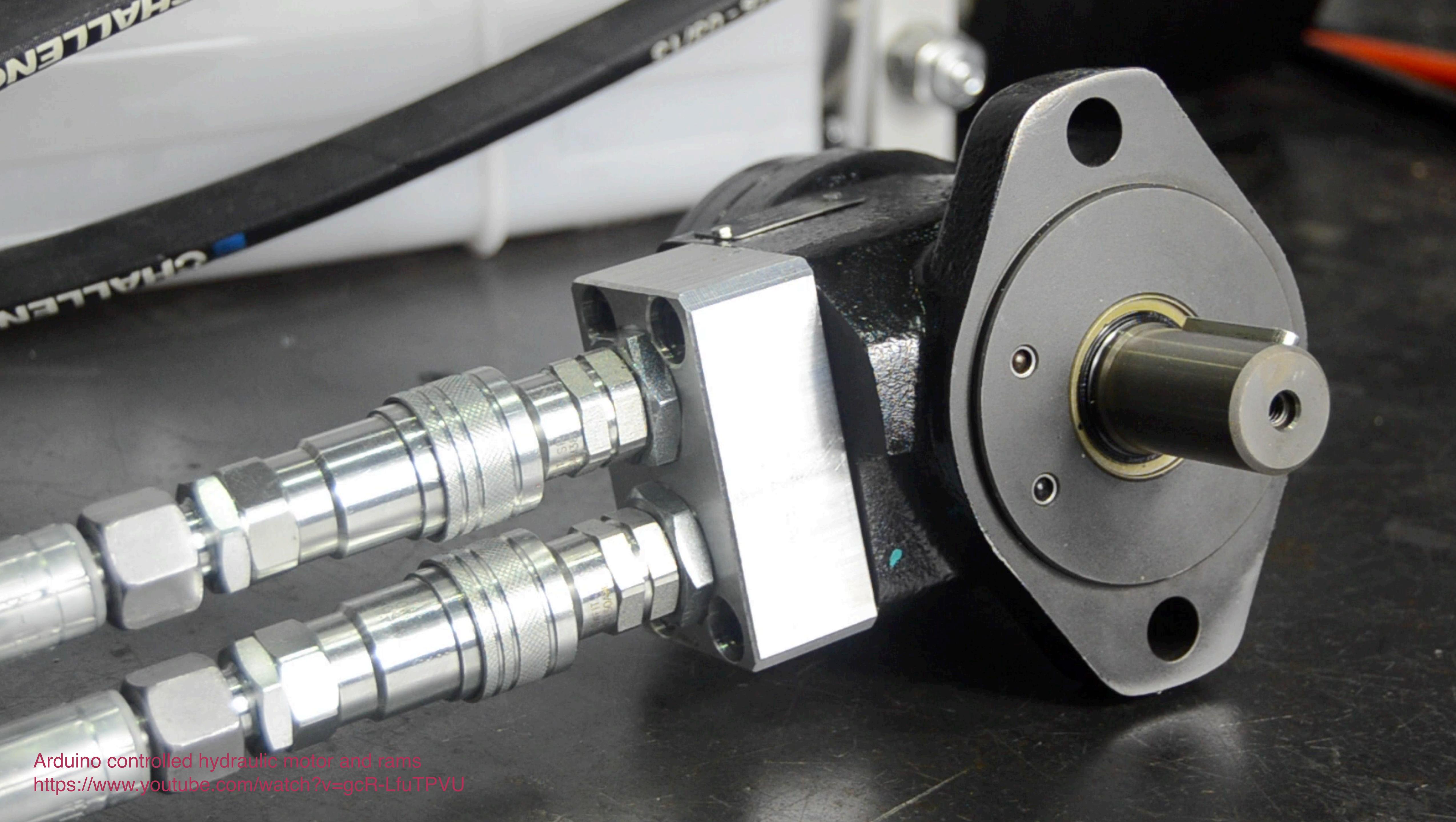
Arduino Pneumatic Cylinder Proportional Control Using Smart Sensors
<https://www.youtube.com/watch?v=5hjAkLg0EXA>

Atuadores Hidráulicos

Os atuadores hidráulicos utilizam fluidos pressurizados para gerar movimento mecânico.

São frequentemente usados em aplicações que requerem alta potência.

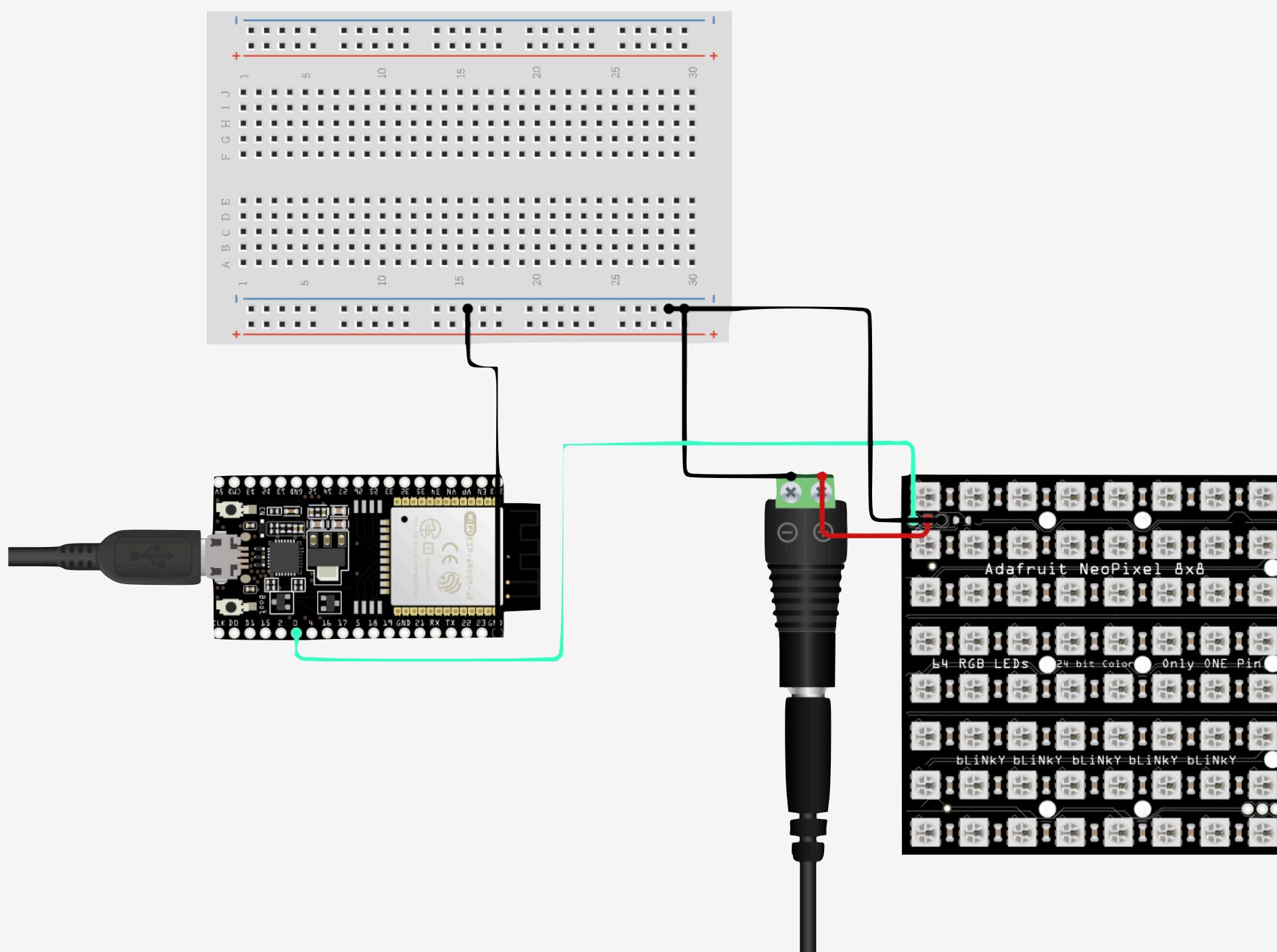




Arduino controlled hydraulic motor and rams
<https://www.youtube.com/watch?v=gcR-LfuTPVU>

Atuadores Térmicos
Atuadores Piezoelétricos
Atuadores Eletromagnéticos
Atuadores Mecânicos
Atuadores Eletroquímicos

RGB led Neopixel

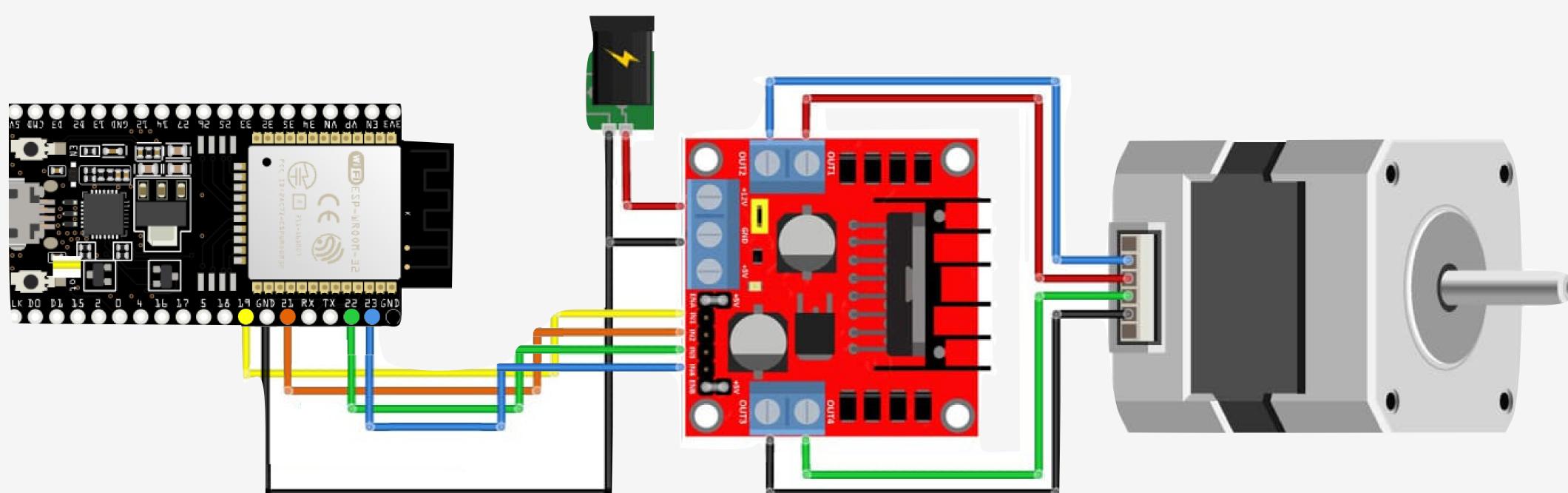


```
teste-neopixel.ino
1 #include <Adafruit_NeoPixel.h>
2
3 int neoPixelPin = 0; //pino onde ligámos os leds
4 int numPixels = 1; //número de leds que temos ligados
5 int BRIGHTNESS = 50;
6
7 Adafruit_NeoPixel strip;
8 void setup() {
9     strip = Adafruit_NeoPixel(numPixels, neoPixelPin, NEO_RGB + NEO_KHZ800);
10    strip.begin();
11    strip.show();
12    strip.setBrightness(BRIGHTNESS);
13 }
14
15 void loop() {
16     // put your main code here, to run repeatedly:
17     strip.setPixelColor(0, strip.Color(255, 0, 0));
18     strip.show();
19     delay(2000);
20
21     strip.setPixelColor(0, strip.Color(0, 255, 0));
22     strip.show();
23     delay(2000);
24
25     strip.setPixelColor(0, strip.Color(0, 0, 255));
26     strip.show();
27     delay(2000);
28 }
```

The screenshot shows the Arduino IDE interface with the file "teste-neopixel.ino" open. The code initializes a NeoPixel strip with 1 pixel on pin 0, sets the brightness to 50, and then enters a loop where it sets each pixel to red, green, and blue respectively, with a 2-second delay between color changes. The status bar at the bottom indicates "Ln 29, Col 1" and "No board selected".

Motores

Motor de passo



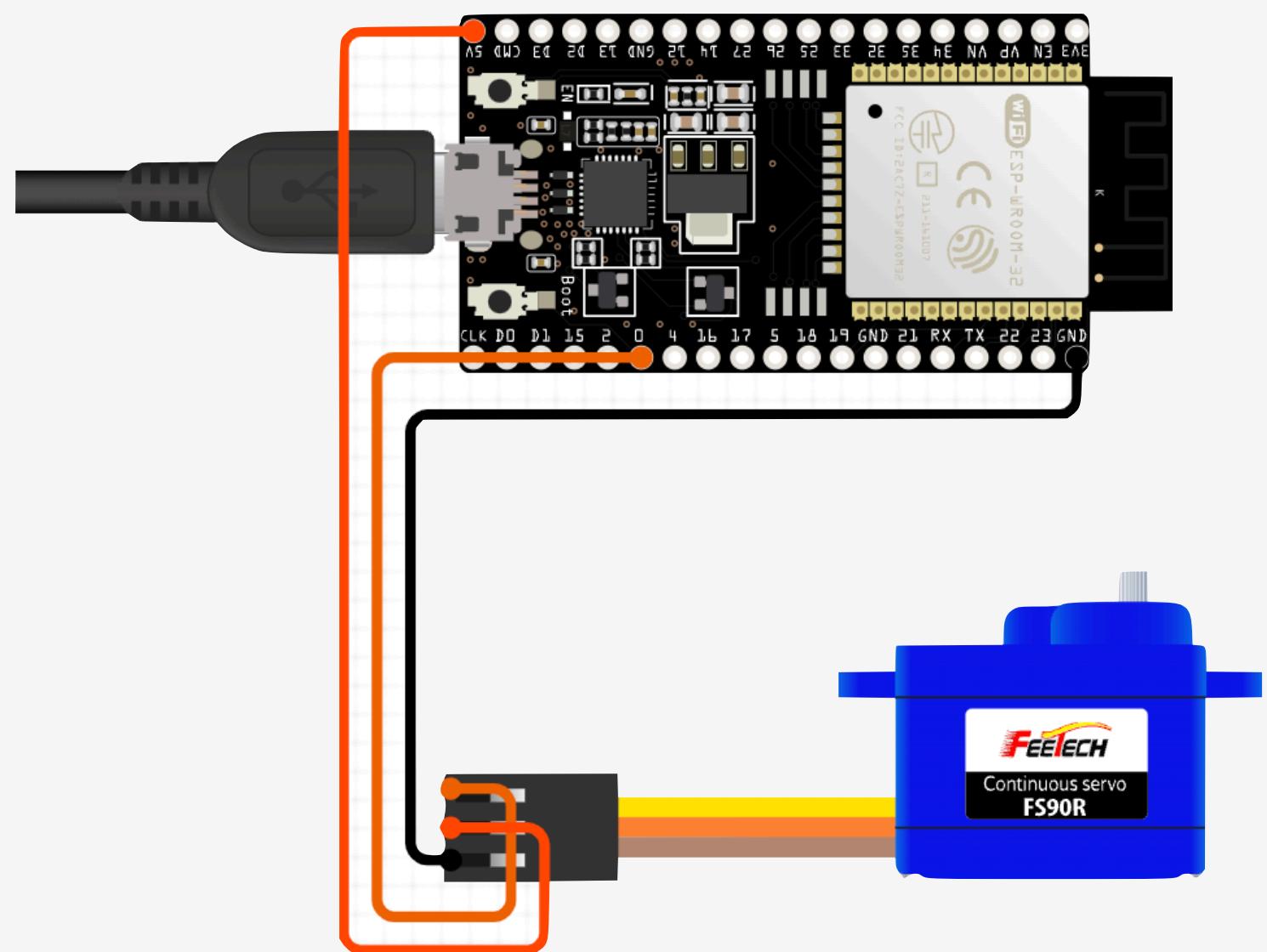
sketch_sep3b | Arduino IDE 2.3.2
DOIT ESP32 DEVKIT V1

```
sketch_sep3b.ino
1 #include <AccelStepper.h> //https://github.com/swissbyte/AccelStepper
2 #define motorInterfaceType 4 /
3
4 AccelStepper stepper = AccelStepper(motorInterfaceType, 25, 27, 26, 14); // IN1, IN3, IN2, IN4
5
6 void setup() {
7     stepper.setMaxSpeed(1000);      // Definir velocidade máxima (passos por segundo)
8     stepper.setAcceleration(500);    // Definir aceleração (passos por segundo^2)
9     Serial.begin(9600);
10
11
12 void loop() {
13     // andar para a frente
14     stepper.moveTo(2038);
15
16     while (stepper.distanceToGo() != 0) {
17         stepper.run(); // Esta função deve ser chamada até o movimento estar completo
18     }
19
20     delay(1000);
21
22     // andar para trás
23     stepper.moveTo(-2038);
24     while (stepper.distanceToGo() != 0) {
25         stepper.run(); // Esta função deve ser chamada até o movimento estar completo
26     }
27     delay(1000);
28 }
```

No Notifications

Ln 9, Col 22 DOIT ESP32 DEVKIT V1 on /dev/cu.usbserial-10 [not connected]

Motores Servo

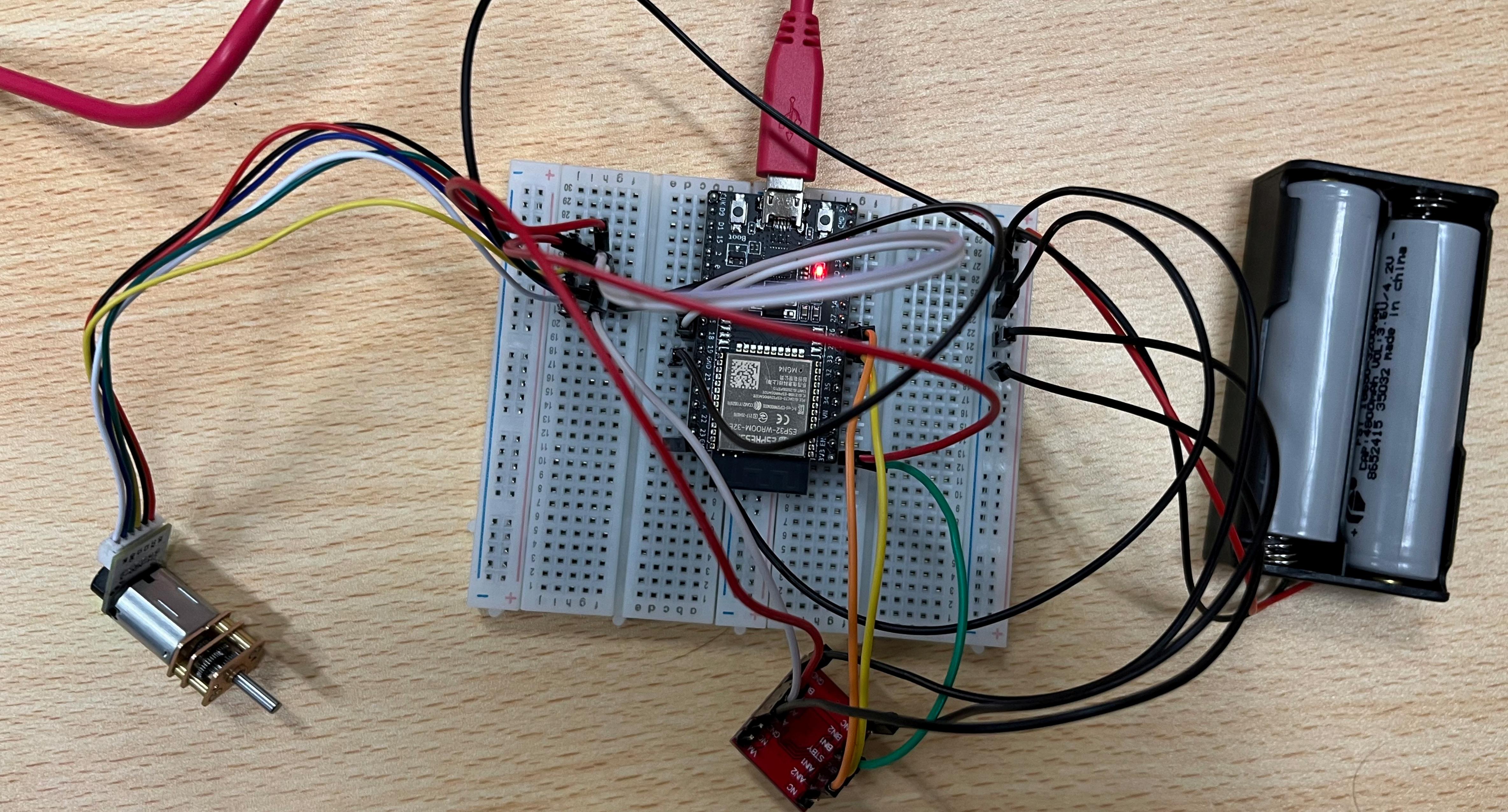


Servo_motor | Arduino IDE 2.0.5-nightly-20230316
DOIT ESP32 DEVKIT V1

```
1 // Incluir a biblioteca ESP32 Arduino Servo Library:  
2 //https://github.com/madhephaestus/ESP32Servo/tree/master  
3 #include <ESP32Servo.h>  
4  
5 Servo myservo; //criar um objeto do tipo servo  
6 int servoPin = 18; // Pin usado  
7  
8 void setup(){  
9     myservo.attach(servoPin); //associar o pin utilizado  
10 }  
11  
12 void loop() {  
13     myservo.write(180); // rodar para uma direção  
14     delay(1000); // esperar 1 segundo  
15  
16     myservo.write(0); // rodar na direção oposta  
17     delay(1000); // esperar 1 segundo  
18 }
```

Output Serial Monitor

Building sketch Ln 1, Col 1 DOIT ESP32 DEVKIT V1 [not connected] 4

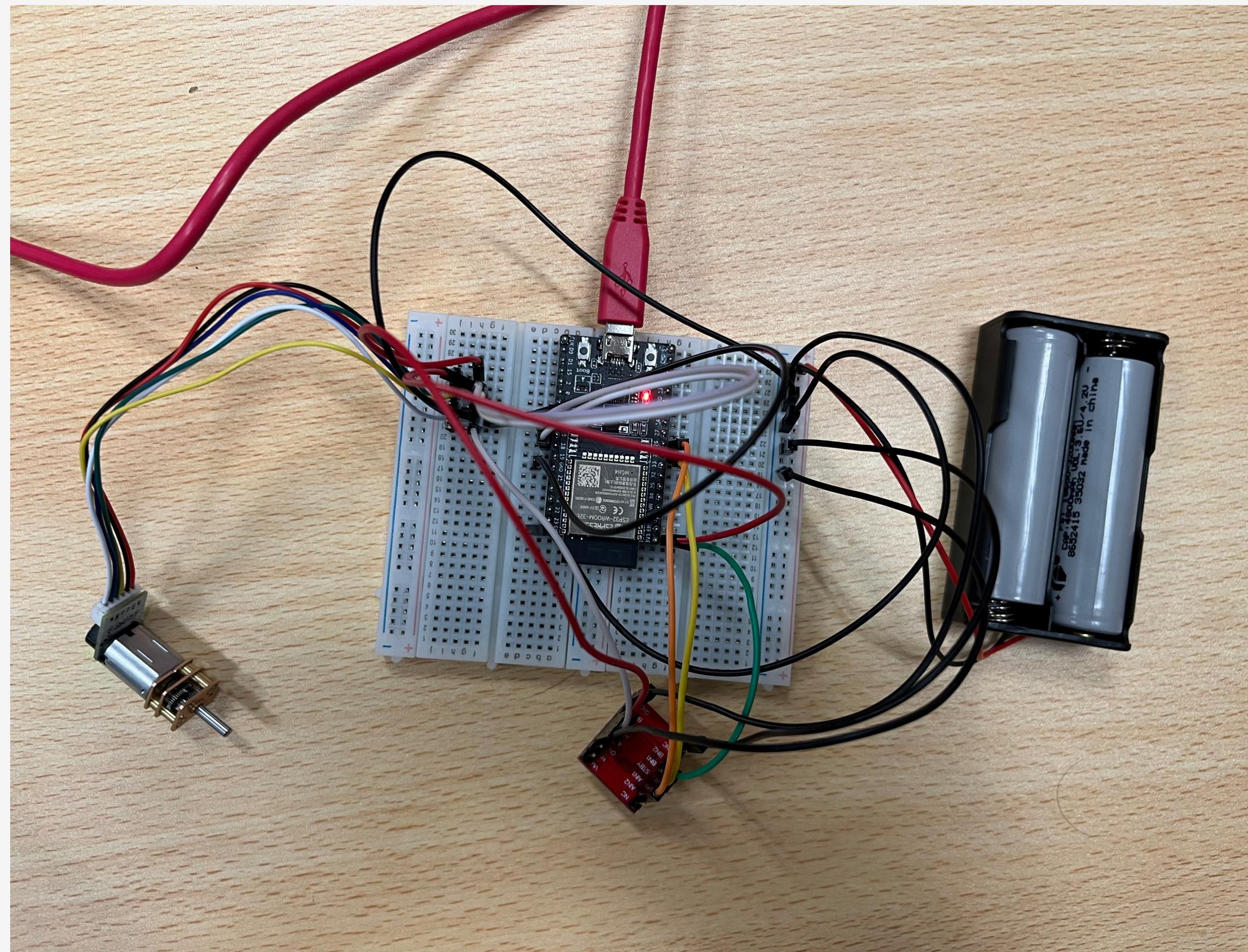


Motores N20

	Motor driver		
/	NC	VM	BAT +
ESP 26	AIN2	NC	/
ESP 25	AIN1	GND	BAT GND
3.3V	STBY	AO1	M1 (Branco Motor)
/	BIN1	AO2	M2 (Vermelho)
/	BIN2	BO2	/
/	NC	BO1	/
BAT GND	GND	GND	GND

Motor	
MD AO1	M1 (Branco)
BAT GND	GND (Azul)
ESP 18	ENC1 (Verde)
ESP 5	ENC2 (Amarelo)
3.3V	VCC (Preto)
MD AO2	M2 (Vermelho)

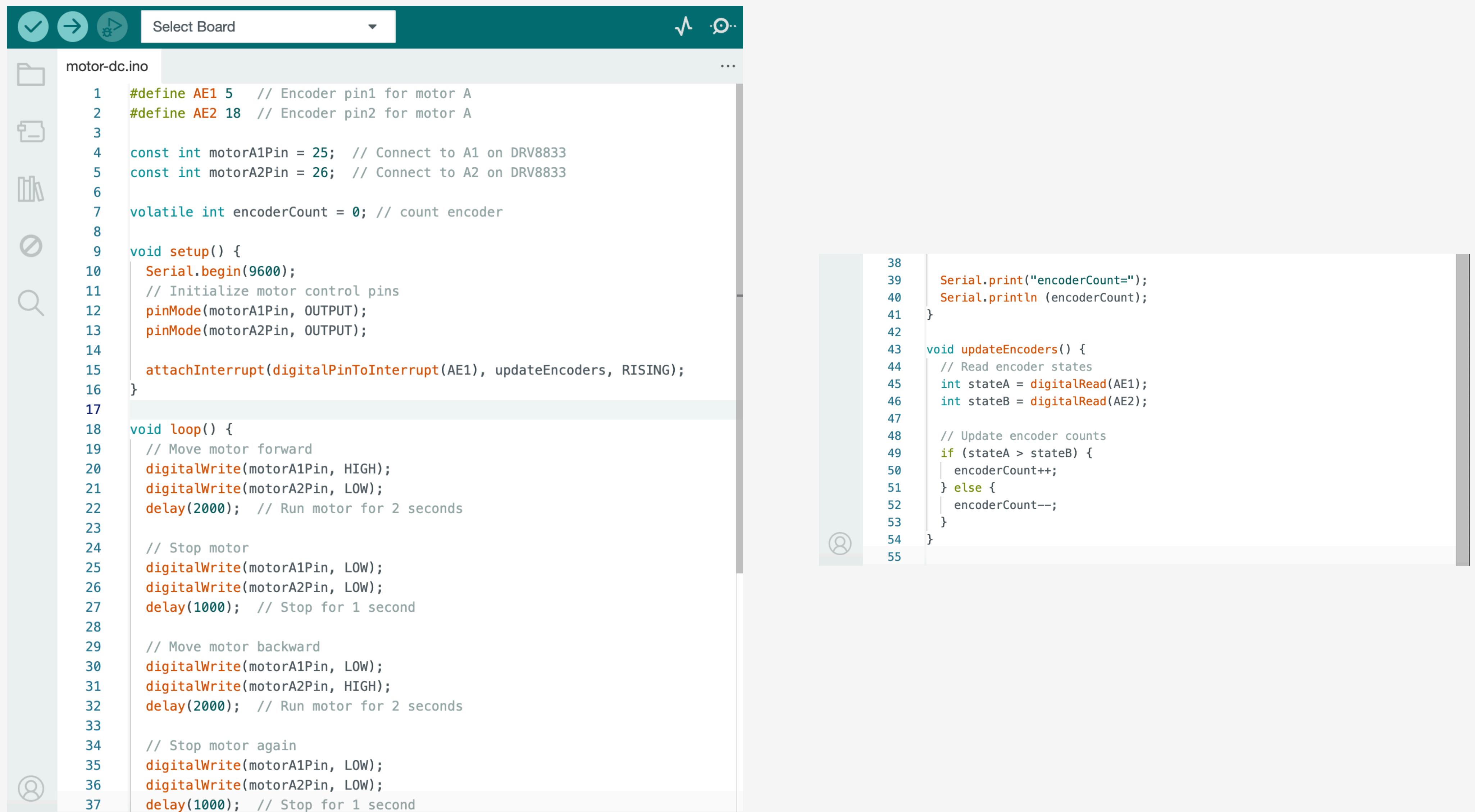
Motores N20



```
motor-dc.ino
1 #define AE1 5 // Encoder pin1 for motor A
2 #define AE2 18 // Encoder pin2 for motor A
3
4 const int motorA1Pin = 25; // Connect to A1 on DRV8833
5 const int motorA2Pin = 26; // Connect to A2 on DRV8833
6
7 volatile int encoderCount = 0; // count encoder
8
9 void setup() {
10   Serial.begin(9600);
11   // Initialize motor control pins
12   pinMode(motorA1Pin, OUTPUT);
13   pinMode(motorA2Pin, OUTPUT);
14
15   attachInterrupt(digitalPinToInterrupt(AE1), updateEncoders, RISING);
16 }
17
18 void loop() {
19   // Move motor forward
20   digitalWrite(motorA1Pin, HIGH);
21   digitalWrite(motorA2Pin, LOW);
22   delay(2000); // Run motor for 2 seconds
23
24   // Stop motor
25   digitalWrite(motorA1Pin, LOW);
26   digitalWrite(motorA2Pin, LOW);
27   delay(1000); // Stop for 1 second
28
29   // Move motor backward
30   digitalWrite(motorA1Pin, LOW);
31   digitalWrite(motorA2Pin, HIGH);
32   delay(2000); // Run motor for 2 seconds
33
34   // Stop motor again
35   digitalWrite(motorA1Pin, LOW);
36   digitalWrite(motorA2Pin, LOW);
37   delay(1000); // Stop for 1 second
```

Motores

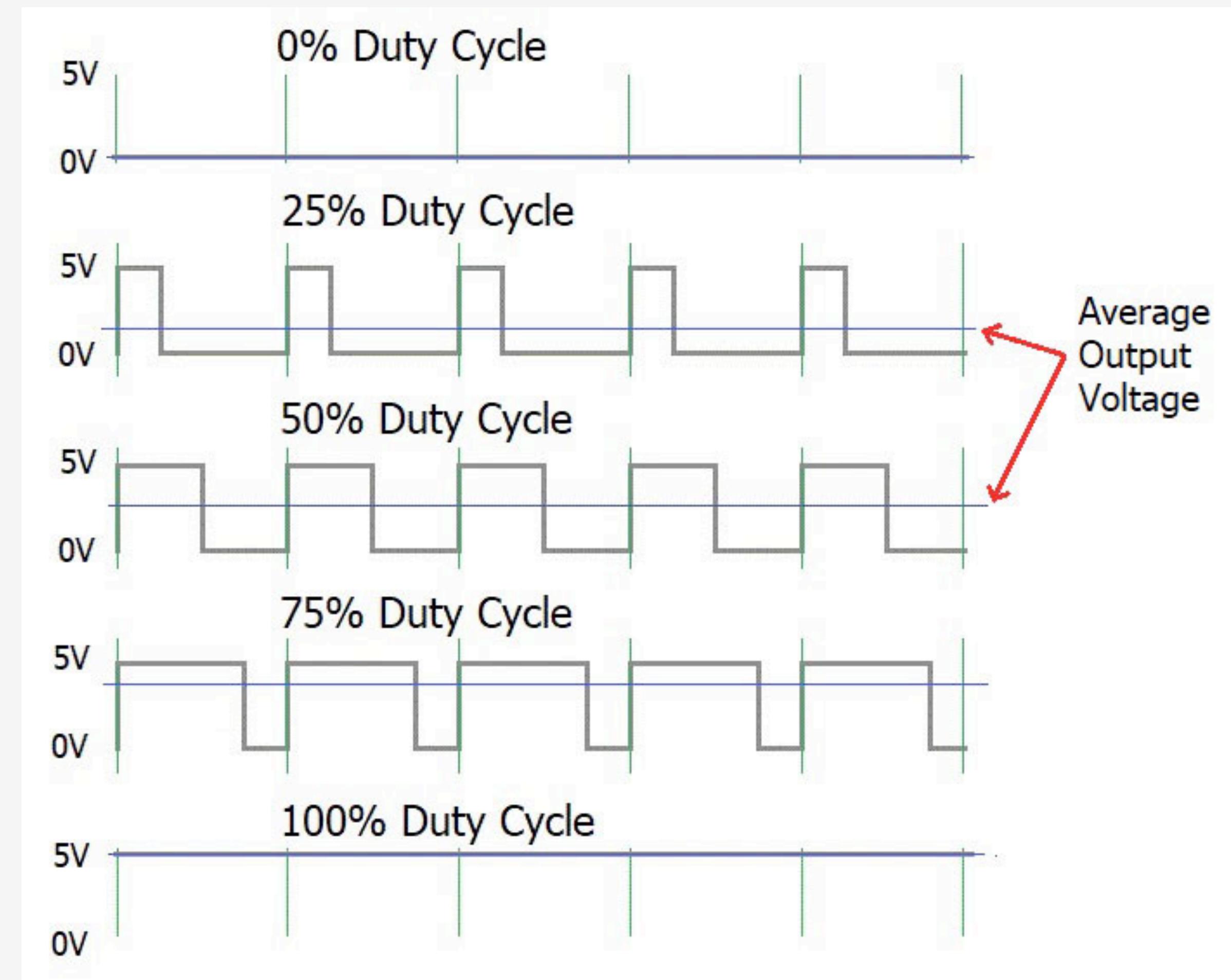
N20



```
motor-dc.ino

1 #define AE1 5 // Encoder pin1 for motor A
2 #define AE2 18 // Encoder pin2 for motor A
3
4 const int motorA1Pin = 25; // Connect to A1 on DRV8833
5 const int motorA2Pin = 26; // Connect to A2 on DRV8833
6
7 volatile int encoderCount = 0; // count encoder
8
9 void setup() {
10     Serial.begin(9600);
11     // Initialize motor control pins
12     pinMode(motorA1Pin, OUTPUT);
13     pinMode(motorA2Pin, OUTPUT);
14
15     attachInterrupt(digitalPinToInterrupt(AE1), updateEncoders, RISING);
16 }
17
18 void loop() {
19     // Move motor forward
20     digitalWrite(motorA1Pin, HIGH);
21     digitalWrite(motorA2Pin, LOW);
22     delay(2000); // Run motor for 2 seconds
23
24     // Stop motor
25     digitalWrite(motorA1Pin, LOW);
26     digitalWrite(motorA2Pin, LOW);
27     delay(1000); // Stop for 1 second
28
29     // Move motor backward
30     digitalWrite(motorA1Pin, LOW);
31     digitalWrite(motorA2Pin, HIGH);
32     delay(2000); // Run motor for 2 seconds
33
34     // Stop motor again
35     digitalWrite(motorA1Pin, LOW);
36     digitalWrite(motorA2Pin, LOW);
37     delay(1000); // Stop for 1 second
38
39     Serial.print("encoderCount=");
40     Serial.println(encoderCount);
41 }
42
43 void updateEncoders() {
44     // Read encoder states
45     int stateA = digitalRead(AE1);
46     int stateB = digitalRead(AE2);
47
48     // Update encoder counts
49     if (stateA > stateB) {
50         encoderCount++;
51     } else {
52         encoderCount--;
53     }
54 }
55
```

Motores N20



Encoders

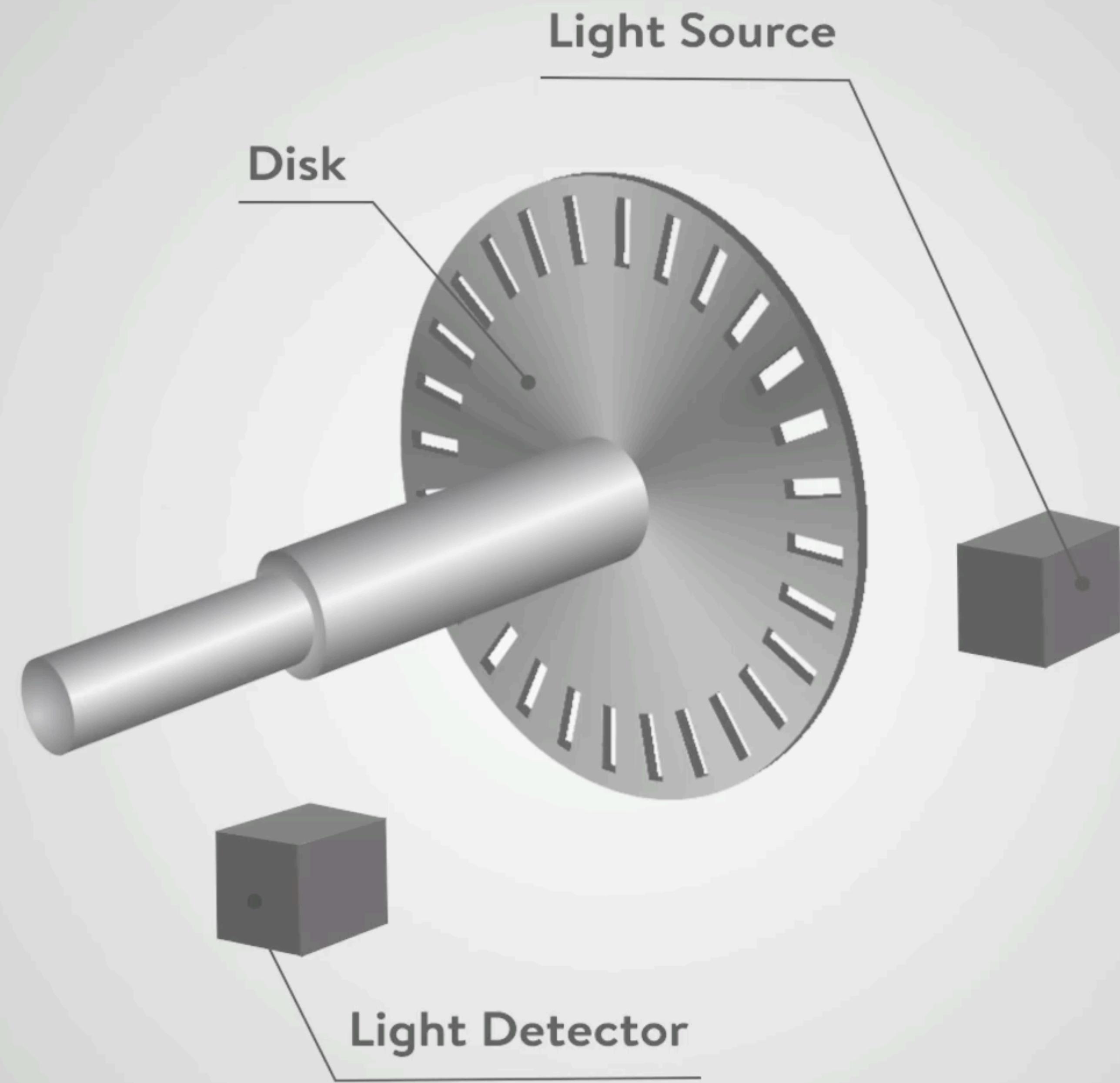
Os encoders são dispositivos que convertem movimento ou posição mecânica em sinais elétricos que podem ser lidos por um sistema de controle. Eles são usados em sistemas de automação, robótica, motores, e outras aplicações que exigem controle preciso de posição ou velocidade. Existem diferentes tipos de encoders, cada um adequado para uma aplicação específica.

Encoder Absoluto

Encoder Incremental

Encoder Óptico

Encoder Magnético

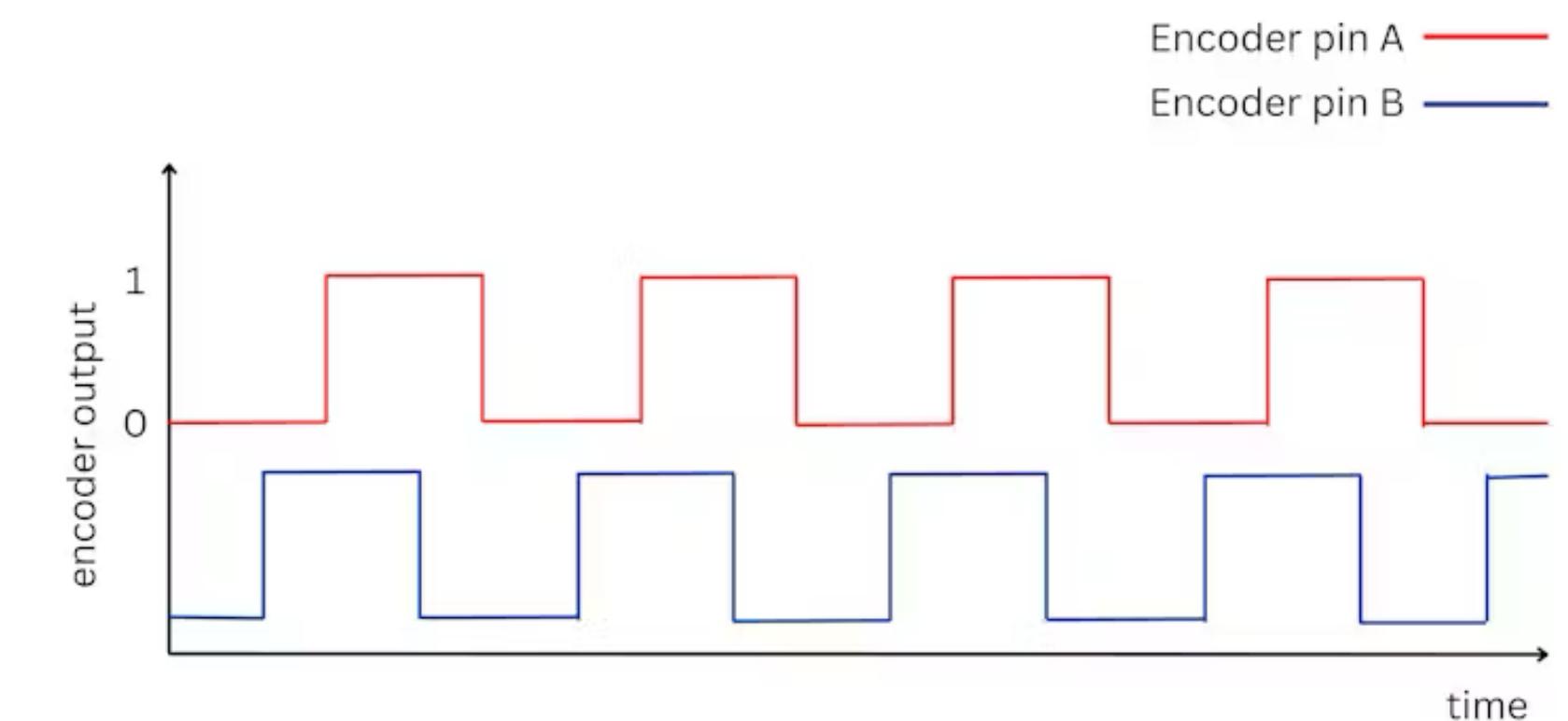
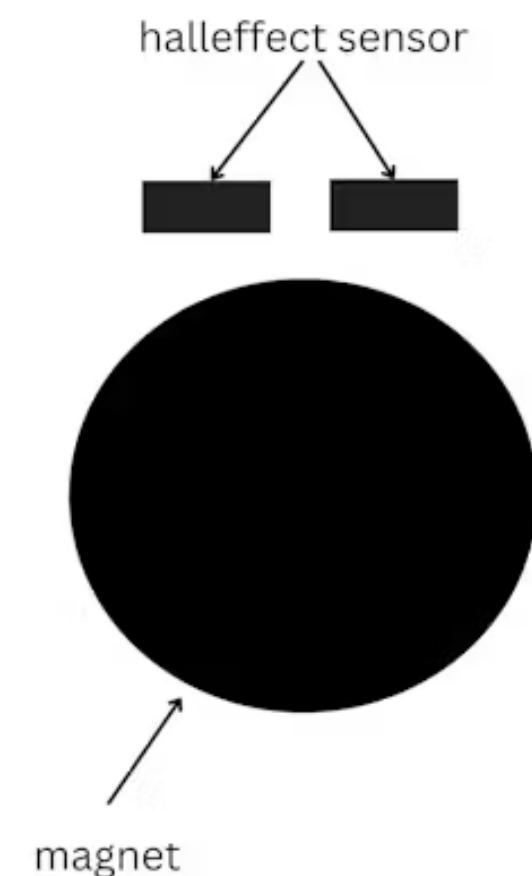
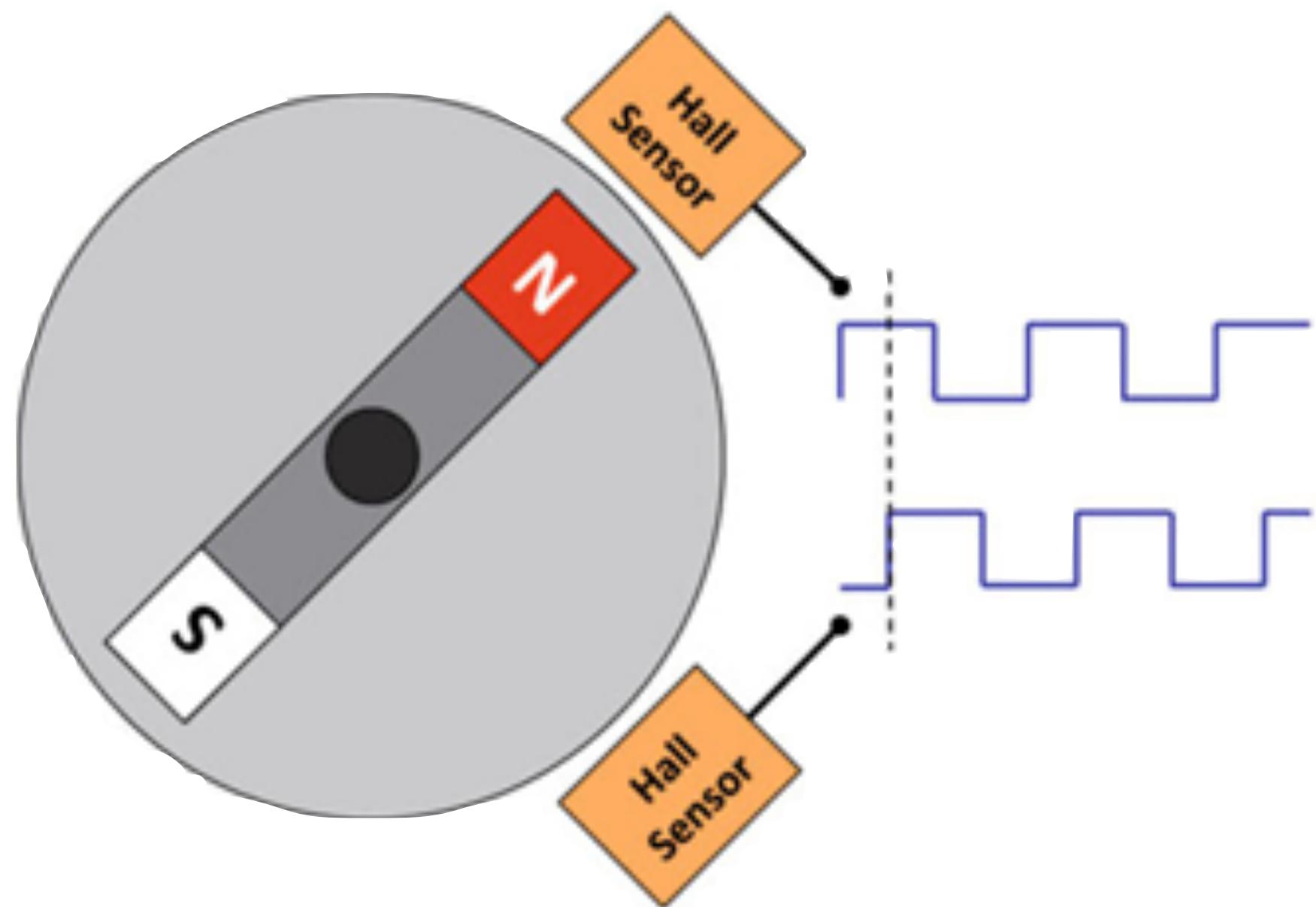


REALPARS

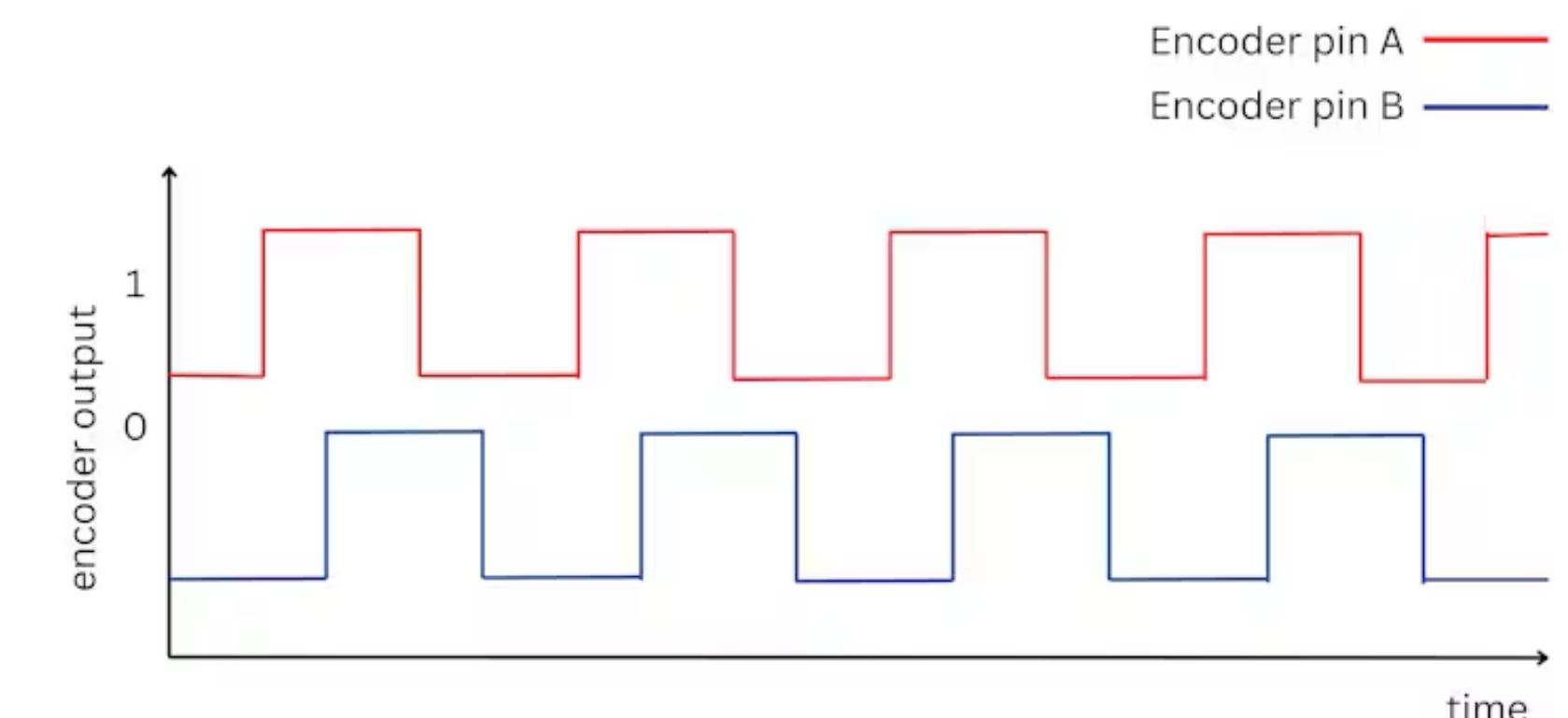
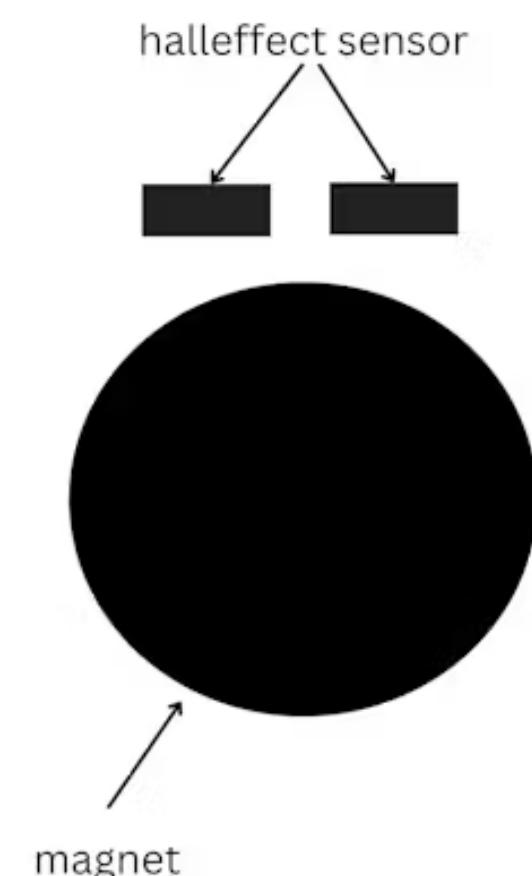
What is Encoder?

<https://www.youtube.com/watch?v=k2GQVJ4z0kM>

Encoders Magnéticos



Motor is rotating in anticlockwise direction



Motor is rotating in clockwise direction

Obrigado!

João Macedo
jmacedo@dei.uc.pt