

[HANDS ON] ARDUINO + DC MOTOR DRIVER + POTÊNCIOMETRO



Ooooiiii GALERA!

Aqui está mais um novo **Hands On**!

Faz tempo que tento fazer este tutorial, na qual eu acho muito legal e útil de fazer. Vamos fazer mexer no motor DC!!!

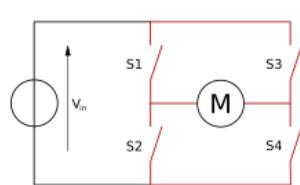
Para realizar este tutorial vamos usar um **DC Motor Driver**.

Mas antes de começar a falar sobre o **DC Motor Driver**, vou começar a explicar um pouco sobre a famosa **Ponte H** e o **PWM**.

De acordo com o *Wikipedia*, a **Ponte H** é um circuito usado para **determinar o sentido da corrente (rotação do motor)** e também o **valor de tensão no controle do motor DC**.

Parando para explicar um pouco a figura abaixo:

A **Ponte H** pode ser resumida em 4 chaves (S1, S2, S3 e S4). O fechamento dos pares de chaves faz mudar o sentido da rotação do motor.



Ponte H

Caso **fechamos as chaves S1 e S4** (lado esquerdo da imagem), vamos ter um circuito fechado e portanto, o motor gira em um sentido. Se **fechamos o S3 e o S2** (lado direito da imagem), invertemos a **polaridade**.

POPULARES

[Hands On] – Arduino + LCD + DS3231 Real Time Clock (Relógio) 8,755 visualizações

[Hands On] – Arduino + LCD + Leitor RFID 7,144 visualizações

Quebra de patentes: a situação do inventor da BINA 7,093 visualizações

[Hands On] – Arduino + LCD + Sensor de distância (HC-SR04) 4,641 visualizações

Business Model Canvas: já ouviu falar? 3,744 visualizações

A Arte de Fazer 3,562 visualizações
Perguntas Criativas e Desafiadoras

NEWSLETTER DO i9

[Assine a newsletter do i9](#)

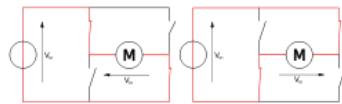
ACESSOS

104930
Ver Estatísticas

MÍDIAS SOCIAIS

Seguir

do motor e assim mudamos o sentido do motor.

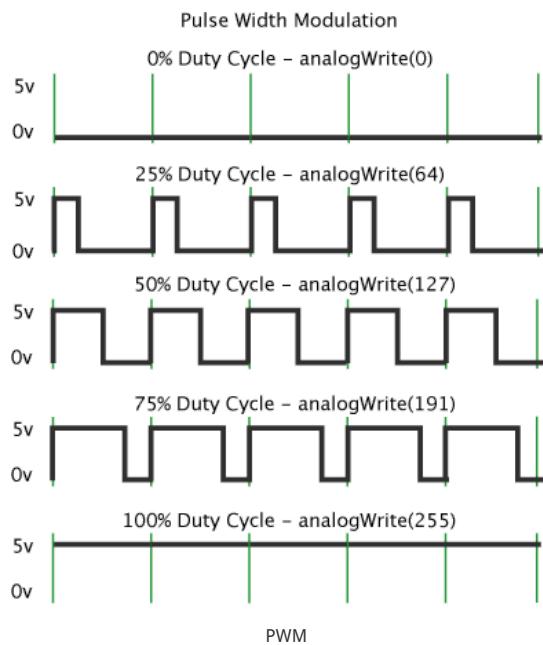


Esquema de Funcionamento

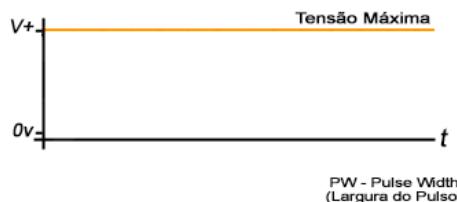
Ao fechar as chaves S1 e S3 ou S2 e S4, causamos um curto nos terminais do motor, freando o motor.

Agora que vimos um resumo rápido sobre **Ponte H**, vou explicar o **PWM**.

O **PWM** (*Pulse-Width Modulation*) que traduzindo fica **Modulação por Largura de Pulso**, é o controle de potência ou velocidade, através do pulso de uma onda quadrada.



"Imagine uma chave simples liga e desliga, quando ligada 100% da tensão e da potência é aplicada a carga, já quando a chave está aberta a tensão é nula e assim a potência é 0. Quando controlamos o tempo que a chave fica ligada e consequentemente o tempo dela desligada podemos controlar a potência média entregue a carga, por exemplo: a chave fica ligada 50% ligada e 50% desligada, isso quer dizer que em média temos 50% do tempo com corrente e 50% sem. Portanto a potência média aplicada na carga é a própria tensão média, ou seja, 50%, portanto quanto maior o tempo que o pulso se manter em nível lógico alto, ou seja, ligado maior a potência entregue a carga, quanto menor o tempo em nível lógico alto menor a entrega de potência." (MecaWeb)



Animação PWM - by Mecaweb

Vou falar agora sobre os DC Motor Drivers.

Podemos usar **Ponte H**, com **controle de velocidade** ([L293](#) ou [L298N](#)) ou **somente acionamento** (**Relés**). No caso deste tutorial, vamos usar um **Shield** que usa uma **Ponte H** com o [L298N](#).

O **Shield** com o [L298N](#) é possível encontrar em diversos lugares, e a minha sugestão é *comprar invés de fazer, pois as minhas Pontes H sempre ficam uma merda*.

FilipeFlop

DealExtreme (este é tem 4 saídas para Motor DC e 2 para motores de passo)

MercadoLivre



DC Motor Drive

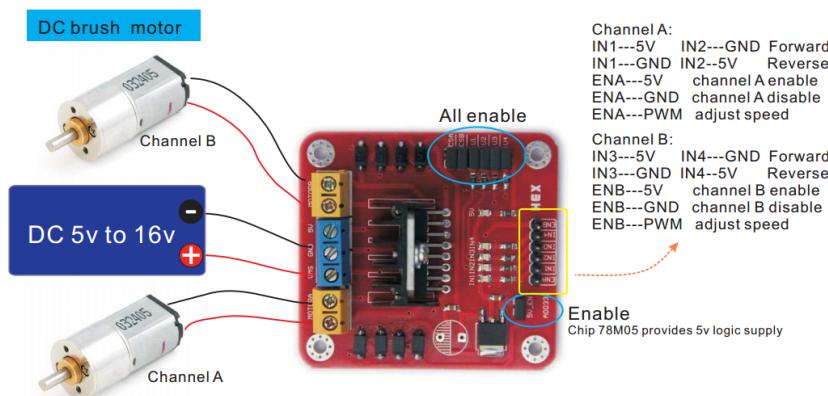
Bora botar a mão na massa!



Hands On do DC Motor Driver

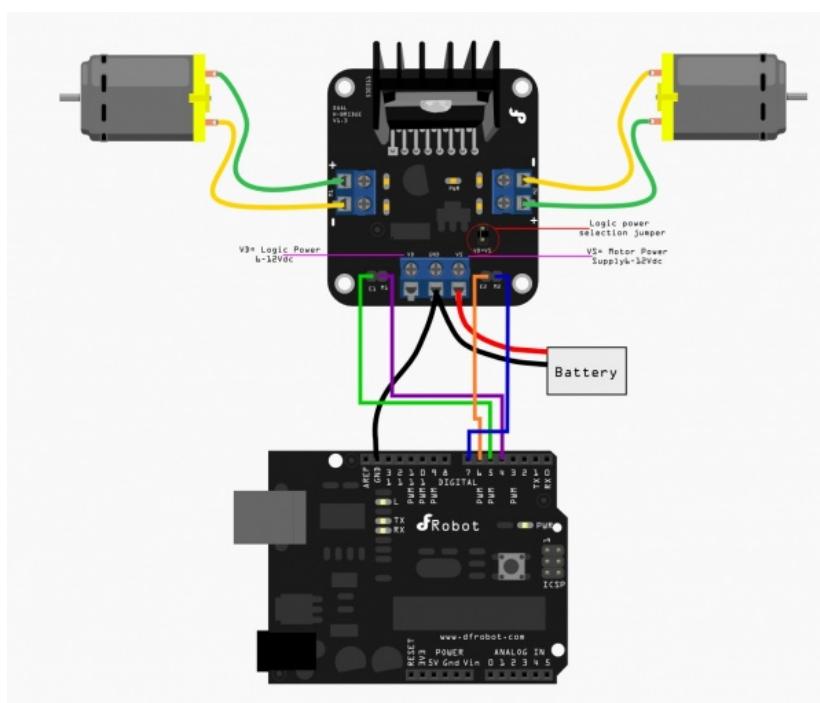
Antes de fazer para a placa que irei usar, vou tentar mostrar no [L298 Dual Bridge Motor Driver](#)

L298 Dual H-Bridge Motor Driver User's Guide



Como conectar as coisas no L298 Dual H-Bridge Motor Driver

Alguns [L298 Dual Bridge Motor Driver](#) não vem com o pino de ENB (Enable), como no Shield seguinte:



L298 sem Enable – DFRobots

Caso tenha um desses, basta seguir a mesma ideia deste [tutorial da DFRobots](#): 2A Dual Motor Controller.

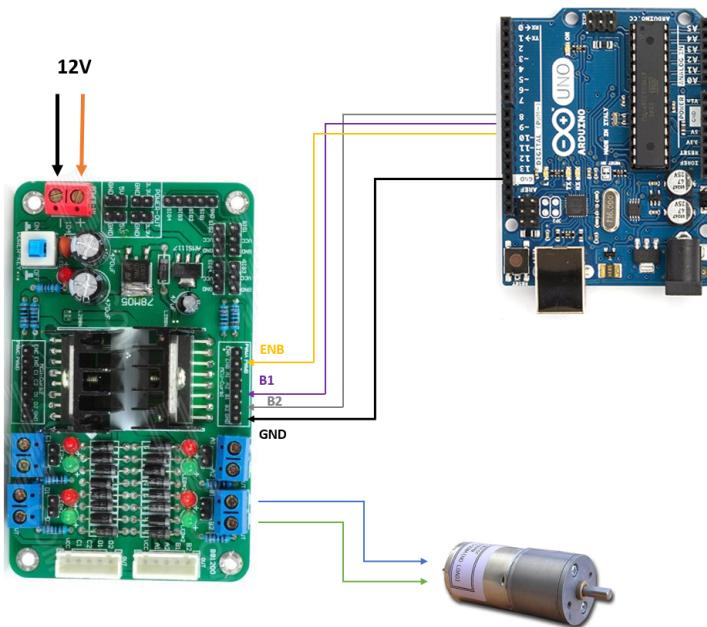
Para este **Hands On**, vou usar um **L298** com **4 saídas para Motores DC e 2 saídas para Motor de Passo**.



Motor Driver usado neste Hands On

Para fazer o motor funcionar, você vai precisar de um **fonte externa, pode ser bateria Li-Po, Pilha 9V, Fonte Variável ou Fixa**. **Cuidado: verifique a tensão de entrada do seu motor, caso coloque mais que ele suporte, você pode acabar danificando o seu motor.**

Caso tenha o mesmo **Motor Driver**, basta ligar da seguinte maneira:



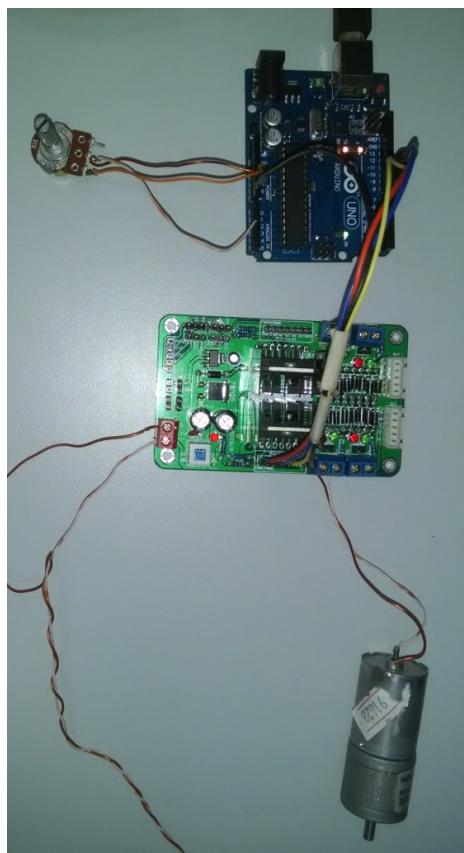
Forma de conectar o Motor Driver no Arduino UNO

Os pinos do **Arduino** são:

- **Arduino UNO -> Motor Driver**
- Digital 8 -> B2
- Digital 9 -> B1
- Digital 10 -> ENB
- GND -> GND

Também adicionei um potêmetro para poder variar o **PWM**.

- Os pinos do potêmetro são:
- **Arduino UNO -> Potômetro**
- Direita -> A2
- Esquerda -> 5V
- Meio -> GND



Controlando um PWM

Nesta primeira parte vou fazer funcionar **somente 1 motor**.

Vamos para o código:

```
// Motor B
int enB = 10; //enable motor B
int inB1 = 9; //input 1 do motor B
int inB2 = 8; //input 2 do motor B
const int PotPin = A2; // Pino analog aonde está o Potômetro

int Pot = 0; // valor do potômetro
int PWM_B = 0; // valor do PWM

void setup()
{
// Configurar todas os pinos para OUTPUT
pinMode(enB, OUTPUT);
pinMode(inB1, OUTPUT);
pinMode(inB2, OUTPUT);

serial.begin(9600);
}
void loop()
{
// Faz a leitura do potômetro
Pot = analogRead(PotPin);
// Faz o map para deixar o valor de 0 até 1023 em 0 até 255
PWM_B = map(Pot, 0, 1023, 0, 255);

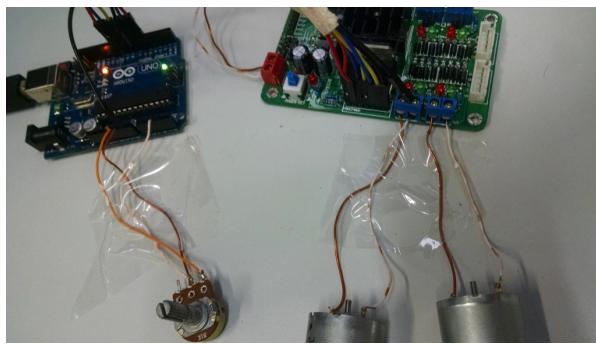
//Configura o input 1 e 2 do motor A para Antihorario
digitalWrite(inB1, LOW);
digitalWrite(inB2, HIGH);

//Escreve no PWM o valor do potômetro
analogWrite(enB, PWM_B);

Serial.print("PWM = ");
Serial.println(PWM_B);
}
```

Na segunda parte, adicionei **mais um motor DC** e com o potômetro, **vario de 0 a 255 o motor A e o motor B, sendo o inverso do PWM A**.

Para adicionar o outro Motor, basta coloca-lo no A1, A2 e no ENA, usando o mesmo GND. (Você pode usar o C e o D caso queira)



Controlando dois motores

O código da segunda parte:

```
// Motor A
int enA = 10; //enable motor A
int inA1 = 9; //input 1 do motor A
int inA2 = 8; //input 2 do motor A

// Motor B
int enB = 5; //enable motor B
int inB1 = 6; //input 1 do motor B
int inB2 = 7; //input 2 do motor B

const int PotPin = A2; // Pino analog aonde está o Potômetro

int Pot = 0; // valor do potômetro
int PWM_A = 0; // valor do PWM do motor A
int PWM_B = 0; // valor do PWM do motor B

void setup()
{
    // Configurar todas os pinos para OUTPUT
    //Motor A
    pinMode(enA, OUTPUT);
    pinMode(inA1, OUTPUT);
    pinMode(inA2, OUTPUT);

    //Motor B
    pinMode(enB, OUTPUT);
    pinMode(inB1, OUTPUT);
    pinMode(inB2, OUTPUT);

    Serial.begin(9600);
}

void loop()
{
    // Faz a leitura do potômetro
    Pot = analogRead(PotPin);
    // Faz o map para deixar o valor de 0 até 1023 em 0 até 255
    PWM_A = map(Pot, 0, 1023, 0, 255);
    // Faz o map para deixar o valor de 0 até 1023 em 255 até 0 (contrário do PWM_A)
    //PWM_B = 100;
    PWM_B = map(Pot, 0, 1023, 255, 0);

    //Configura o input 1 e 2 do motor A para Antihorario
    digitalWrite(inA1, LOW);
    digitalWrite(inA2, HIGH);

    //Configura o input 1 e 2 do motor B para Horario
    digitalWrite(inB1, HIGH);
    digitalWrite(inB2, LOW);

    //Escreve no PWM o valor do potômetro
    //Motor A
    analogWrite(enA, PWM_A);

    //Motor B
    analogWrite(enB, PWM_B);

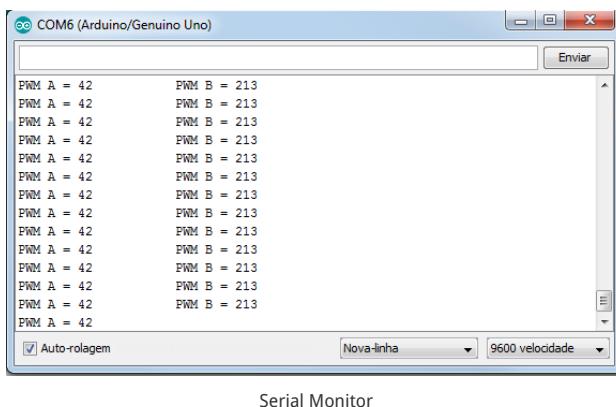
    Serial.print("PWM A = ");
    Serial.print(PWM_A);
}
```

```

  serial.print("      ");
  serial.print("PWM B = ");
  Serial.println(PWM_B);
}

```

A saída no **Serial Monitor**, deverá ficar desta maneira:



Aqui esta o super **Hands On** sobre **PWM**.

Espero que tenham gostado.

Aguardem pois esta vindo mais tipos de sensores e shields.

COMPARTILHE!

Like 23

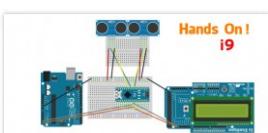
Sobre Gabriel Nakata



Estudante de Engenharia Elétrica da UDESC-CCT. Fascinado por novas tecnologias, em programação e jogos.



Artigos Relacionados



[Hands On] Sensores de Distância + Estação Metereológica + LCD + Comunicação I2C (Arduino Mega, UNO e Nano)

20/09/2016



[Hands On] Estação Meteorológica: Arduino + DHT11 + Sensor de Chuva

19/09/2016



[Hands On] – Arduino UNO + Arduino Mega – Protocolo I2C

02/05/2016

2 Comentários

Andre Orsato
03/08/2016 em 11:12 am

Gabriel, é possível adicionar mais um potenciômetro e controlar a velocidade de cada um dos motores individualmente? Obrigado



Gabriel Nakata
18/09/2016 em 9:18 pm

Olá André
É possível sim

```
// Faz a leitura do primeiro potênciometro
Pot_1 = analogRead(PotPin_1);
// Faz o map para deixar o valor de 0 até 1023 em 0 até 255
PWM_B = map(Pot_1, 0, 1023, 0, 255);

// Faz a leitura do segundo potênciometro
Pot_2 = analogRead(PotPin_2);
// Faz o map para deixar o valor de 0 até 1023 em 0 até 255
PWM_A = map(Pot_2, 0, 1023, 0, 255);
```

DEIXE UMA RESPOSTA

O seu endereço de email não será publicado.

Nome

E-Mail

Website

[Publicar comentário](#)

