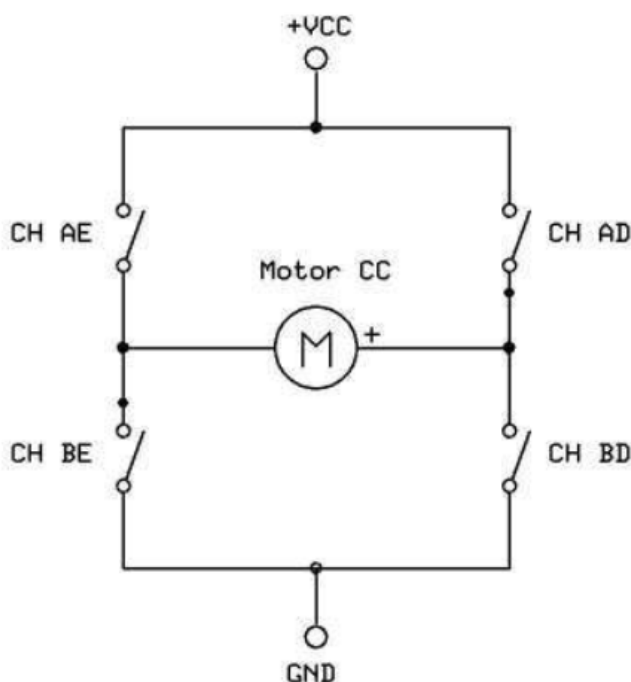


## Sumário

Sumário .....	3
<b>Parte I - Revisão de circuitos elétricos e componentes básicos.....</b>	<b>4</b>
<b>Introdução .....</b>	<b>4</b>
<b>Revisão de circuitos elétricos.....</b>	<b>5</b>
Carga e corrente elétrica .....	6
Tensão elétrica .....	6
Potência e energia.....	7
Conversão de níveis lógicos e alimentação correta de circuitos.....	8
<b>Revisão de componentes eletrônicos .....</b>	<b>9</b>
Resistores elétricos .....	9
Capacitores.....	11
<b>Parte II - Seção de Exemplos Práticos.....</b>	<b>16</b>
Exemplo 1 - Leitura de Sinais Analógicos com Potenciômetro .....	16
Exemplo 2 - Divisores de Tensão com Resistores .....	19
Exemplo 3 - Controle de LED's com Botões .....	21
Exemplo 4 - Acionamento de LED conforme do Luz Ambiente.....	24
Exemplo 5 - Controle de Servo Motor com Potenciômetro.....	27
Exemplo 6 - Medindo a Temperatura do Ambiente .....	30
Exemplo 7 - Acendendo um LED com Sensor Reflexivo Infravermelho .....	33
Exemplo 8 - Trena Eletrônica com Sensor Ultrassônico.....	36
Exemplo 9 – Controle de Motores CC com Ponte H L298N.....	40
Exemplo 10 – Carro Automático com Ponte H e Arduino.....	49
Exemplo 11 – Braço Robótico Controlado por Potenciômetros .....	62
Considerações finais .....	65

## Exemplo 9 – Controle de Motores CC com Ponte H L298N

O uso de Ponte H é interessante para diversas implementações com motores, porque estes componentes permitem uma série de facilidades. Então, o primeiro ponto a citar é que a ponte H nos permite girar o motor para ambos os lados, com um circuito que funciona da seguinte forma:



Se queremos que o motor funcione no sentido 1, fechamos somente as chaves CH AD e CH BE. Se caso queremos que gire no sentido 2, abrimos as anteriores e fechamos CH AE e CH BD.

Então, por meio de comutação de chaves, a ponte H consegue fazer com que um motor gire nos dois sentidos. Como no módulo Ponte H L298N essa comutação não é feita de forma física, existe um chip (L298N) embarcado na placa que faz essa comutação de chaves de forma eletrônica, que é mais eficiente do que uma comutação mecânica.

As especificações do chip L298, que faz todo o controle da ponte H, pode ser conferido clicando neste link: <https://www.st.com/resource/en/datasheet/l298.pdf>.

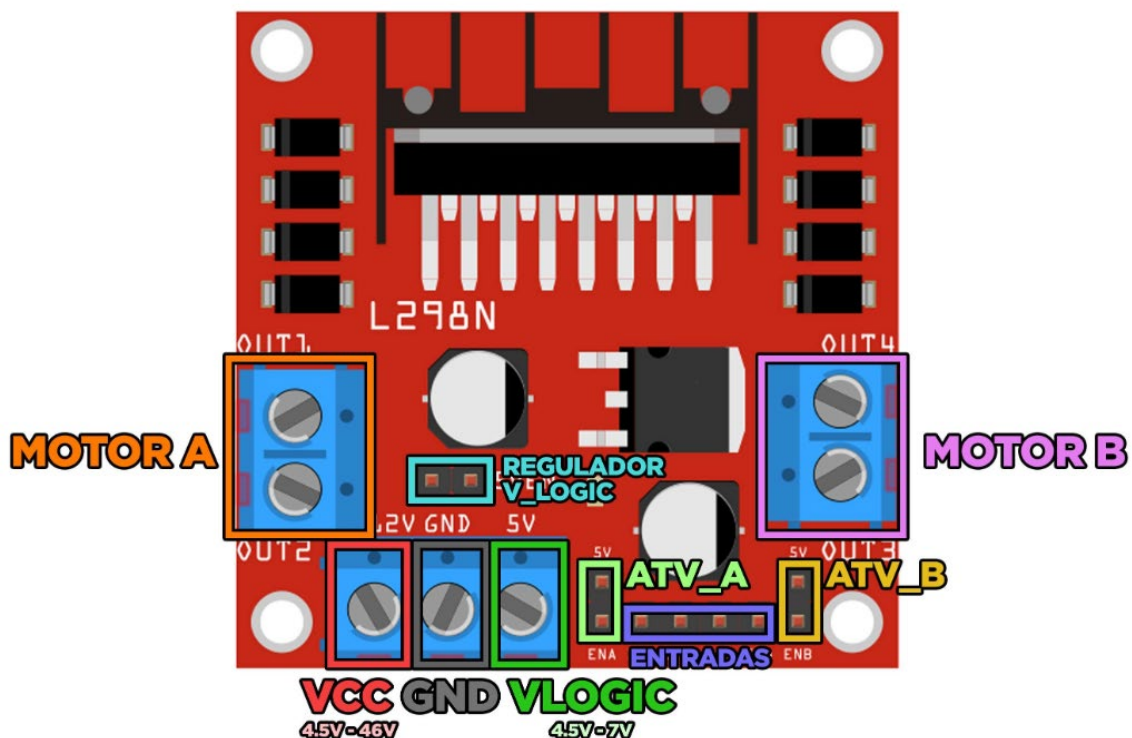
As principais especificações são:

- Tensão de Operação: 4.5 V~46 V
- Controle de 2 motores DC ou 1 motor de passo
- Corrente de Operação máxima: 2 A por canal ou 4 A total.

- Tensão lógica: 4.5 V ~ 7 V
- Corrente lógica: 0~36 mA
- Limites de Temperatura: -20 a +135 °C
- Potência Máxima: 25 W
- Dimensões: 43 x 43 x 27 mm.

Como funciona a Ponte H L298N :

Abaixo será explicado com maiores detalhes como funciona a placa e como usar:



#### Vcc:

Este borne é responsável pela alimentação dos motores, com isso a tensão colocada neste borne de Vcc será copiada para os motores. Por isso, muita atenção: Se você possui um motor que opera somente até 12V, por exemplo, não alimente Vcc com mais de 12V, pois danificará o motor.

- **Observação:** Esta placa possui um regulador de tensão do modelo 78m05, e a tensão máxima dele é 35V, mas a margem segura para uso é abaixo de 20V, sendo o recomendado 12V. Por isso, se for usar tensão de alimentação Vcc acima de 12V, não use a porta VLogic para alimentar outros circuitos, e para tensões acima de 20V, desative o regulador de tensão (como será demonstrado

mais abaixo) e faça a alimentação Vlogic através do borne, observando sua limitação de tensão.

#### **GND:**

É o GND da placa, que deverá ser o mesmo GND do Arduino ou de outro microcontrolador. Por isso, é importante interligar os GNDs sempre com fios. Serão demonstrados na seção de exemplos.

#### **Vlogic:**

Este borne é responsável pela alimentação lógica da ponte H. O chip L298N presente na placa recebe comandos do Arduino, e para ele conseguir operar os comandos e atuar na saída ele precisa de operar como um componente lógico (como o Arduino), e para isso precisa atuar em uma faixa bem restrita de tensão, que no caso é 4.5V a 7V. Como foi dito a placa possui um regulador de tensão de 5V, que a função dele é justamente fazer a regulação de Vcc para a alimentação lógica do chip em 5V. Por isso, se o regulador estiver ativo, este pino Vlogic terá a tensão de 5V, pois, é a tensão presente na saída do regulador.

É possível desativar o regulador de tensão e fazer a alimentação lógica do chip por conta própria, através do borne Vlogic. Em algumas situações é necessário desativar o regulador de tensão e fazer a alimentação lógica através de Vlogic, essas situações serão abordadas e explicadas na seção de exemplos de uso da placa.

- **Importante:** Como dito, quando o regulador de tensão está ativo, a tensão VLogic está sendo regulada em 5V automaticamente, logo, este borne terá 5V. Essa tensão é para uso interno do chip e pode ser aproveitada para alimentar algum circuito ou microcontrolador que use até 5V, com a limitação de corrente de aproximadamente 200mA. **Porém, se o regulador de tensão estiver acionado, JAMAIS alimente este pino, pois causará danos a placa.**

#### **Regulador Vlogic:**

Este jumper é responsável por ativar ou desativar o regulador de tensão da alimentação de Vlogic. Quando o jumper está encaixado entre os terminais, o regulador de tensão está ativado e a Vlogic como a ser regulado em 5V a partir da tensão de Vcc. Por isso, é muito importante saber manusear esse jumper do regulador. Caso ficou dúvidas sobre o funcionamento do regulador, releia a parte Vlogic.

#### **Atv\_A:**

Este jumper é responsável por desativar ou ativar o motor A para ser controlado pelo Arduino ou outra placa.

Este jumper funciona da seguinte forma: um dos terminais possui 5V (ou a tensão de alimentação VLogic), e o outro terminal é o que vai para o chip L298N. A placa possui um resistor de pull-down no outro resistor, o que significa que quando o jumper é solto, o terminal ficará com nível lógico baixo, e o chip entenderá que é necessário desabilitar o motor A. Mas quando o jumper é colocado entre os terminais, o 5V é ligado ao terminal de sinal pelo jumper. Com isso, o sinal de nível alto é enviado ao chip L298n e ele entende que os motores devem estar habilitados para os acionamentos.

#### Atv\_B:

Este jumper é responsável por desativar ou ativar o motor B para ser controlado. O funcionamento é exatamente idêntico ao jumper Atv\_A.

#### Entradas:

As entradas de controle da placa possuem os pinos IN1, IN2, IN3 e IN4, sendo que a entrada IN1 e IN2 são as entradas que controlam os bornes do motor A, e as entradas IN3 e IN4 controlam os bornes do Motor B. Para controlar os motores, acione os pinos da seguinte forma:

MOTOR A	IN1	IN2
Sentido direto	HIGH	LOW
Sentido reverso	LOW	HIGH
Freio	LOW	LOW
Freio	HIGH	HIGH

E no motor B, acione da mesma forma:

MOTOR B	IN3	IN4
Sentido direto	HIGH	LOW
Sentido reverso	LOW	HIGH
Freio	LOW	LOW
Freio	HIGH	HIGH

#### Motor A e Motor B:

Estes bornes são de fato onde os motores DC são conectados. A posição dos fios do motor na conexão com o borne altera o sentido de rotação do motor. Se for invertido os fios no borne, o sentido de rotação também é invertido no acionamento. Essa é uma boa dica para alterar a rotação dos motores, caso estejam girando em um sentido não desejado.

Perceba também, que como juntando o borne de Motor A e Motor B possuímos 4 conexões, podemos usar essa ponte para controlar 1 motor de passo. Se o motor for de

4 fios, podemos utilizar conectando direto. Se for de 5, ou 6, podemos conectar o tap central ao VLogic.

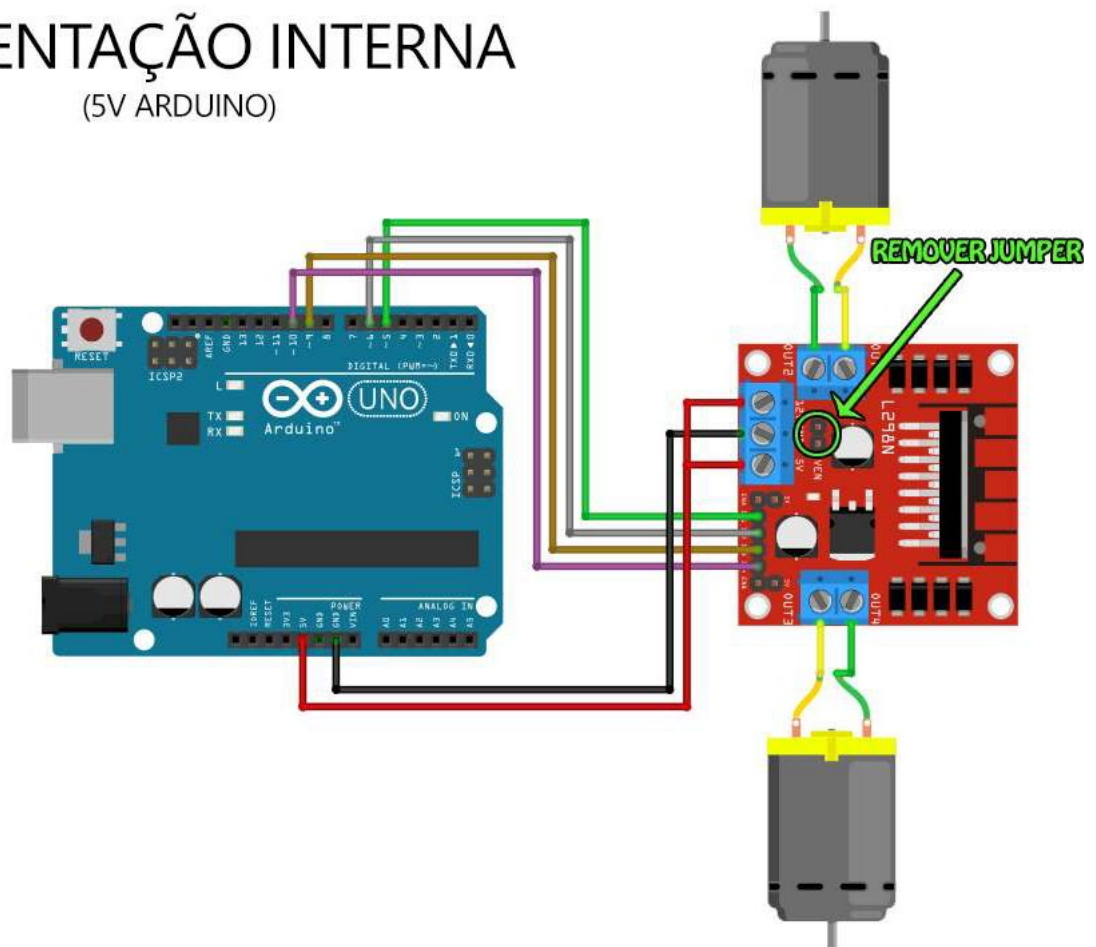
#### Montagem da Ponte H L298N com Arduino:

A montagem a ser feita dependerá muito da tensão disponível, por isso, aqui será abordado 4 tipos comuns de ligações que podem ser feitas:

##### 5V diretamente pelo USB

Esta conexão é destinada para ser feita usando a própria alimentação do cabo USB, para fins de testes ou alguma aplicação onde é importante testar o sistema constantemente e precisa fazer mudanças no código. É importante que a conexão seja feita **com o motor sem carga e consumindo baixa corrente**. Para isso, deve **remover o jumper do regulador de tensão** e conectar os 5V do Arduino tanto em Vcc quanto em VLogic. Fazendo a conexão abaixo:

## ALIMENTAÇÃO INTERNA (5V ARDUINO)

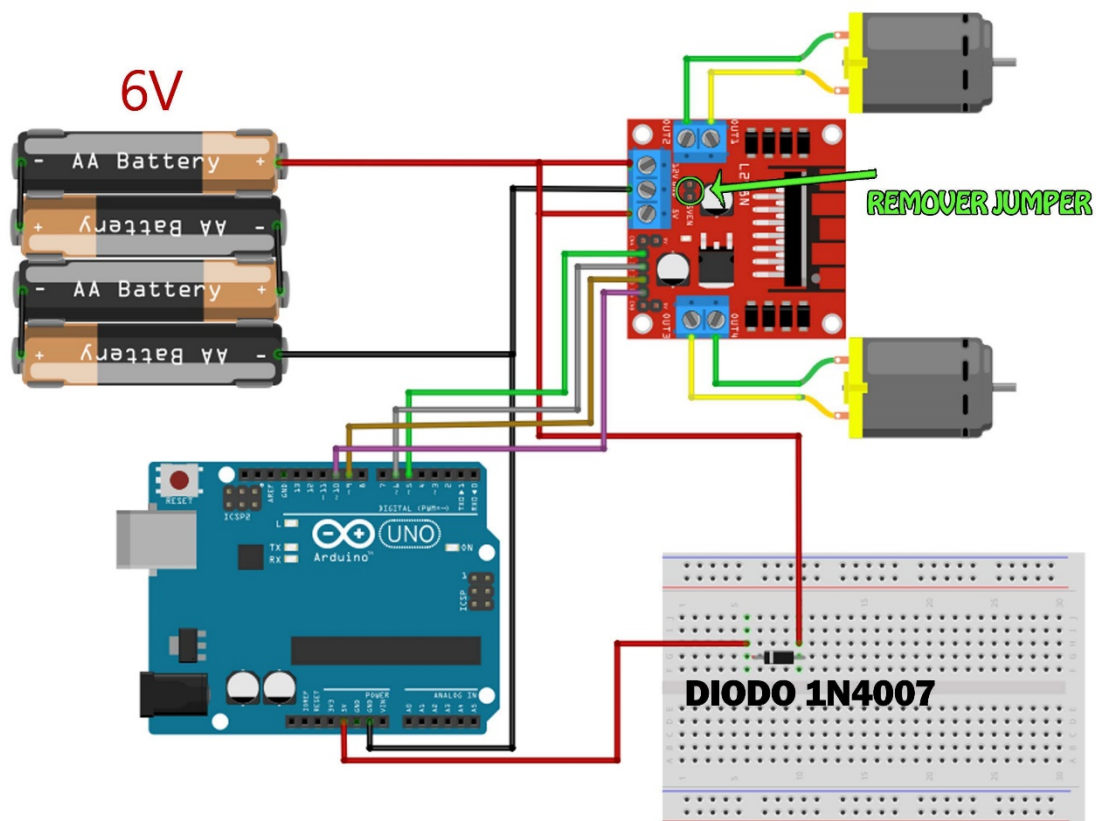




### 6V de alimentação externa

Com um suporte de 4 pilhas AA, e usando pilhas alcalinas na montagem, temos um total de 6V na alimentação da ponte H. A montagem é similar a anterior, mas com um detalhe: O fio do suporte de pilhas não pode ser ligado diretamente ao pino 5V do Arduino UNO, porque o microcontrolador Atmega 328P só suporta até 5,5V, então, para contornar isso, usaremos um diodo na montagem. Este diodo tem a função de provocar uma queda de tensão de aproximadamente 0,7V em cima dele, e com isso, a tensão no microcontrolador do Arduino seria de 5,3V, estando no intervalo de operação.

Para uma margem ainda maior de segurança, insira dois diodos em série.

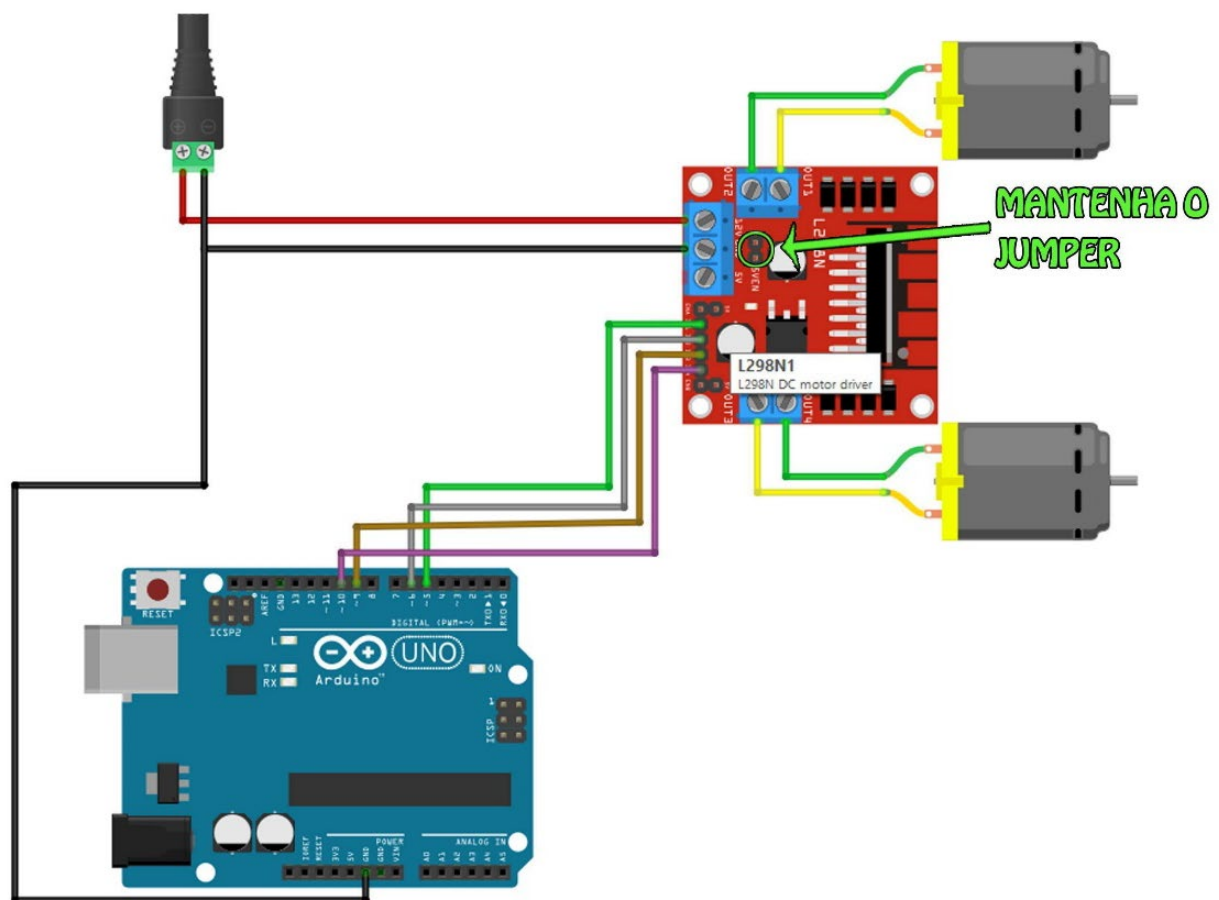


É importante fazer a observação que se caso for inserido na montagem componentes adicionais. Pode ser que seja necessário fazer uma alimentação separada para o Arduino para conseguir suprir a demanda de corrente do sistema.

### 7V – 20V de alimentação externa

Para uma tensão externa entre 7V e 20V, é recomendado fazer uma alimentação separada entre a ponte H e o Arduino, sendo que até 12V, você pode usar o borne VLogic para alimentar alguns circuitos que necessitam de 5V e demandam baixa corrente. Nesta montagem é necessário **manter o jumper do regulador de tensão**, já que o circuito VLogic será alimentado pela saída do regulador de tensão interno.

## Tensão externa (7V - 20V)



É possível, em até 12V, fazer uma alimentação única entre o Arduino e a ponte H, colocando a mesma tensão de 12V no pino de Vin do Arduino e no borne Vcc da ponte H.

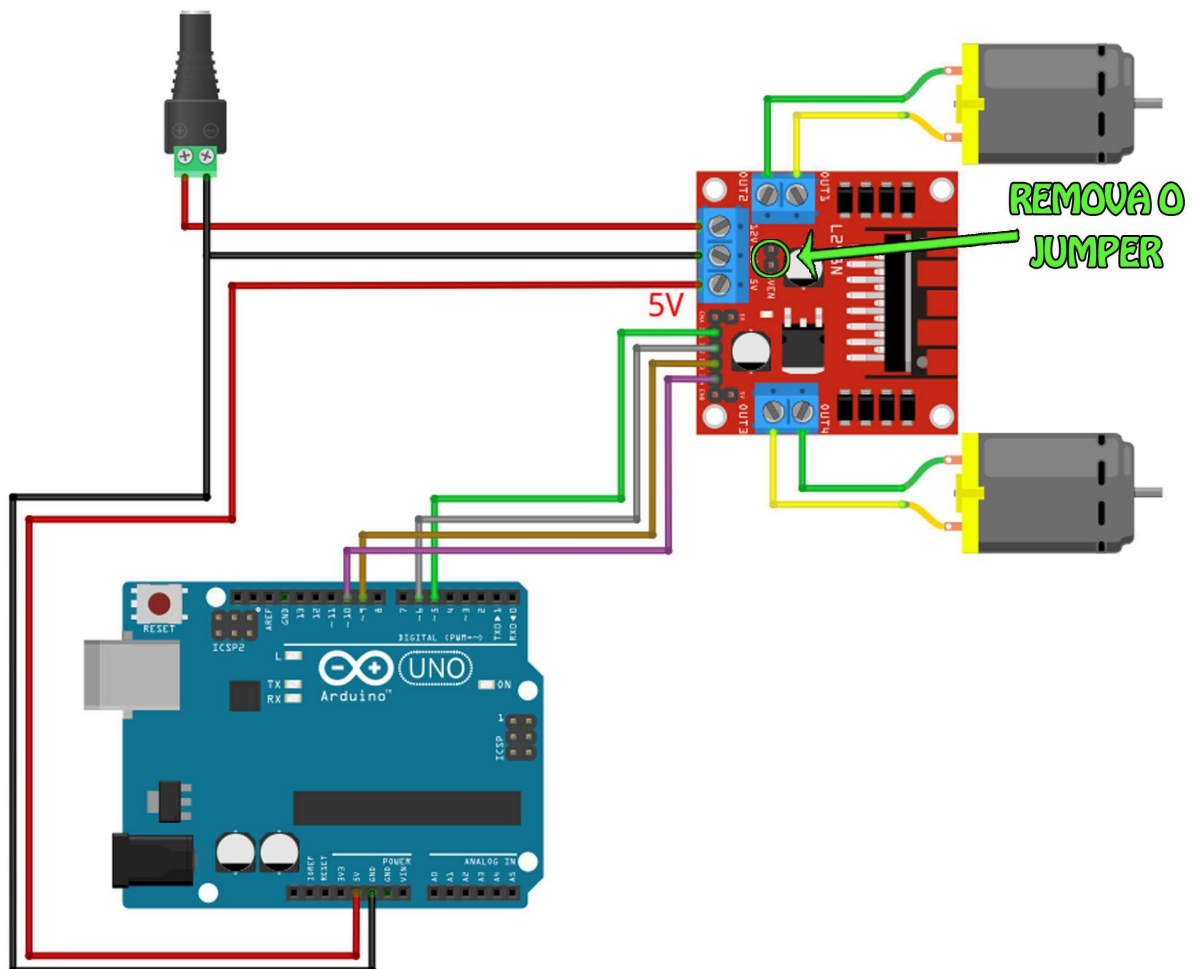


### 20V – 46V de alimentação externa

Para uma alimentação de 20V até 46V -que é o limite máximo - é extremamente **necessário remover o jumper do regulador de tensão**. Este procedimento é necessário porque este regulador só opera em uma margem segura abaixo de 20V, e acima disso ele poderá sobreaquecer e danificar (**principalmente acima de 35V**). Logo, é necessário remover o jumper e fazer a alimentação de Vlogic com o 5V do Arduino.

Lembrando que nesse caso a alimentação do Arduino e da ponte H é feita de maneira independente. O circuito é este:

### Tensão externa (20V - 46V)



## Código para controle da Ponte H:

```
// Exemplo 10 - Ponte H Dupla L298N
// Apostila Eletrogate - KIT Robotica

// Iremos fazer uma classe para facilitar o uso da ponte H L298N na manipulação dos motores na função
// Setup e Loop.

class DCMotor {
    int spd = 255, pin1, pin2;

public:
    void Pinout(int in1, int in2){ // Pinout é o método para a declaração dos pinos que vão controlar
o objeto motor
        pin1 = in1;
        pin2 = in2;
        pinMode(pin1, OUTPUT);
        pinMode(pin2, OUTPUT);
    }
    void Speed(int in1){ // Speed é o método que irá ser responsável por regular a velocidade
        spd = in1;
    }
    void Forward(){ // Forward é o método para fazer o motor girar para frente
        analogWrite(pin1, spd);
        digitalWrite(pin2, LOW);
    }
    void Backward(){ // Backward é o método para fazer o motor girar para trás
        digitalWrite(pin1, LOW);
        analogWrite(pin2, spd);
    }
    void Stop(){ // Stop é o método para fazer o motor ficar parado.
        digitalWrite(pin1, LOW);
        digitalWrite(pin2, LOW);
    }
};

DCMotor Motor1, Motor2; // Criação de dois objetos motores, já que usaremos dois motores, e eles já
estão prontos para receber os comandos já configurados acima.

void setup() {
    Motor1.Pinout(5,6); // Seleção dos pinos que cada motor usará, como descrito na classe.
    Motor2.Pinout(9,10);
}

void loop() {
    Motor1.Speed(200); // A velocidade do motor pode variar de 0 a 255, onde 255 é a velocidade máxima.
    Motor2.Speed(200);

    Motor1.Forward(); // Comando para o motor ir para frente
    Motor2.Forward();
    delay(1000);
    Motor1.Backward(); // Comando para o motor ir para trás
    Motor2.Backward();
    delay(1000);
    Motor1.Stop(); // Comando para o motor parar
    Motor2.Stop();
    delay(500);
}
```

## Exemplo 10 – Carro Automático com Ponte H e Arduino

O kit chassi é um conjunto de peças e componentes que permitem a montagem de um robô móvel, com 2 rodas que giram em sentidos independentes, fazendo assim o robô ter uma possibilidade de movimentos e direções muito grandes.

O chassi presente neste kit tem uma série de furos onde pode ser encaixado sensores, atuadores, e que te permite incluir diversos sensores presente no kit Arduino Robótica para realizar as mais variadas tarefas.

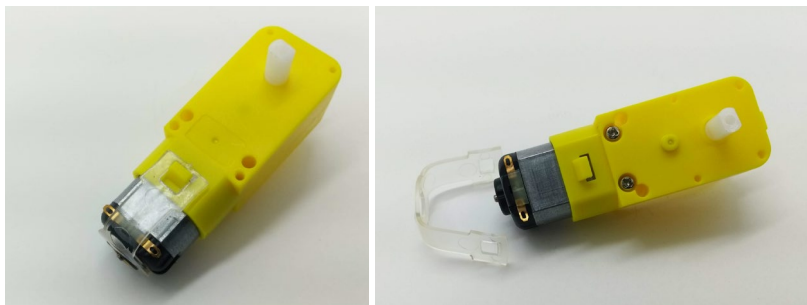
Esta montagem do carrinho também está disponível no nosso em <https://blog.eletrogate.com>, onde existem diversas outras montagens e experimentos.

### Lista de Materiais

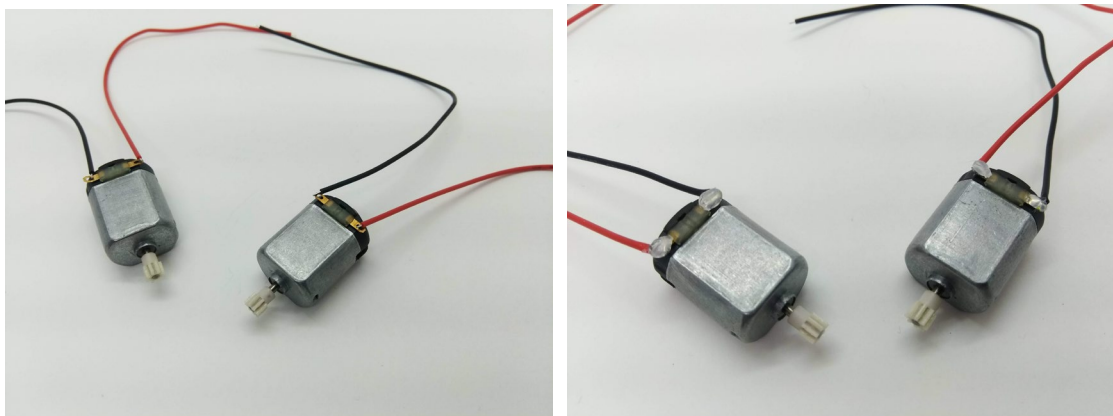
- Módulo Ponte H Dupla L298N
- Chassi 2WD em acrílico
- 2 Motores DC 3V – 6V com caixa de redução
- 2 rodas 68mm de borracha
- Roda boba (universal)
- Suporte para 4 pilhas AA
- Jogo de parafusos e acessórios
- Jumpers macho-fêmea
- Jumpers macho-macho
- Adaptador de bateria 9V
- Chave Gangorra On/Off
- 4 Pilhas Alcalinas AA
- Bateria Alcalina 9V
- Protoboard
- Arduino e cabo de conectar no computador



O primeiro passo é soldar os fios soltos aos terminais do motor (peça ajuda a um adulto, caso você seja criança). Para isso, vamos soltar o motor de sua caixa de redução, removendo a alça que une os dois



Após isso, retire-o e faça a ligação de fios aos terminais do motor com solda, e também é recomendado passar um pouco de cola quente em cima da solda para evitar algum curto ou contato indesejado.

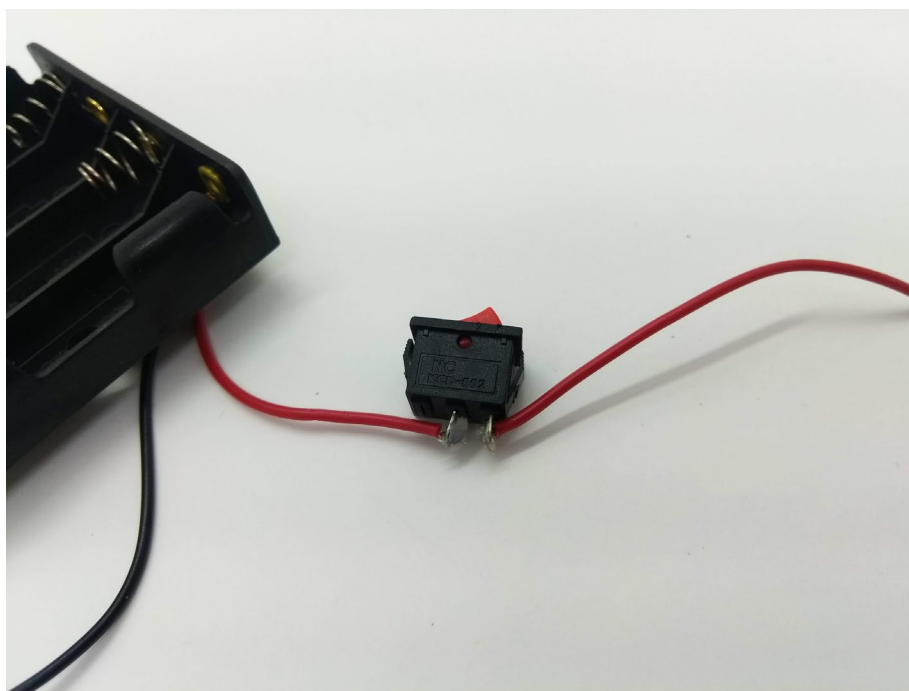


Faça a solda para os dois motores. E depois, volte com os motores para as suas respectivas caixas de redução (as caixas amarelas) e coloque a alça transparente, e assim, os motores já estão prontos para serem usados.



Com o ferro de solda ainda ligado, solde a chave gangorra no suporte de pilhas, para podermos desligar o carrinho e economizar energia. O procedimento será bem simples, basta colocar a chave on/off no final do fio vermelho do suporte e soldar em um dos dois terminais, e pegar um outro fio disponível (pode ser um jumper macho/macho que veio junto ao kit) e soldar no outro fio. Se quiser também, uma possibilidade é cortar o fio vermelho ao meio, descascar só um pouco cada ponta, e fazer como nos passos anteriores; a desvantagem fica sendo a limitação será o limite para posicionar a chave para desligar na montagem do carrinho.





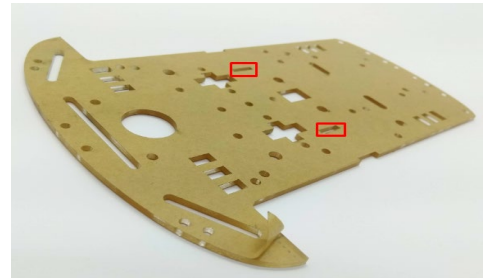
Com tudo soldado, volte a montagem da estrutura do carrinho. Observe que a peça maior do carrinho parece de madeira. Se trata de uma película protetora, pois a case é de acrílico transparente. Se quiser deixar o seu carrinho mais estético, remova a película marrom de ambos os lados, só puxando, como na imagem abaixo.



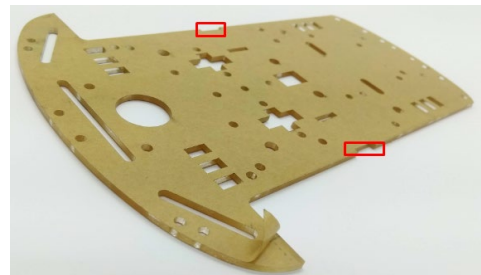


Com as peças da imagem acima em mãos, pegue os motores e o conjunto de parafusos, e faça o seguinte:

1° - Encaixe duas peças menores dessas em formato de T, nas fendas internas.

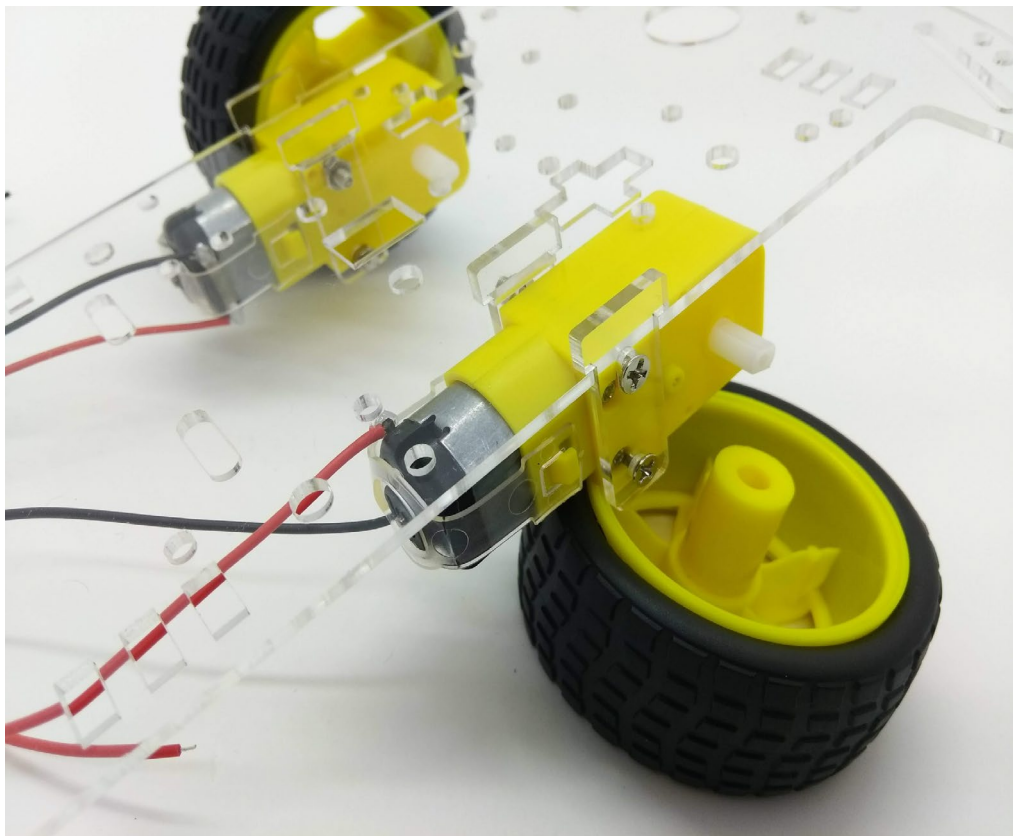


2° - Coloque o motor com seus furos laterais alinhados aos furos do suporte menor, que foi encaixado anteriormente

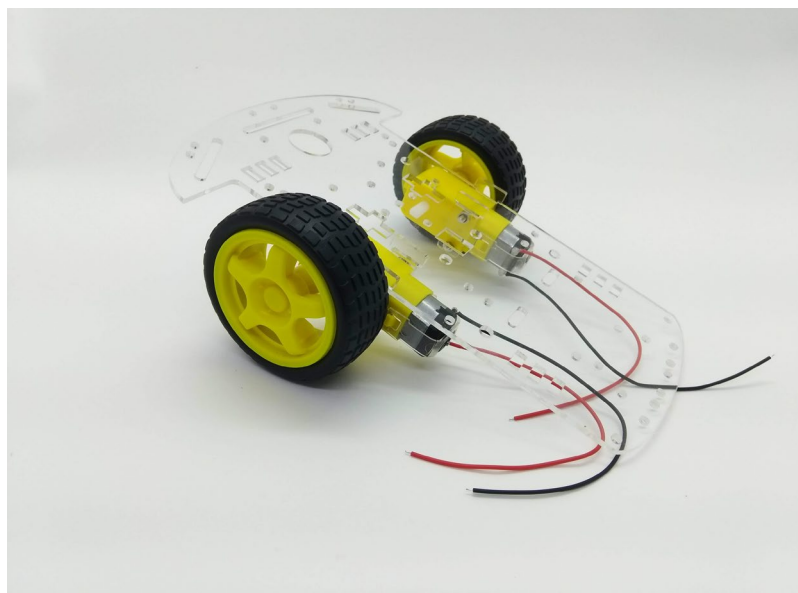


3° - Encaixe outros dois suportes pequenos em formato de T com os furos alinhados aos furos do motor também e parafuse usando porcas.

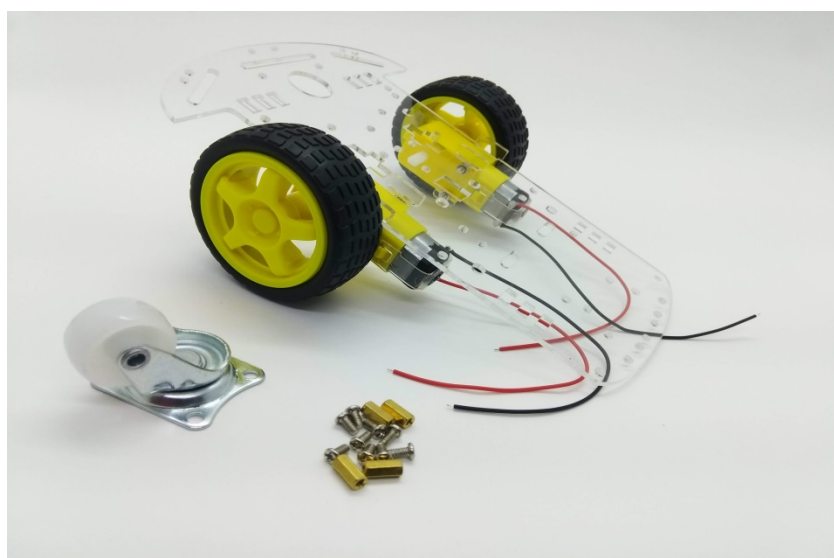
É interessante observar que devemos colocar o eixo do motor virado para a extremidade do chassi onde tem um círculo vazado no acrílico. Fazendo para os dois lados, e encaixando a roda no eixo do motor, o resultado obtido deverá ser este:



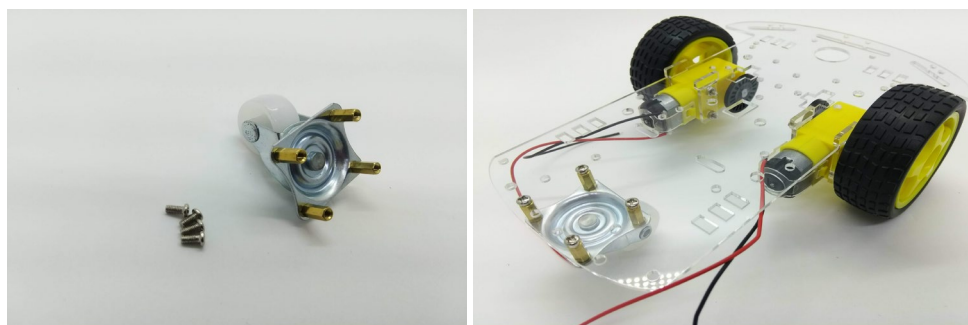
E encaixando a roda no eixo do motor, fica assim:



O próximo passo será a instalação da roda boba, e para isso precisaremos das seguintes peças da imagem abaixo:



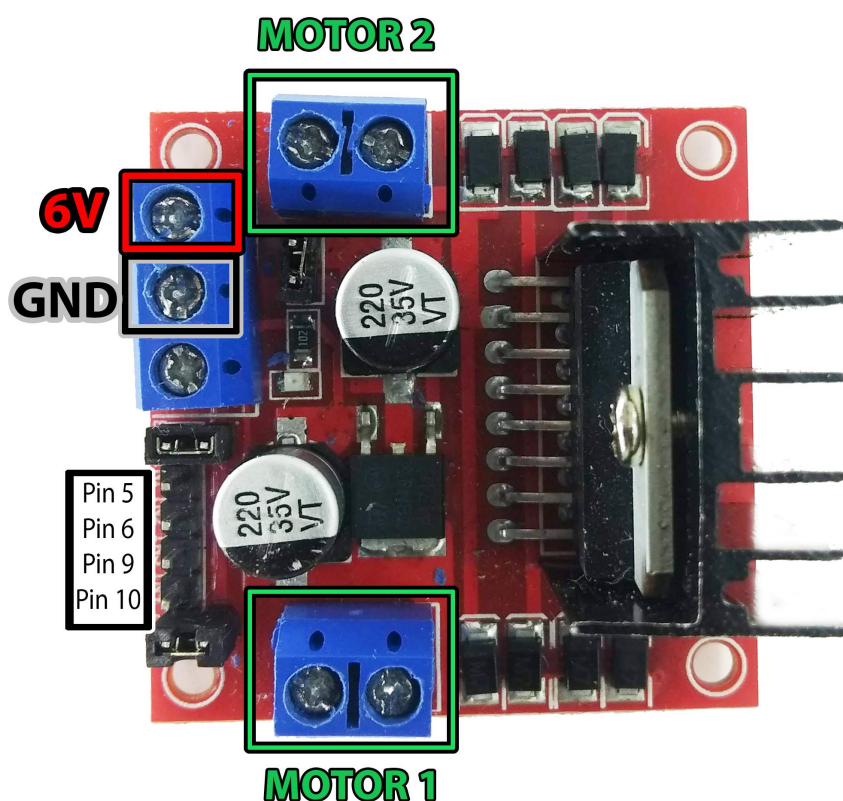
Começaremos parafusando o extensor, que é essa peça dourada no kit de parafusos à roda boba, e depois parafusando o extensor ao chassi do carrinho da seguinte forma:



E assim, está concluído a montagem do esqueleto básico do carrinho, e agora iremos montar a parte para controlar ele.

Para fazer o acionamento dos motores, precisaremos de algo que consiga fornecer energia o suficiente para isso, e que possa ser controlado. Para isso iremos usar um driver ponte H L298N. A vantagem de usar este driver é permitir a rotação nos dois sentidos, como visto no exemplo anterior, e também permitir que uma corrente bem maior que a do que o Arduino suporta passe pelo motor.

O módulo ponte H usado é desta forma:

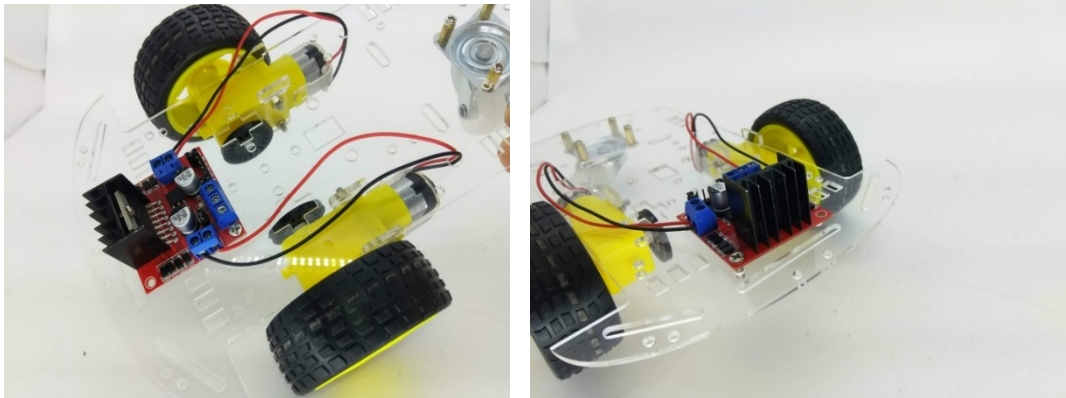


Onde iremos, inicialmente, fazer as conexões dos motores 1 e 2. Inicialmente não será necessário preocupar a posição dos fios do motor no borne. Para instalar o motor, basta



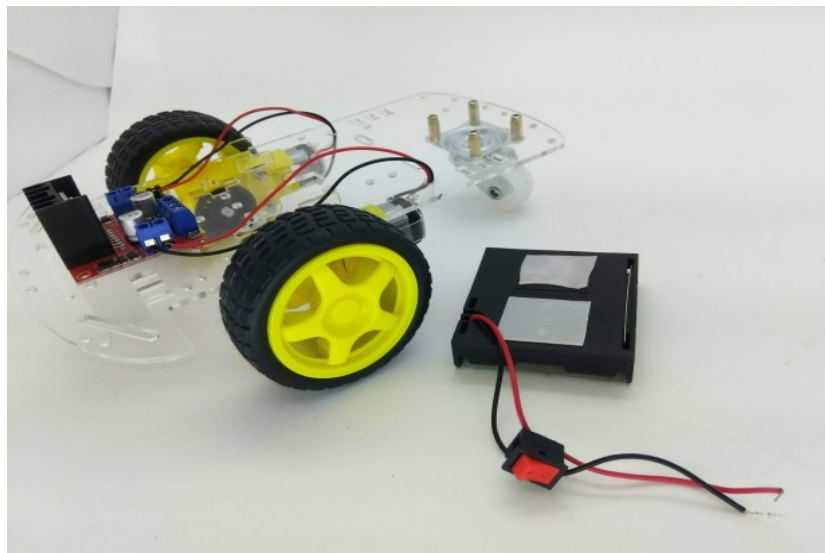
afrouxar com uma chave de fenda ou Philips o borne, encaixar o fio do motor e apertar. Faça para os dois motores.

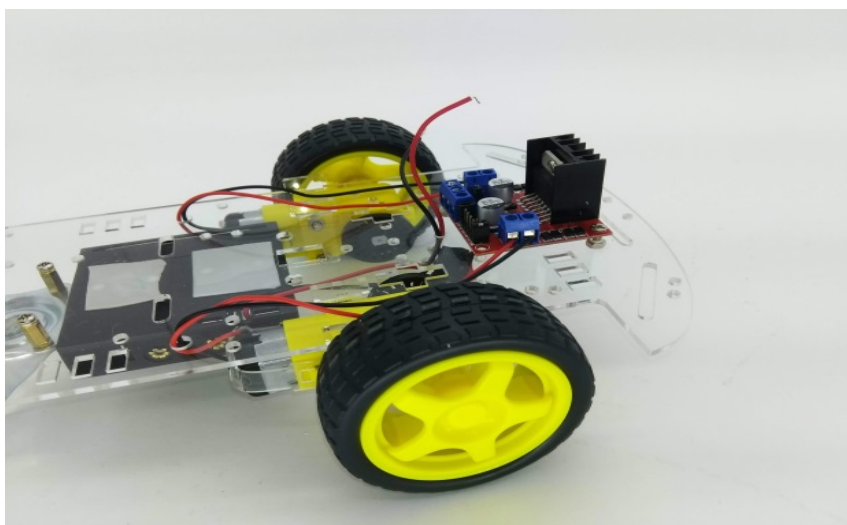
A montagem deve estar da seguinte forma:



Perceba pela foto acima que há um parafuso no módulo ponte H. Esse parafuso está ligado com uma arruela ao chassi. Ela foi adicionada porque isso fará com que possamos dispensar o uso de colas e o deixar bem fixo, mas se quiser, pode usar fita dupla face para prender a ponte H ao chassi. Com a ponte H instalada e os motores conectados, faremos a instalação do suporte de pilhas.

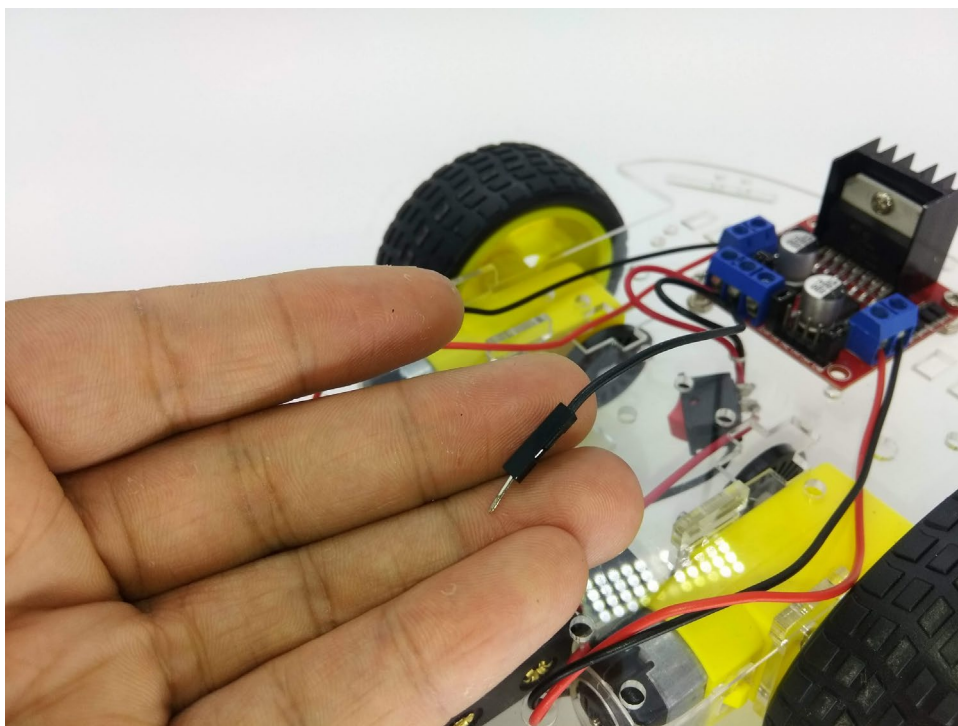
Coloque uma dupla face nas costas do suporte e então ele foi fixado na parte de baixo do chassi, da seguinte forma:





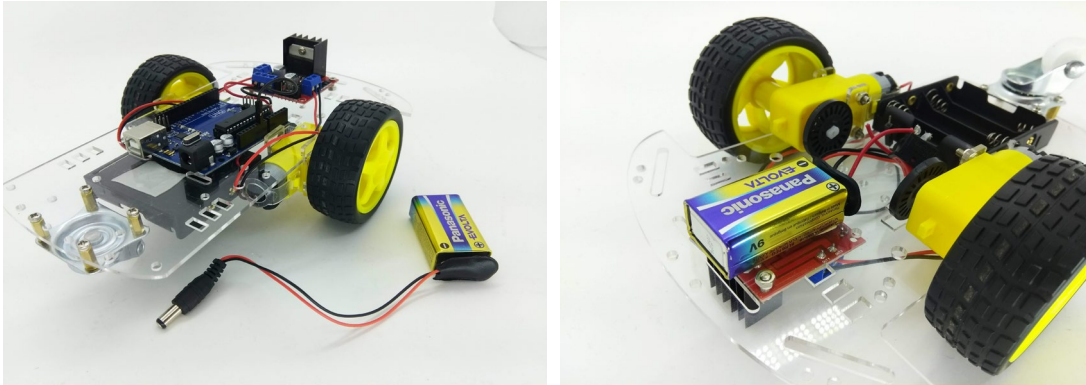
Conectando os fios de alimentação na ponte, Coloque o fio vermelho, que sai do suporte de pilhas, no borne 6V do módulo ponte H, e o preto do suporte de pilhas no borne de GND. Para consultar essas informações, basta ver a imagem da ponte H um pouco mais acima.

Coloque também um jumper macho/macho dentro do borne de GND junto com o fio preto do suporte de pilhas, porque mais tarde ele será usado. A ligação ficou assim:

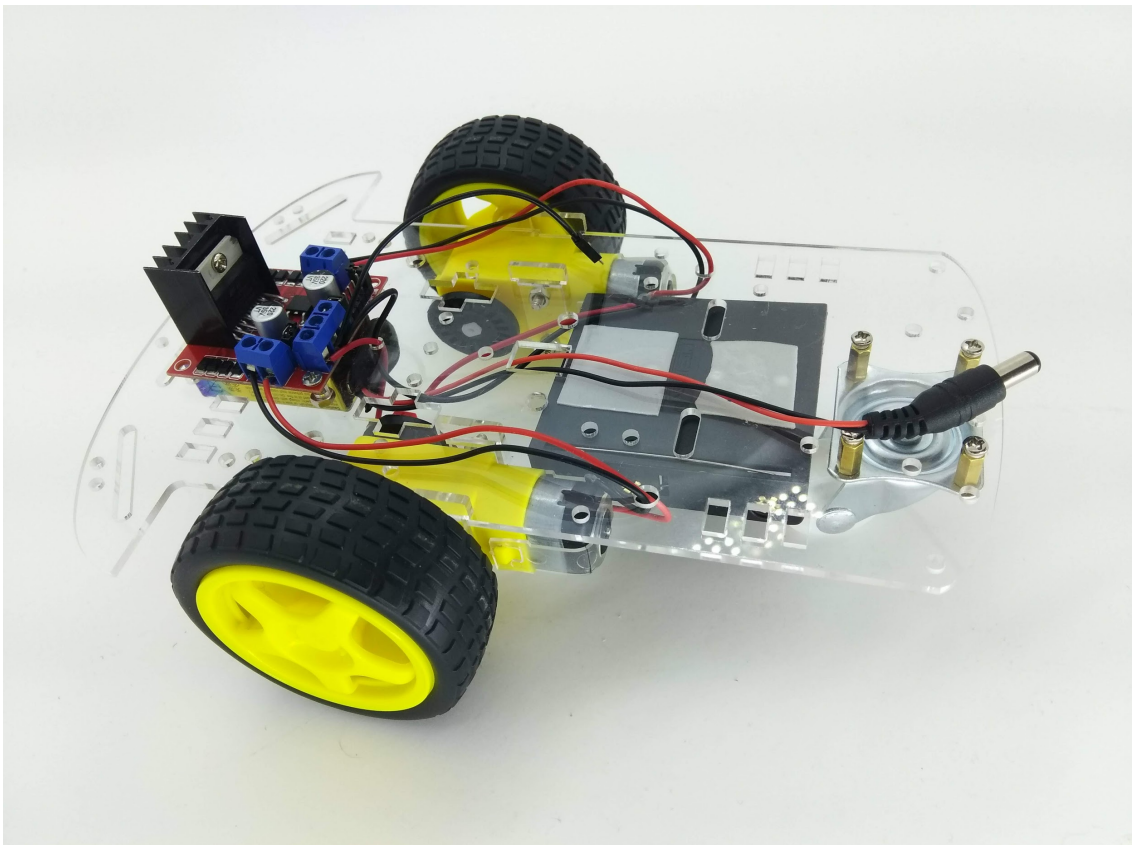


O jumper colocado é para ser usado sendo colocado na porta GND do Arduino. Agora só está faltando uma bateria de 9V, para fornecer energia ao Arduino. Essa alimentação separada foi escolhida porque as 4 pilhas possuem uma disponibilidade de energia um pouco baixa, e separando as alimentações fará que a durabilidade energética do robô aumente.

Coloque uma fita dupla face na bateria de 9V e instale-a exatamente embaixo (no chassi) do módulo ponte H. O resultado previsto é este:

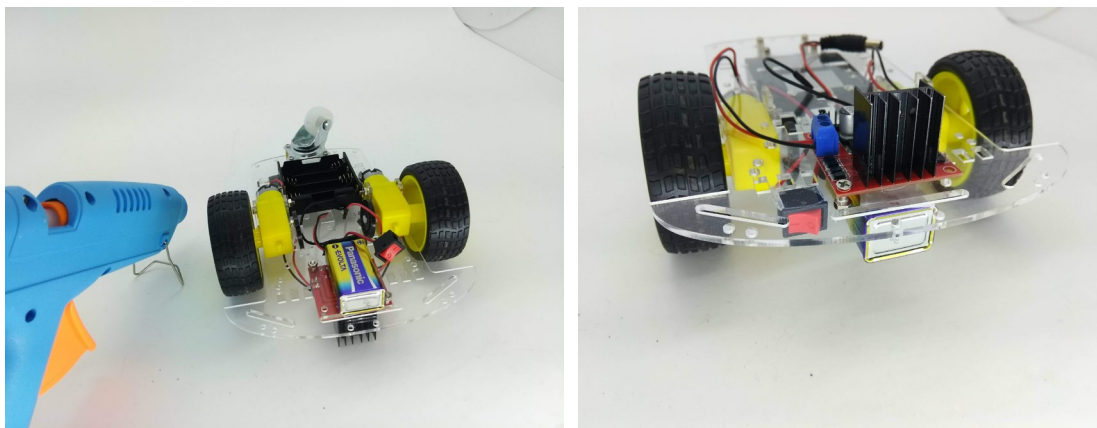


A imagem abaixo demonstra em como deve estar a montagem do robô, se caso não estiver, revise os outros passos.

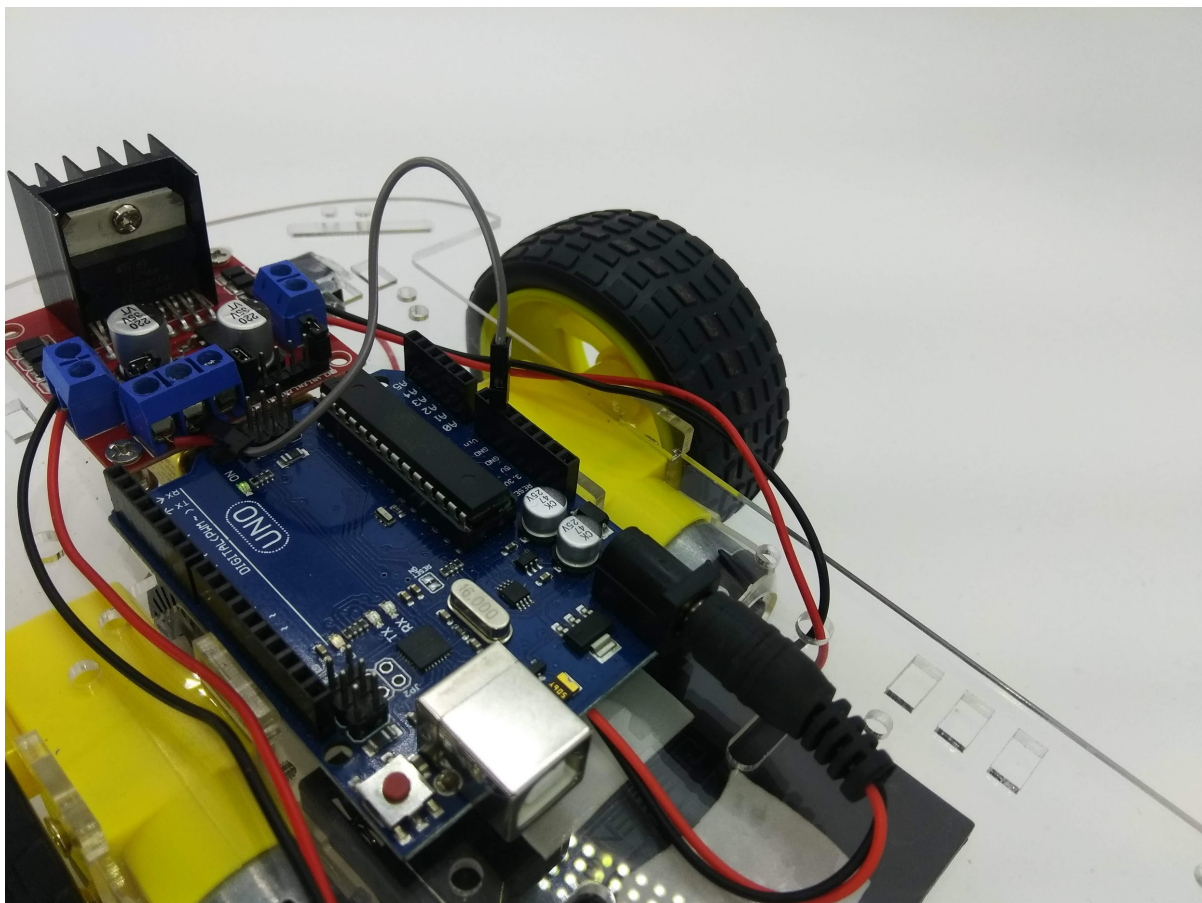




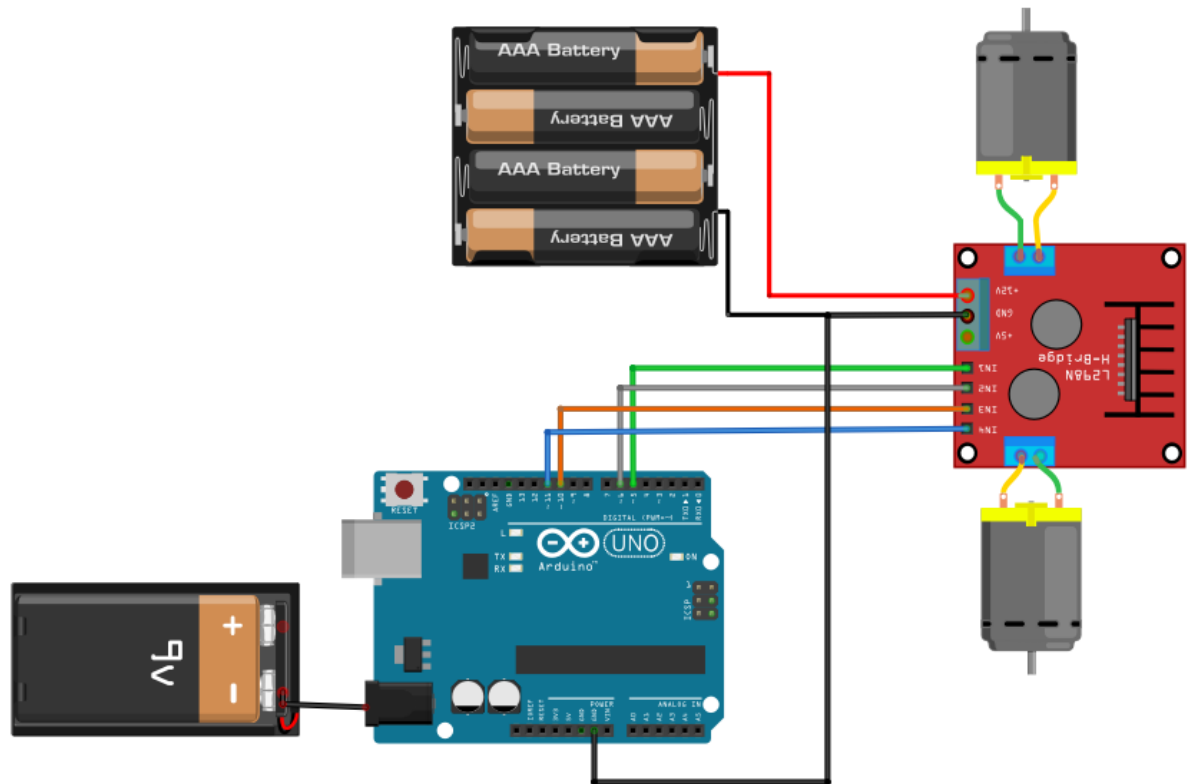
Para melhorar um pouco a estética e facilitar o uso, a chave on/off foi instalada bem na frente do carrinho, e colada com cola quente. Para isso foi preciso alongar um pouco mais o fio.



Coloque um pouco de fita dupla face na parte inferior da placa Arduino e encaixe ela na parte superior da placa, da seguinte forma:



Colocando a placa na parte central. As conexões da ponte H com o Arduino deverão ser feitas da seguinte forma:



Observações:

- É extremamente importante que os GND estejam ligados juntos, isso significa que temos que ligar o GND da ponte H com o GND do Arduino. (Estamos fazendo isso com o jumper que foi colocado junto com o GND da bateria no borne da ponte H. Esse GND foi ligado na porta GND do Arduino).
- Remova a alimentação da bateria de 9V ao colocar o cabo USB,.

Código para teste do robô:

```
// Exemplo 11 - Chassi 2 rodas
// Apostila Eletrogate - KIT Robotica

// Iremos fazer uma classe para facilitar o uso da ponte H L298N na manipulação dos motores na função
// Setup e Loop.

class DCMotor {
    int spd = 255, pin1, pin2;

public:

    void Pinout(int in1, int in2){ // Pinout é o método para a declaração dos pinos que vão controlar
o objeto motor
        pin1 = in1;
        pin2 = in2;
        pinMode(pin1, OUTPUT);
        pinMode(pin2, OUTPUT);
    }
    void Speed(int in1){ // Speed é o método que irá ser responsável por regular a velocidade
        spd = in1;
    }
    void Forward(){ // Forward é o método para fazer o motor girar para frente
        analogWrite(pin1, spd);
        digitalWrite(pin2, LOW);
    }
    void Backward(){ // Backward é o método para fazer o motor girar para trás
        digitalWrite(pin1, LOW);
        analogWrite(pin2, spd);
    }
    void Stop(){ // Stop é o método para fazer o motor ficar parado.
        digitalWrite(pin1, LOW);
        digitalWrite(pin2, LOW);
    }
};

DCMotor Motor1, Motor2; // Criação de dois objetos motores, já que usaremos dois motores, e eles já
estão prontos para receber os comandos já configurados acima.

void setup() {
    Motor1.Pinout(5,6); // Seleção dos pinos que cada motor usará, como descrito na classe.
    Motor2.Pinout(10,11);
}

void loop() {
    Motor1.Speed(200); // A velocidade do motor pode variar de 0 a 255, onde 255 é a velocidade máxima.
    Motor2.Speed(200);

    Motor1.Forward(); // Comando para o robô ir para frente
    Motor2.Forward();
    delay(1000);
    Motor1.Backward(); // Comando para o robô ir para trás
    Motor2.Backward();
    delay(1000);
    Motor1.Forward(); // Comando para o robô girar
    Motor2.Backward();
    delay(1000);
    Motor1.Stop(); // Comando para o robô parar
    Motor2.Stop();
    delay(500);
}
```

E com isso o robô deve ir para frente, para trás girar e parar, e repetir este ciclo. Altere o código para gerar mais funções.