

PI – Protótipos e Circuitos Elétricos

Nomes: João Victor Cavelagna 1012022200891

Rodrigo T Generoso 1012023100549

Henrique Junio Ferreira de Freitas 1012021200166

Ana Rafaela Mauricio 1012023100443

Carlos Eduardo da Silva 1012023100534

Objetivos

- Este trabalho tem como objetivo o estudo e o desenvolvimento de um sistema de controle na plataforma de prototipagem Arduino, para um braço mecânico multipropósito (braço robótico), que nasceu da necessidade de um aprofundamento do estudo no ramo da robótica.
- O braço mecânico tem controle automático, programável e reprogramável para quatro graus de liberdade do protótipo.
- Esta tecnologia é adotada por muitas fábricas e indústrias.

Funções

1. Manuseio de materiais: Um braço robótico pode ser utilizado para manuseio de material com grandes cargas, a fim de preservar a saúde de pessoas.

2. Montagem de estruturas: O braço robótico pode ser usado para montar estruturas pré-fabricadas, como painéis de paredes e pisos.

3. Controle de materiais: O braço robótico pode ser utilizado para ter um maior controle de um material utilizado em suas construções, a fim de desperdiçar menos os materiais e ter uma redução no custo geral de sua construção.

4. Inspeção de estruturas: O braço robótico pode ser equipado com câmeras e sensores para inspecionar estruturas, como pontes e prédios, para detectar possíveis problemas.

Em resumo, um braço robótico pode ser uma ótima ferramenta para aumentar a produtividade e a segurança em projetos de engenharia civil.

Componentes

Abaixo listam-se os materiais utilizados na confecção final do protótipo robótico:

1x Uno R3 + Cabo Usb para Arduino

1x Protoboard 400 Pontos

1x Jumpers – Macho/Macho – 20 Unidades de 20cm

1x Sensor Shield V5.0 para Arduino

1x Fonte 5V 1A Bivolt

4x Micro Servos 9g SG90 Tower Pro – 180°

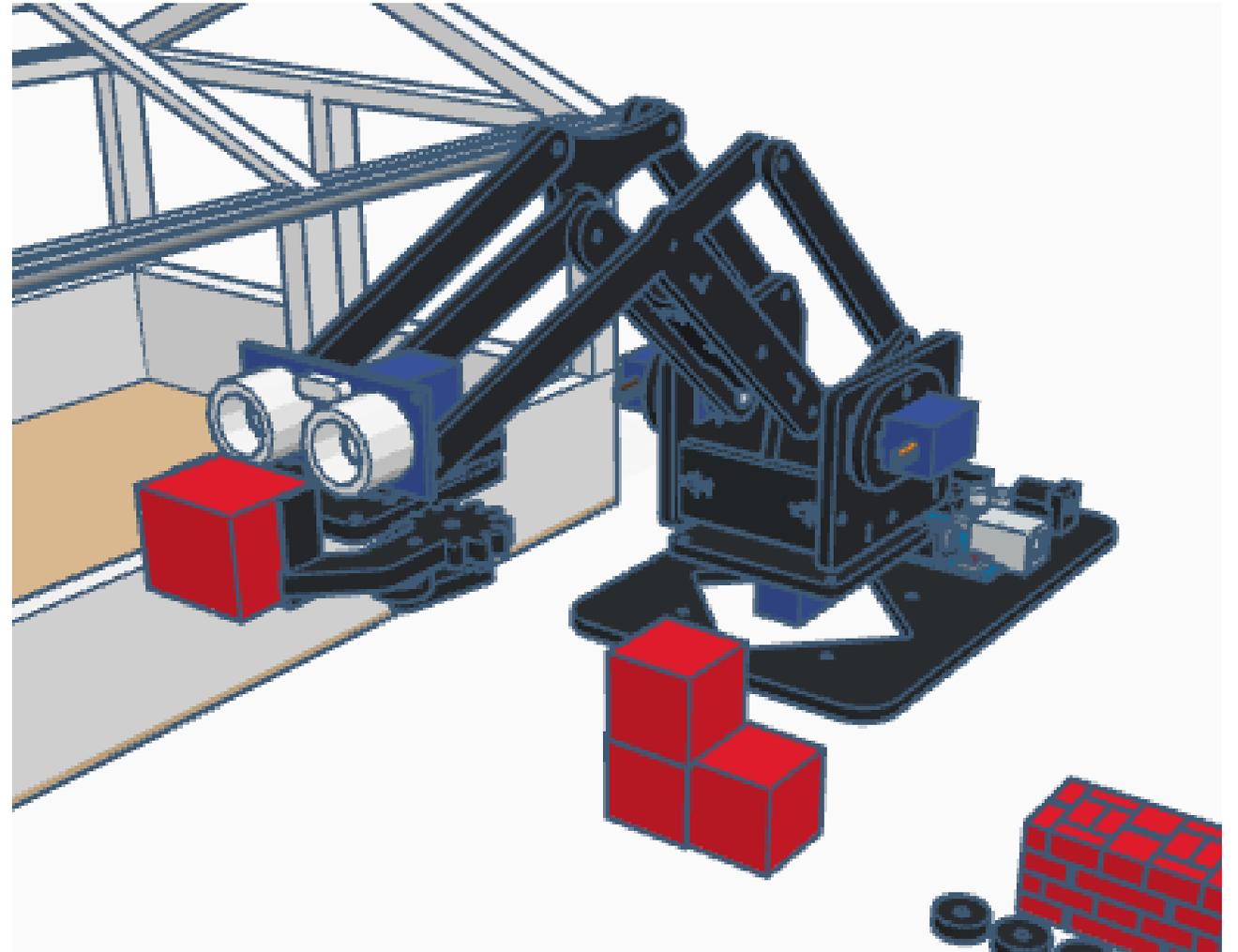
4x Potenciômetro Linear 10KΩ

1x Kit Braço Robótico em MDF

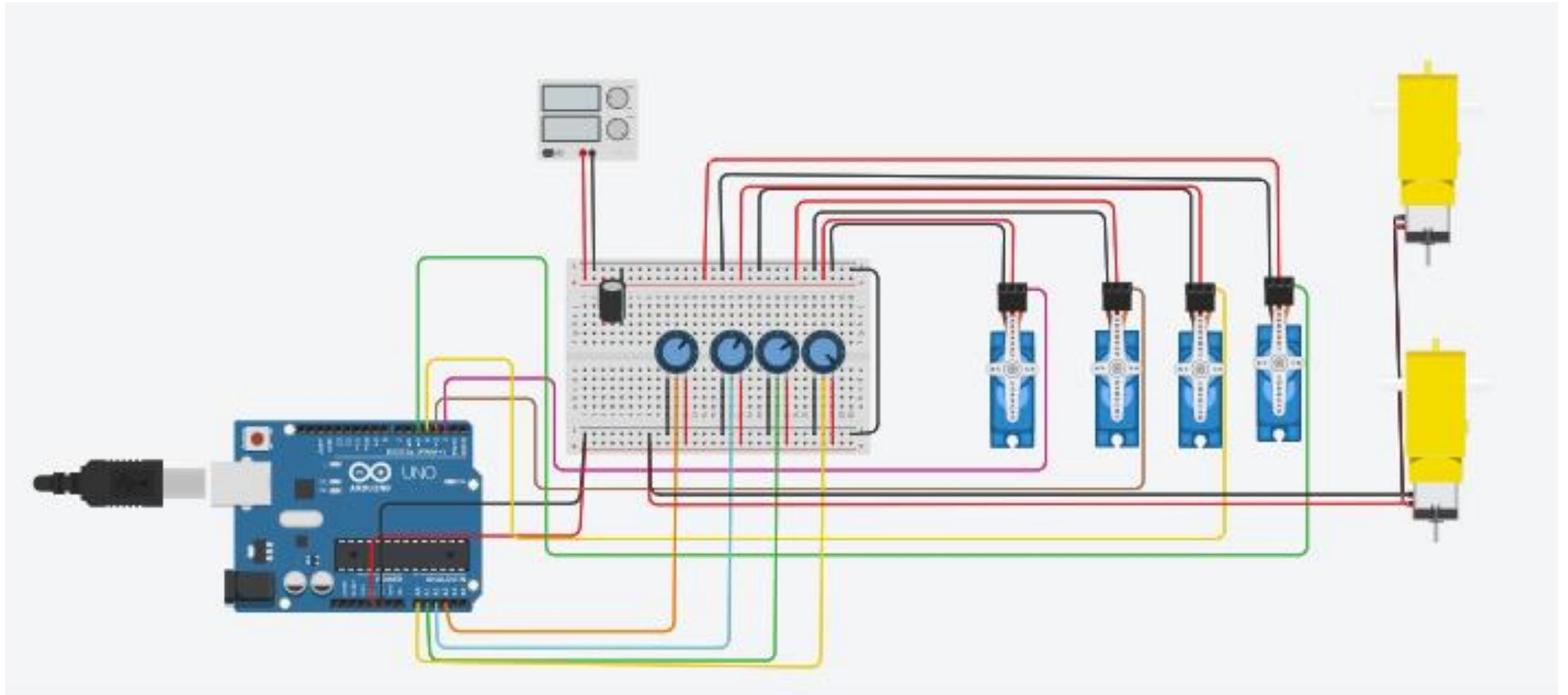
1x Sensor de distância

1x Fonte de energia 5W

2x Motor de engrenagem



Circuito e Programação



Programação

```
#include <Servo.h>
// define pinos dos servos#define pinServ1
2#define pinServ2 3#define pinServ3 4#define
pinServ4 5
```

```
// define as portas dos potenciômetros
```

```
#define pot1 A0
```

```
#define pot2 A1#define pot3 A2
```

```
#define pot4 A3
```

```
// nomeia os servosServo
```

```
serv1,serv2,serv3,serv4;
```

```
// cria as variáveis dos ângulos de cada motor
```

```
Int motor1,motor2,motor3,motor4;
```

```
unsigned long mostradorTimer = 1;const
unsigned long intervaloMostrador = 5000;
```

```
void setup() {
```

```
//inicia o monitor serial Serial.begin(9600);
```

```
// atribui pinos dos servos
serv1.attach(pinServ1); serv2.attach(pinServ2);
serv3.attach(pinServ3); serv4.attach(pinServ4);
}
```

```
void loop(){
```

```
// leitura dos potenciômetros
motor1 =
map(analogRead(pot1),0,1023,0,180);
motor2 =
map(analogRead(pot2),0,1023,0,180);
motor3 =
map(analogRead(pot3),0,1023,85,180);
motor4 =
map(analogRead(pot4),0,1023,0,43); //
posicionamento dos potenciômetros
serv1.write(motor1); serv2.write(motor2);
serv3.write(motor3); serv4.write(motor4);
```

```
//posicionamento dos potenciômetros
serv1.write(motor1);
serv2.write(motor2);
serv3.write(motor3);
serv4.write(motor4);
```

```
if ((millis() - mostradorTimer) >=
intervaloMostrador) {
```

```
// envio para o monitor serial do
posicionamentos dos motores
```

```
Serial.println("*****");
Serial.print("Pot1:");
Serial.print(analogRead(pot1));
Serial.print(" Angulo Motor1:");
Serial.println(motor1);
```

```
Serial.print("Pot2:");  
Serial.print(analogRead(pot2));  
Serial.print(" Angulo Motor2:");  
Serial.println(motor2);
```

```
Serial.print("Pot3:");  
Serial.print(analogRead(pot3));  
Serial.print(" Angulo Motor3:");  
Serial.println(motor3);
```

```
Serial.print("Pot4:");  
Serial.print(analogRead(pot4));  
Serial.print(" Angulo Motor4:");  
Serial.println(motor4);
```

```
mostradorTimer = millis();  
  
}  
  
// tempo de espera para recomeçar  
delay(100);  
  
}
```