

GUILHERME FABIANO TERRA DA SILVA

**ROBÓTICA**

Lixeira Automática

**ORIENTADOR:** Prof. Leandro

**2°Ano C**

**GUARATINGUETÁ – SP**

**06 / 11 / 20**

****

GUILHERME FABIANO TERRA DA SILVA

**TRABALHO DE ROBÓTICA**

Lixeira Automática

Trabalho de Robótica realizado no curso de

automação industrial, como forma de ampliação

de conhecimento e obtenção parcial de nota.

**Orientador:** Prof. Pimenta

GUARATINGUETÁ

06 / 11 / 20

**Sumário**

[**1. SITUAÇÃO PROBLEMA** 4](#_Toc53928969)

[**1.1. Introdução** 4](#_Toc53928970)

[**1.2. Nosso projeto** 4](#_Toc53928971)

[**2. OBJETIVO DO TRABALHO** 6](#_Toc53928972)

[**3. ETAPAS DO PROJETO** 6](#_Toc53928973)

[**4. COMO REALIZAR O PROJETO** 7](#_Toc53928974)

[**4.1. Como testar se a lixeira está com uma sacola** 7](#_Toc53928975)

[**4.2. Como fechar a sacola** 7](#_Toc53928976)

[**4.3. Como tirar a sacola da lixeira** 7](#_Toc53928977)

[**4.4. Como testar se ela já chegou aos 90% do espaço total da lixeira** 7](#_Toc53928978)

[**5. FLUXOGRAMA** 8](#_Toc53928979)

[**6. COMO FUNCIONA O SENSOR ULTRASSÔNICO** 8](#_Toc53928980)

[**7. COMO FUNCIONA O CI L293D** 9](#_Toc53928981)

[**8. SIMULAÇÂO DO FDC ATRAVÉS DE BOTÃO** 10](#_Toc53928982)

[**9. DIAGRAMA DO PROJETO** 10](#_Toc53928983)

[**10. PROGRAMAÇÃO DO PROJETO** 11](#_Toc53928984)

[**10.1. Início da Programação** 11](#_Toc53928985)

[10.1.1. Definições Iniciais 11](#_Toc53928986)

[1.1.2. Configuração dos pinos/bits do Arduino 12](#_Toc53928987)

[10.1.3. Programa Principal 14](#_Toc53928988)

[10.1.4. Função que captura o tempo do envio até o retorno do sinal ultrassônico 20](#_Toc53928989)

# **1. SITUAÇÃO PROBLEMA**

## **1.1. Introdução**

Durante toda a sua existência, o homem presenciou e agiu para que diversas evoluções acontecessem, desde a mais simples criação até a mais desenvolvida tecnologia atual. E nesses processos de desenvolvimento, a automatização é uma importante evolução, que facilita e auxilia as ações humanas de diversas maneiras. Sendo assim, é muito importante que estejamos em constante progresso e desenvolvimento, para que as mais diversas áreas possam ser ajudadas com a utilização de sistemas robóticos que dão total apoio para que sejam realizadas as mais diversas tarefas.

## **1.2. Nosso projeto**

Algo que está presente na vida de provavelmente todos nós, são os resíduos produzidos por nós nas mais diversas áreas, seja o de casa, o hospitalar, o industrial, dentre outros.

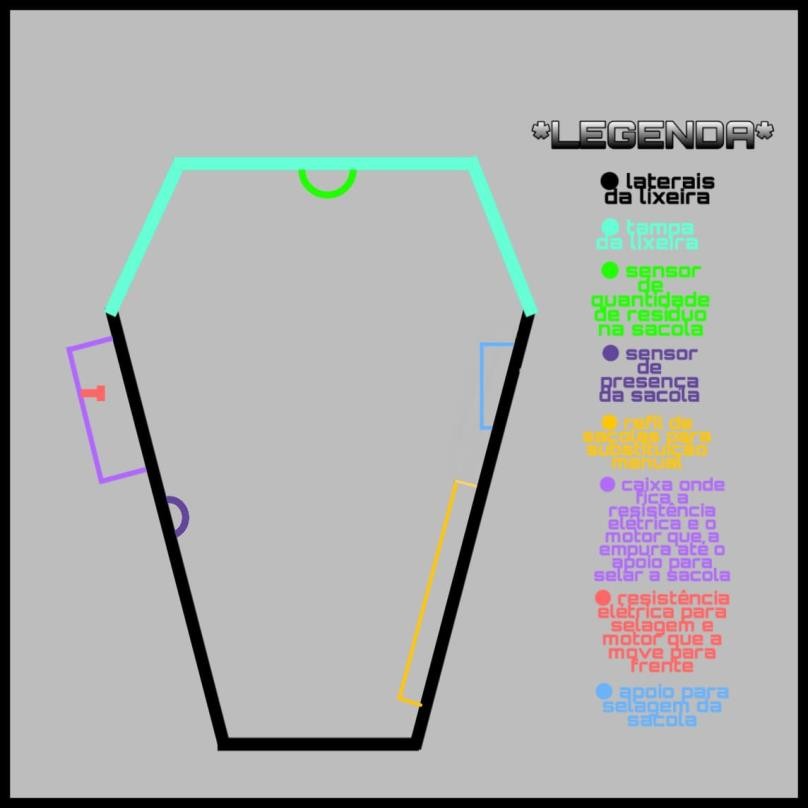
Todo esse resíduo passa por manipulação para que possa ser descartado, assim dando lugar para o resíduo mais recente, que também será descartado. Porém, algo que todos sabemos é que esse lixo produzido é sujo, contendo diversas bactérias que fazem mal para nós seres humanos e mesmo aqueles que fazem a limpeza com preparo e equipamentos adequados, estão sujeitos a acidentalmente ter contato com esses dejetos, podendo se infectar ou contrair doenças, pois muitas vezes o lixo de hospitais ou industrias é nocivo e deve ser descartado imediatamente para que não ofereça riscos à saúde.

Pensando nisso, nosso projeto visa a criação de uma lixeira automatizada, que realizara todos os processos de forma automática, facilitando o trabalho de quem faz a manutenção e limpeza dos locais que possuem lixeiras.

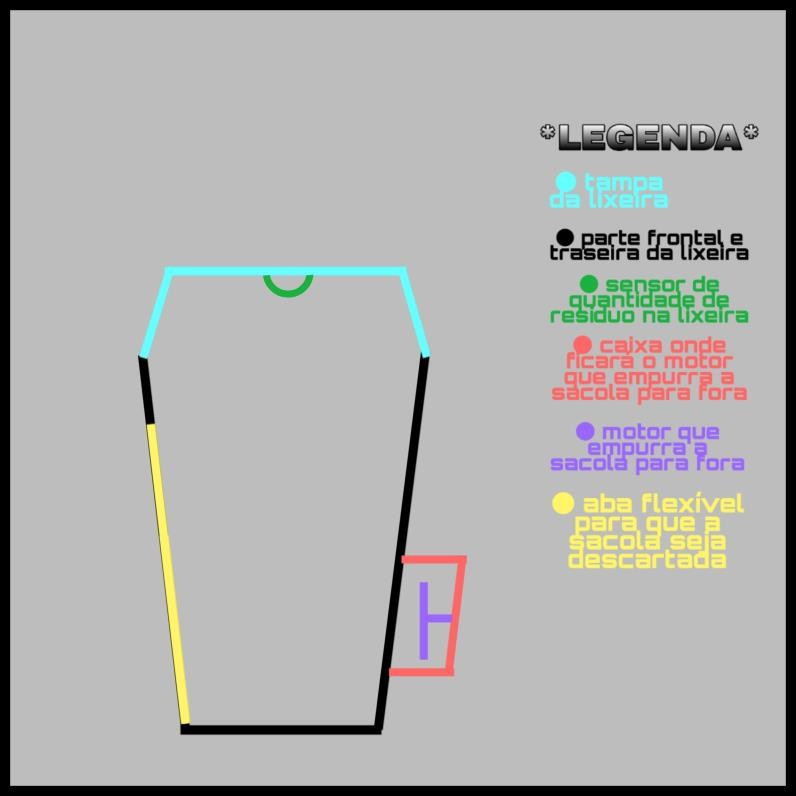
A ideia desse projeto é uma lixeira que evite ao máximo o contato com a própria lixeira e com a sacola onde se situam os dejetos, de forma que com alguns sensor de proximidade seja possível verificar o nível do lixo na lixeira e testar se a lixeira tem uma sacola, quando o sensor que verifica o nível da lixeira atingir 90% do tamanho da lixeira ou alguém pressionar o botão de retirar sacola realiza-se o selamento da sacola e em seguida a remoção da sacola da lixeira.

 Abaixo, veja a esquematização desta lixeira:

**Imagem 1 –** Exterior da lixeira com as identificações



**Imagem 2 –** Interior das laterais da lixeira com as identificações



**Imagem 3 –** Interior das partes frontal e traseira da lixeira com as identificações

# **2. OBJETIVO DO TRABALHO**

O objetivo do nosso trabalho é criar uma lixeira que automatize o processo de captação do lixo domiciliar, tornando-o mais rápido e evitando o contato da pessoa com o lixo.

# **3. ETAPAS DO PROJETO**

**Etapa 1:** Inserir a sacola no lixo, grudando-a nos lados e abrindo a sacola, em seguida alimentar o Arduino UNO.

**Etapa 2:** Verificar quanto do nível da lixeira está ocupado.

**Etapa 2.1:** Se estiver com mais de 90% ocupada ela solta a sacola dos lados e fecha, empurrando a sacola para fora da lixeira por meio de uma aba flexível, otimizando o trabalho de quem irá retirar o lixo, pois já entrega a sacola totalmente fechada e do lado de fora da lixeira.

**Etapa 2.2:** Se não chegou aos 90% não acontece nada.

**Etapa 3:** Retorna ao início do projeto.

# **4. COMO REALIZAR O PROJETO**

## **4.1. Como testar se a lixeira está com uma sacola**

Para testar se temos uma sacola na lixeira podemos usar um sensor ultrassônico em uma posição que a sacola esteja no seu “campo de visão”.

## **4.2. Como fechar a sacola**

Para selar a sacola usaremos uma resistência elétrica que será esquentada e levada até a sacola, e para fazer isso usaremos mais uma placa que será menor que a placa que empurra a sacola. Esta nova placa irá ir até a sacola e pressiona-la contra um apoio lateral da lixeira, fazendo com que os 2 lados da sacola se toquem e a resistência encoste na sacola, selando-a.

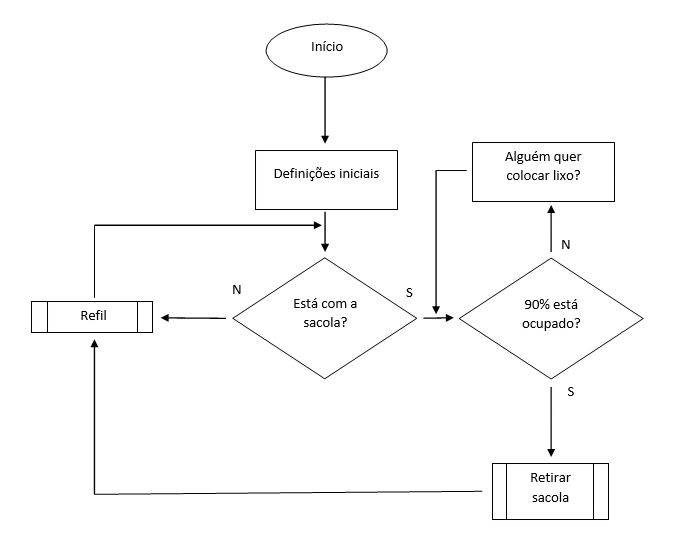
## **4.3. Como tirar a sacola da lixeira**

Para tirar a sacola da lixeira usaremos um motor CC que ao ser ativado fará com que uma placa empurre a sacola para fora da lixeira, além disso utilizaremos um botão simulando uma chave fim de curso para parar o motor após ele empurrar a sacola pra fora do lixo, e em seguida retornar ao ponto inicial invertendo a rotação através do **CI** **L293D** e usando outro botão como chave fim de curso para indicar quando ele voltou a posição inicial.

## **4.4. Como testar se ela já chegou aos 90% do espaço total da lixeira**

Para verificar se a ocupação da lixeira já passou ou não do limite (90% do tamanho da lixeira) usaremos um sensor ultrassônico.

# **5. FLUXOGRAMA**

****

S

N

Botão foi pressionado?

# **6. COMO FUNCIONA O SENSOR ULTRASSÔNICO**

O sensor ultrassônico é um sensor que envia uma onda sonora que apresenta frequência acima da frequência que o ser humano pode ouvir (por isso ULTRAssônico), e, de acordo com o princípio de reflexão sonora essa onda incide em um obstáculo e retorna ao meio de origem, desta maneira é possível captar o tempo em que a onda saiu e voltou ao sensor, e, para calcular a distância basta considerar a velocidade do som no ar como 343 m/s. Porém, devemos lembrar que a onda vai até o objeto e retorna, ou seja, a onda faz o percurso 2 vezes, logo para obter a distância correta basta calcular a distância através da fórmula: d = v \* t; e em seguida dividir por 2, já que a onda vai e volta.

O Tinkercad apresenta dois sensores ultrassônicos, um com 4 pinos e outro com 3 pinos, sendo que:

O de 4 pinos recebe – VCC, GND, ENTRADA DE ONDA e SAÍDA DE ONDA

O de 3 pinos recebe – VCC, GND e ENTRADA / SAÍDA DE ONDA

No nosso projeto utilizamos o sensor de 3 pinos para economizar as portas do Arduino, lembrando que seu pino faz tanto a função de enviar quanto de captar a onda, e, para isso, o pino varia entre INPUT e OUTPUT.



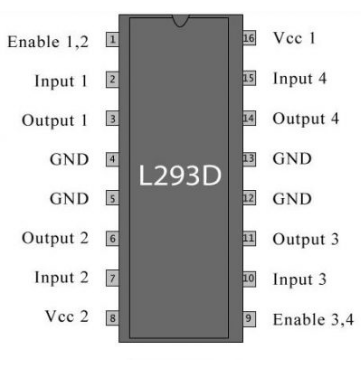
**Imagem 4 –** Sensor Ultrassônico

# **7. COMO FUNCIONA O CI L293D**

Este CI possui 2 pontes H completas e isso permite que ele controle até 2 motores, sendo que ele é alimentado com 5V e pode suportar tensões de 4,5V até 36V.

Nos pinos 1 e 9 do CI L293D podemos colocar valores de 0 a 255 e eles representam a velocidade do motor, sendo que 0 é a velocidade mínima do motor e 255 a velocidade máxima do motor.

**Pinagem do CI L293D:**

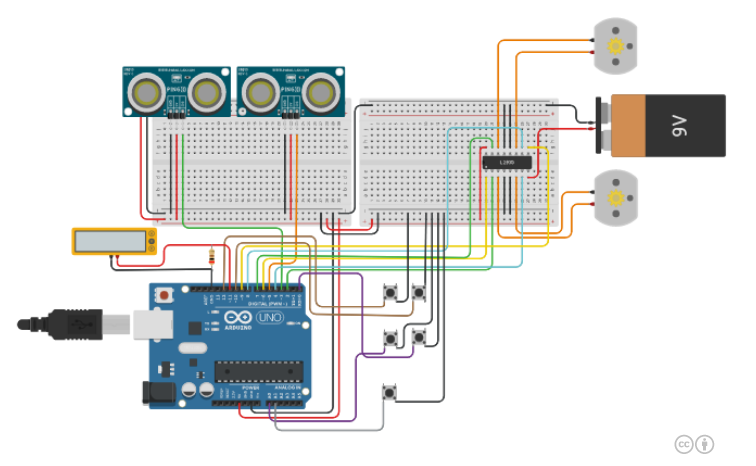
****

**Imagem 5 –** CI L293D

# **8. SIMULAÇÂO DO FDC ATRAVÉS DE BOTÃO**

Para simular as chaves fim de curso utilizamos botões que durante a rotação do motor serão testados e quando forem pressionados o motor para.

# **9. DIAGRAMA DO PROJETO**

****

**Imagem 6 –** Circuito do Projeto

# **10. PROGRAMAÇÃO DO PROJETO**

## **10.1. Início da Programação**

### 10.1.1. Definições Iniciais

**int sensor\_1 = 3; // Sensor que verifica se a lixeira tem sacola**

**float distancia\_1; // Variável de verificação de distância**

**float segunda\_etapa; // Variável que controla segunda parte do Projeto**

**int sensor\_2 = 5; // Sensor que verifica se a lixeira tem sacola**

**float distancia\_2; // Variável de verificação de distância**

**int controle = 6; // Controle de Velocidade do Motor1 para Selar sacola**

**int M1 = 2; // Contorle de Sentido de Rotação**

**int M2 = 4; // Controle de Sentido de Rotação**

**int controle\_2 = 9; // Controle de Velocidade do Motor1 para Selar sacola**

**int M1\_2 = 7; // Contorle de Sentido de Rotação**

**int M2\_2 = 8; // Controle de Sentido de Rotação**

**int botao\_FDC1 = 10; // FDC simulada do Motor 1**

**int botao\_FDC2 = 12; // FDC2 simulada do Motor 1**

**int botao\_2\_FDC1 = 0; // FDC simulada do Motor 2**

**int botao\_2\_FDC2 = 14; // FDC simulada do Motor 2**

**int bot\_remove = 15; // Botão que retira sacola**

**int V\_Resistencia = 11; // Controle da Tensão da Porta 11**

**int temp; // Controle de Temperatura da R**

**float rem = 0.0;**

**void teste\_1(); // Pegando valor da distância**

**void teste\_2(); // Pegando valor da distância 2**

### 1.1.2. Configuração dos pinos/bits do Arduino

**void setup()**

**{**

**Serial.begin(9600);**

**pinMode(sensor\_1, OUTPUT);**

**digitalWrite(sensor\_1, LOW);**

**pinMode(sensor\_2, OUTPUT);**

**digitalWrite(sensor\_2, LOW);**

**pinMode(controle, OUTPUT);**

**digitalWrite(controle,LOW);**

**analogWrite(controle, 0);**

**pinMode(M1, OUTPUT);**

**digitalWrite(M1,LOW);**

**pinMode(M2, OUTPUT);**

**digitalWrite(M2,LOW);**

**digitalWrite(controle\_2,LOW);**

**analogWrite(controle\_2, 0);**

**pinMode(M1\_2, OUTPUT);**

**digitalWrite(M1\_2,LOW);**

**pinMode(M2\_2, OUTPUT);**

**digitalWrite(M2\_2,LOW);**

**pinMode(botao\_FDC1, INPUT\_PULLUP);**

**pinMode(botao\_FDC2, INPUT\_PULLUP);**

**pinMode(botao\_2\_FDC1, INPUT\_PULLUP);**

**pinMode(botao\_2\_FDC2, INPUT\_PULLUP);**

**pinMode(bot\_remove, INPUT\_PULLUP);**

**pinMode(V\_Resistencia, OUTPUT);**

**digitalWrite(V\_Resistencia, LOW);**

**}**

### 10.1.3. Programa Principal

**void loop()**

**{**

**// Primeira etapa do Projeto**

**teste\_1();**

**distancia\_1 = distancia\_1 / 2; // Calculando dist**

**distancia\_1 = distancia\_1 \* 0.034029; // Calculando dist**

**if (distancia\_1 < 30){**

**Serial.println("Tem sacola na Lixeira");**

**segunda\_etapa = 1.0;**

**}else{**

**Serial.println("NAO Tem sacola na lixeira");**

**segunda\_etapa = 0.0;**

**}**

**delay(300);**

**while (segunda\_etapa == 1.0){**

**// Segunda Etapa do Projeto**

**teste\_2();**

**distancia\_2 = distancia\_2 / 2; // Calculando dist**

**distancia\_2 = distancia\_2 \* 0.034029; // Calculando dist**

**if (distancia\_2 <= 11 || rem == 1.0){ // 110 (Tamanho máximo da lixeira) - 90% da capacidade = 11cm**

**Serial.println("Hora de tirar o Lixo !!");**

**// Esquentar resistência**

**for(temp=0; temp <= 255; temp++){**

**analogWrite(V\_Resistencia, temp);**

**delay(100);**

**}**

**// Selar a Sacola (Motor)**

**analogWrite(controle, 100);**

**digitalWrite(M1, HIGH);**

**digitalWrite(M2, LOW);**

**// Levar a resistência até a sacola**

**float F\_ctrl = 0.0;**

**while (F\_ctrl == 0.0){**

**if(digitalRead(botao\_FDC1) == 0){**

**Serial.println("Resistencia chegou");**

**F\_ctrl = 1.0;**

**}else{**

**Serial.println("Resistencia indo ate a sacola ...");**

**}**

**}**

**// Fim da ida**

**Serial.println("Selando a sacola . . .");**

**analogWrite(controle, 0);**

**digitalWrite(M1, LOW);**

**digitalWrite(M2, LOW);**

**delay(3000); // Tempo de selamento**

**temp = 0;**

**analogWrite(V\_Resistencia, temp);**

**analogWrite(controle, 100);**

**digitalWrite(M1, LOW);**

**digitalWrite(M2, HIGH);**

**// Retorna a Resistência ao ponto inicial**

**float F\_ctrl2 = 0.0;**

**while (F\_ctrl2 == 0.0){**

**if(digitalRead(botao\_FDC2) == 0){**

**Serial.println("Resistencia voltou ao ponto inicial");**

**F\_ctrl2 = 1.0;**

**}else{**

**Serial.println("Resistencia voltando ...");**

**}**

**}**

**// Fim da volta**

**analogWrite(controle, 0);**

**digitalWrite(M1, LOW);**

**digitalWrite(M2, LOW);**

**delay(3000);**

**// Empurrar para Fora**

**analogWrite(controle\_2, 100);**

**digitalWrite(M1\_2, HIGH);**

**digitalWrite(M2\_2, LOW);**

**// Inicio da ida**

**float F2\_ctrl1 = 0.0;**

**while (F2\_ctrl1 == 0.0){**

**if(digitalRead(botao\_2\_FDC1) == 0){**

**Serial.println("Sacola fora");**

**F2\_ctrl1 = 1.0;**

**}else{**

**Serial.println("Empurrando para fora ...");**

**}**

**}**

**// Fim da ida**

**analogWrite(controle\_2, 0);**

**digitalWrite(M1\_2, LOW);**

**digitalWrite(M2\_2, LOW);**

**delay(3000);**

**analogWrite(controle\_2, 100);**

**digitalWrite(M1\_2, LOW);**

**digitalWrite(M2\_2, HIGH);**

**// Inicio da volta**

**float F2\_ctrl2 = 0.0;**

**while (F2\_ctrl2 == 0.0){**

**if(digitalRead(botao\_2\_FDC2) == 0){**

**Serial.println("Retornou a posicao inicial");**

**F2\_ctrl2 = 1.0;**

**}else{**

**Serial.println("Voltando ao ponto inicial ...");**

**}**

**}**

**// Fim da volta**

**analogWrite(controle\_2, 0);**

**digitalWrite(M1\_2, LOW);**

**digitalWrite(M2\_2, LOW);**

**delay(3000);**

**// Reiniciar o Projeto**

**segunda\_etapa = 0.0;**

**rem = 0.0;**

**}else{**

**Serial.println("Ainda tem espaco suficiente !! ");**

**// Retirar lixo na hora**

**if(digitalRead(bot\_remove) == 0){**

**Serial.println("Remover lixo agora");**

**rem = 1.0;**

**}else{**

**Serial.println("Nao remover Lixo");**

**}**

**}**

**delay(300);**

**}**

**}**

### 10.1.4. Função que captura o tempo do envio até o retorno do sinal ultrassônico

**void teste\_1() // Pegando valor da distância**

**{**

**pinMode(sensor\_1,OUTPUT);**

**digitalWrite(sensor\_1, HIGH);**

**delayMicroseconds(5);**

**digitalWrite(sensor\_1,LOW);**

**pinMode(sensor\_1, INPUT);**

**distancia\_1 = pulseIn(sensor\_1,HIGH);**

**}**

**void teste\_2() // Pegando valor da distância**

**{**

**pinMode(sensor\_2,OUTPUT);**

**digitalWrite(sensor\_2, HIGH);**

**delayMicroseconds(5);**

**digitalWrite(sensor\_2,LOW);**

**pinMode(sensor\_2, INPUT);**

**distancia\_2 = pulseIn(sensor\_2,HIGH);**

**}**

**REFERÊNCIAS**

Link1: <https://balluffbrasil.com.br/sensor-ultrassonico-como-ele-funciona-e-de-que-modo-pode-ajudar-a-sua-industria/>

Link2: <https://www.youtube.com/watch?v=QdjhJhORJMI&t=6s>

Link3: <https://create.arduino.cc/projecthub/sandromesquitamecatronica/lixeira-automatizada-4dad81>

Link4: <https://www.youtube.com/watch?v=m_znuYr8-zo>

Link5: <http://www.bosontreinamentos.com.br/eletronica/controlador-l293d-ponte-h-dupla/>

Link6: <https://www.youtube.com/watch?v=q04QsOcxg94&list=PLx4x_zx8csUgWBTvA-fluHV970SzDJRBw&index=23>

Link7: <https://www.youtube.com/watch?v=vBEloNGfFpg&list=PLx4x_zx8csUgWBTvA-fluHV970SzDJRBw&index=21>

Link8: <https://www.youtube.com/watch?v=9LMe9MPzneg&list=PLx4x_zx8csUgWBTvA-fluHV970SzDJRBw&index=20>