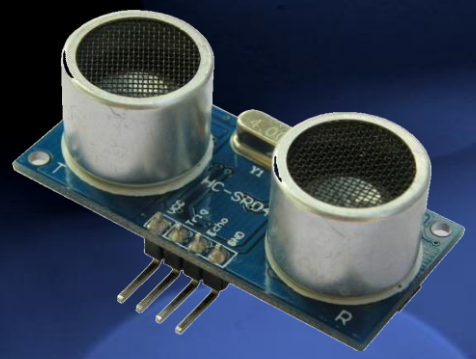
PMR 3100 – Introdução à Engenharia Mecatrônica

Módulo 05 – Meu Primeiro Robô

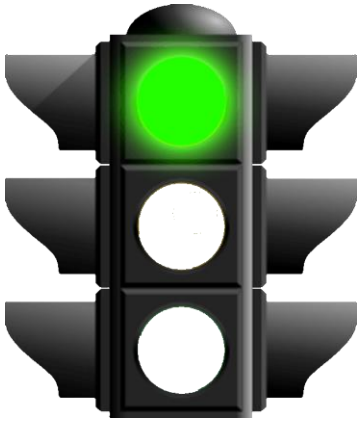
Aula 21 –Sensores e atuadores (PMW, linha, distancia) no arduino

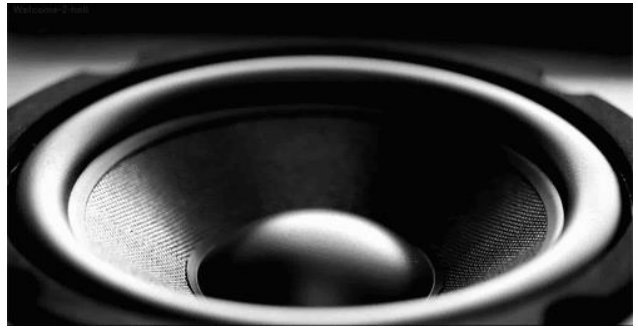


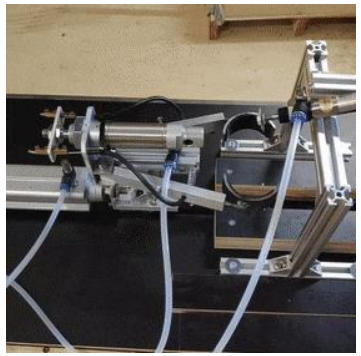
Prof. Dr. Rafael Traldi Moura

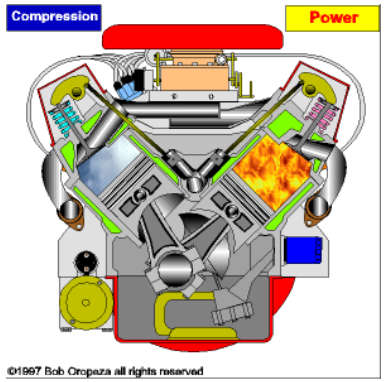
Atuadores

- Os atuadores podem ser de diversos tipos, como:
  - Luminosos;
  - Sonoros;
  - Térmicos;
  - Movimento;
- Os de movimento podem ser:
  - Pneumático;
  - Hidráulico;
  - Combustão;
  - Elétrico;
  - Etc.







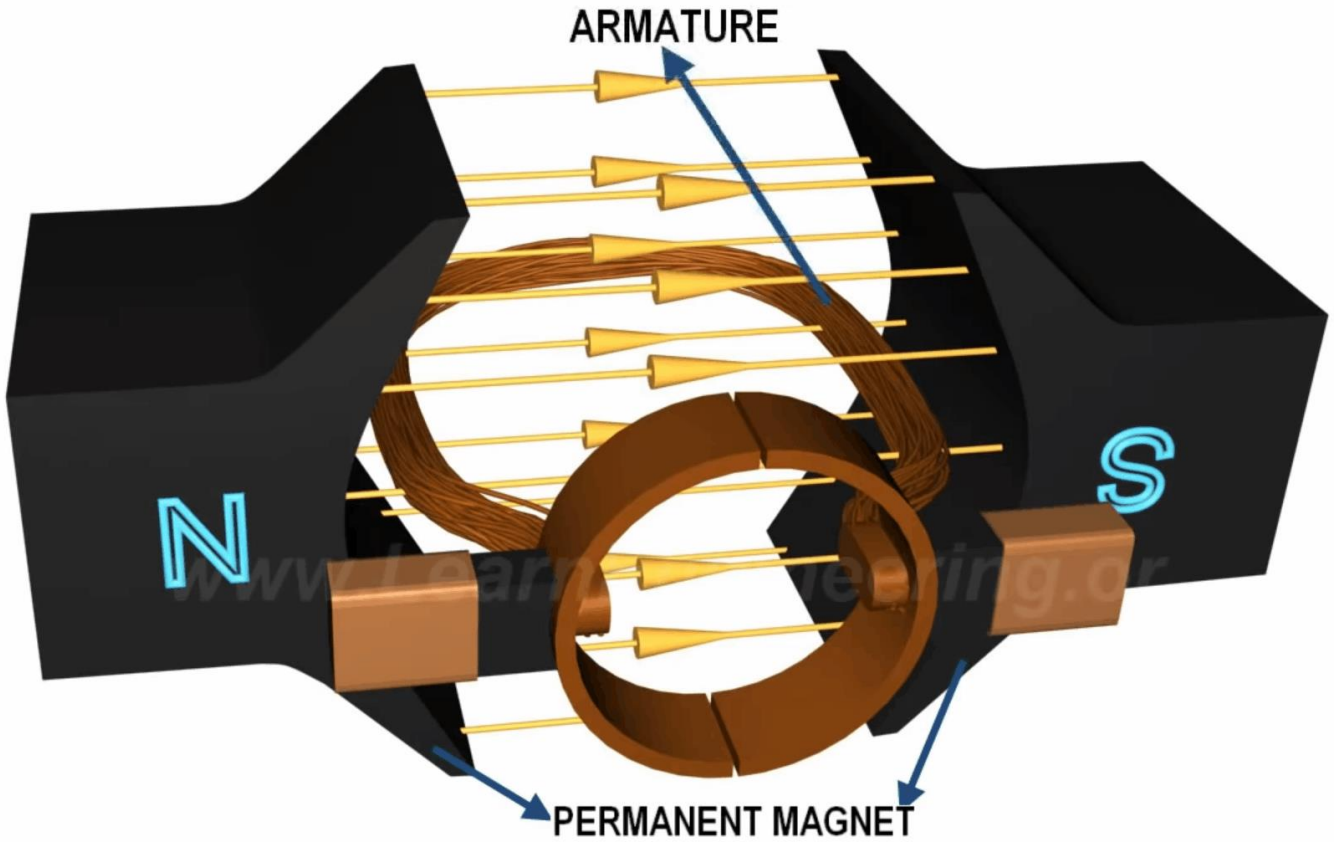


Motor

- Os atuadores são fundamentais na mecatrônica, permitindo ao engenheiro interferir no sistema mudando seu estado;
- Podemos ter acionamentos eletromagnéticos, como motores, pneumáticos, hidráulicos, etc etc.
- O mais utilizados são motores. Estes podem ser divididos da seguinte maneira:
  - Motores DC: com escovas, sem escovas, motores de passo, etc;
  - Motores AC: de indução, de relutância, etc;
  - Outros: motor universal.

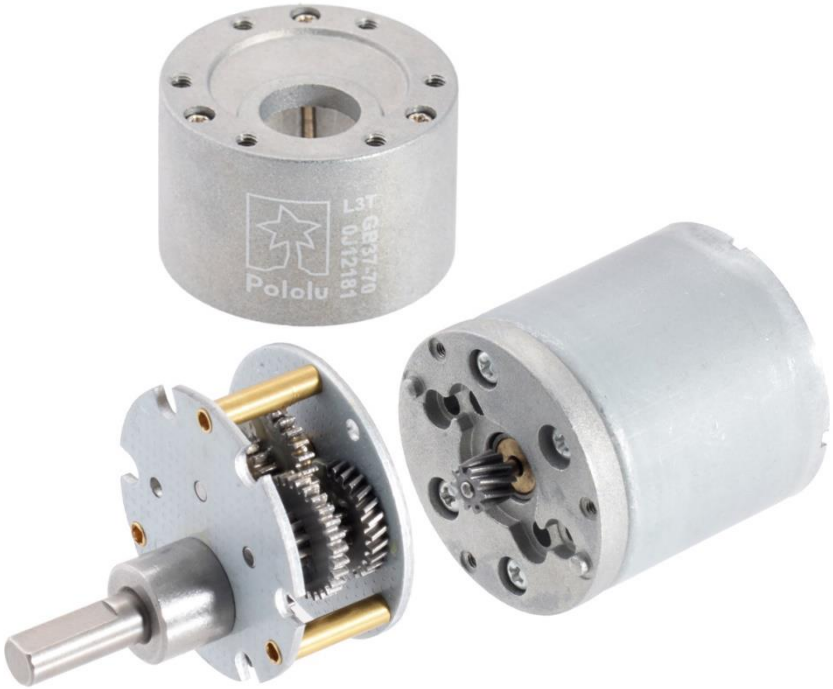
Motor DC com escovas

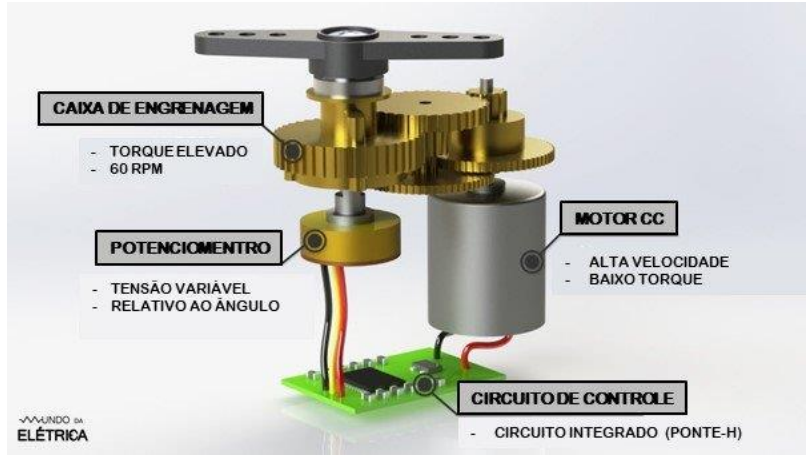
- O gif ilustra o funcionamento geral de um motor DC com escovas;
- Assistir o vídeo completo em DC Motor, How it works? ;



Redução

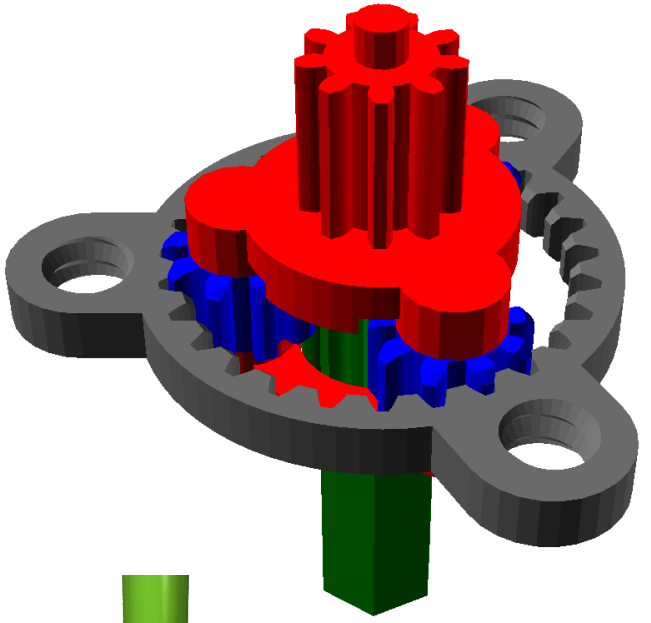
- Os motores são fabricados em grande escala, com características selecionadas para atender as maiores demandas;
- Para adequar o motor especificamente para sua aplicação, podem ser usadas reduções. Estas funcionam como as marchas de bicicleta, modificando torque e velocidade angular;
- Fique atento para o fato de que reduções possuem grande atrito, “gastando” parte do torque do motor que iria para sua aplicação.

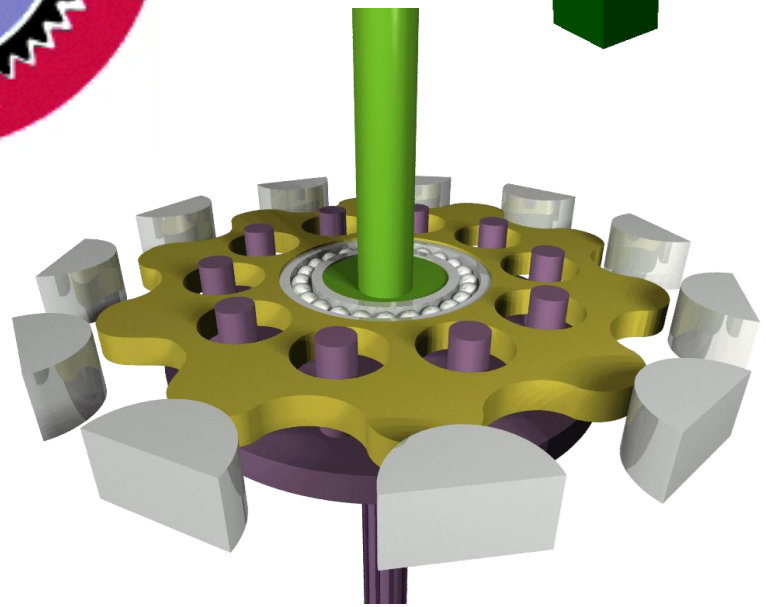




Redução

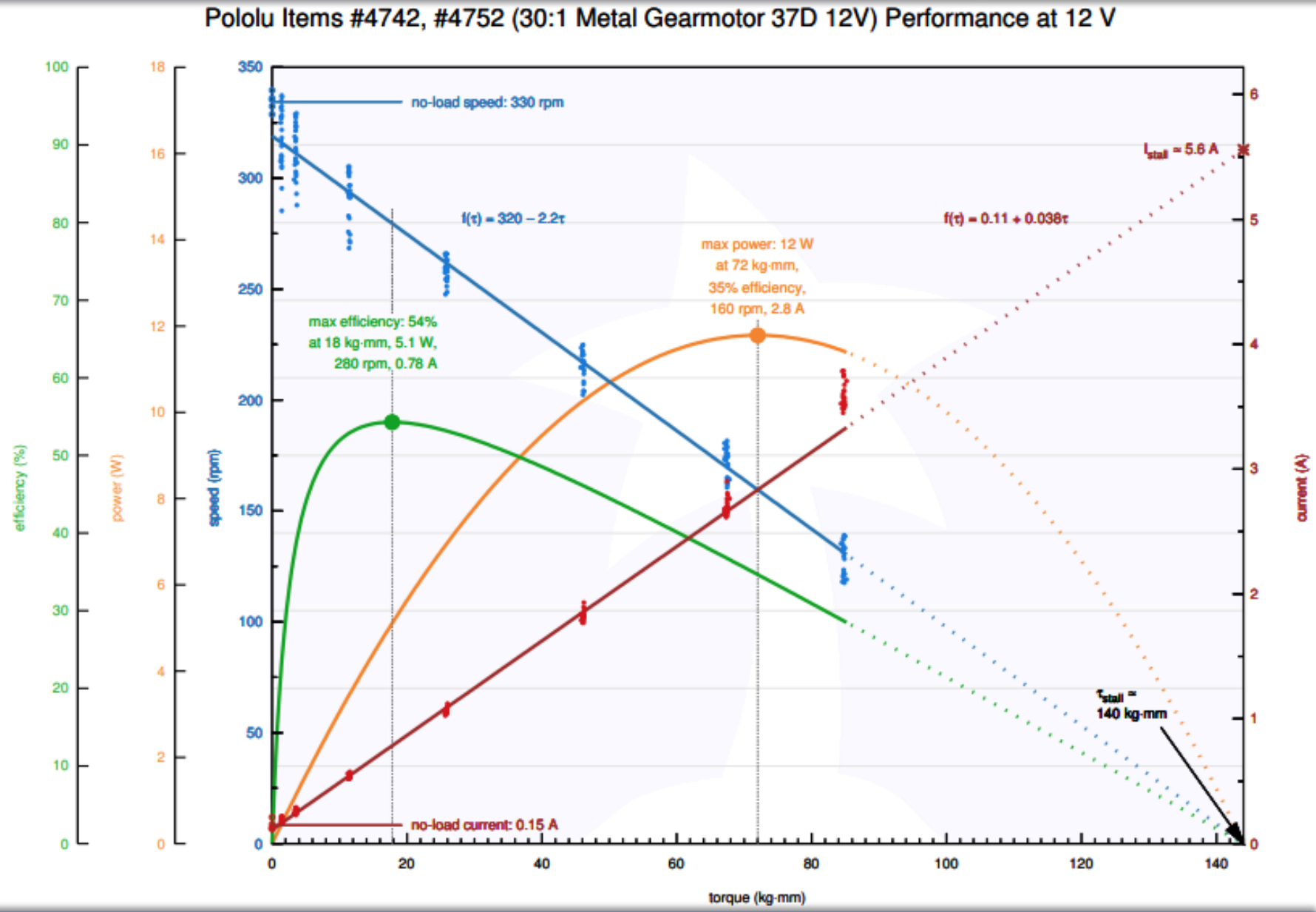
- Exemplos de tipos de redução são:
  - Engrenagens em série;
  - Redução planetária;
  - Redução harmônica;
  - Redução cicloidal;
- Os parâmetros para escolher reduções são:
  - Valor da redução;
  - Relação entre volume e valor de redução;
  - Atrito;
  - Etc.





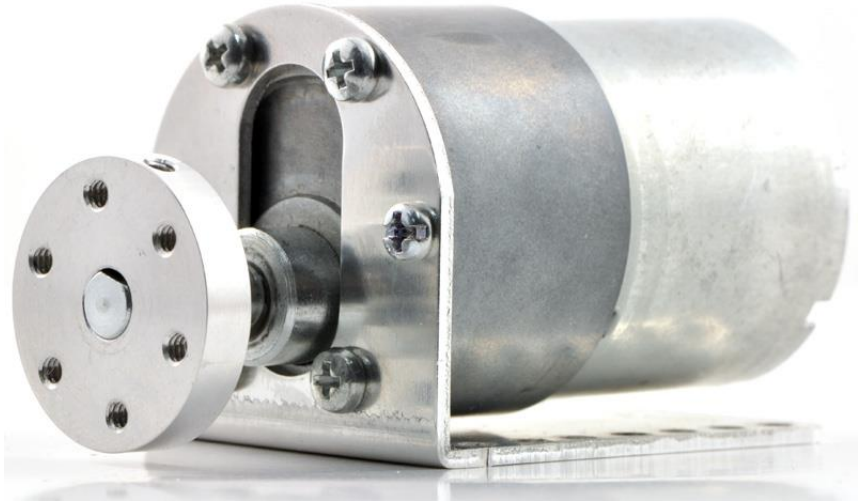
Seleção de moto-redutores

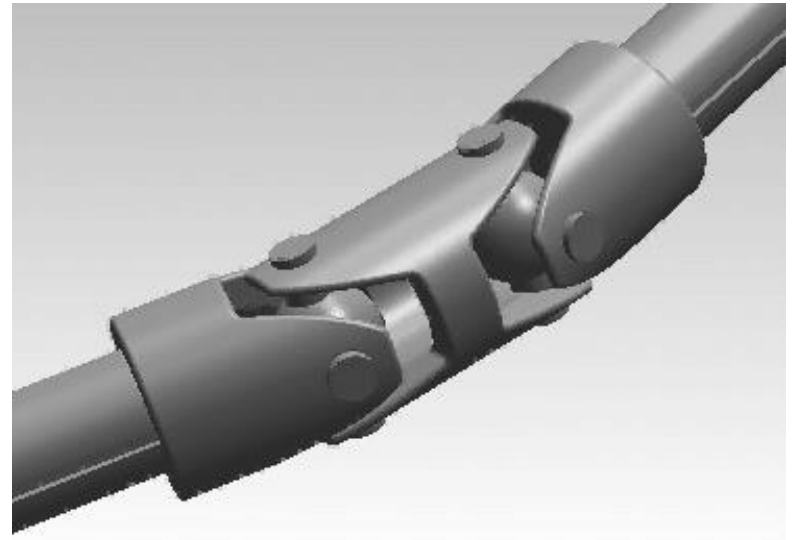
- Para selecionar um motor, devemos abrir seu ***datasheet*** e procurar a curva de Torque por Rotação;
- A curva ao lado foi retirada do datasheet do motor indicado para o projeto.



Acoplamento em eixo

- Os eixos dos motores geralmente vem com chanfros ou rasgos de chaveta para serem acoplados a outras peças e/ou sistemas de transmissão;
- Esta ligação é feita através de acoplamentos;
- Estes devem *atenuar* problemas de desalinhamento entre os eixos e outras partes do sistema.



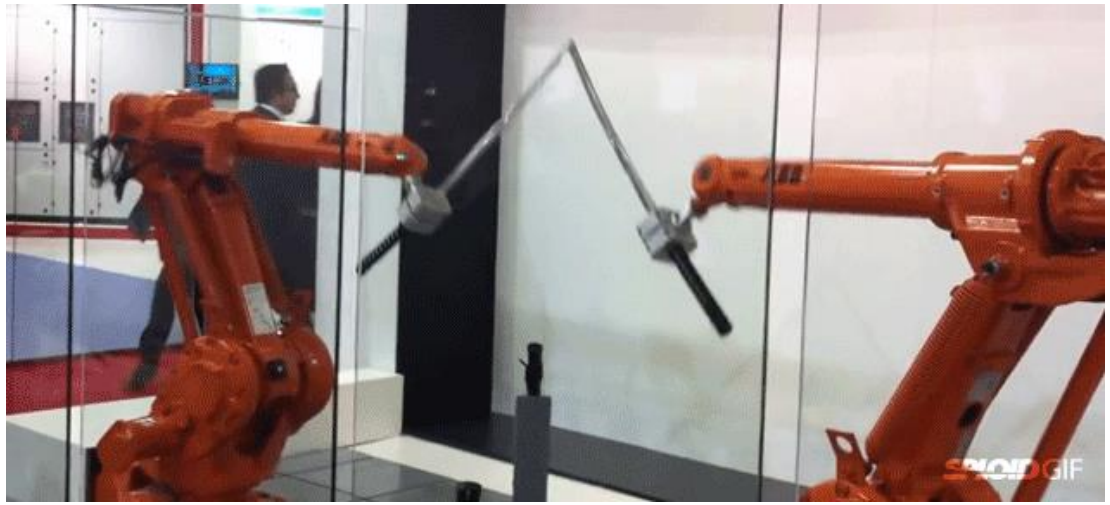


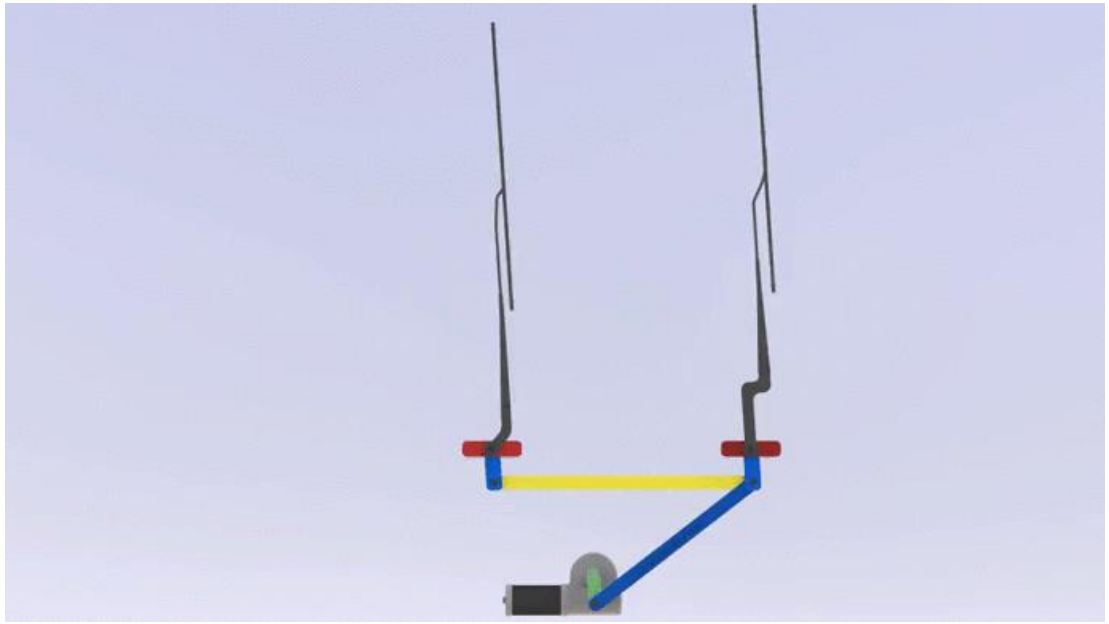
Suporte dos eixos

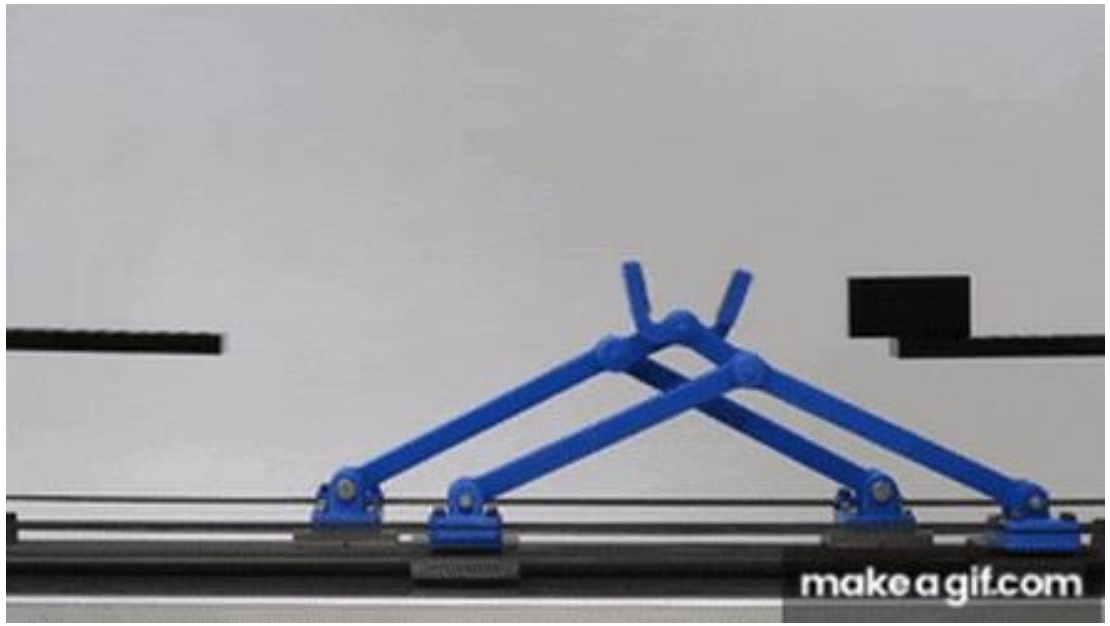
- O suporte dos elementos girantes deve ser feito de forma a minimizar o atrito, ou seja, reduzir o desgaste e desperdício de energia na forma de calor;
- Este suporte é feito através de mancais. Estes podem ser:
  - Mancais de deslizamento;
  - Mancais hidrodinâmicos;
  - Mancais aerostáticos;
  - Mancais magnéticos;
  - Mancais de rolamento;
  - etc

Mecanismo

- Um mecanismo é um dispositivo mecânico que transforma o movimento rotativo em um movimento de trajetória projetada;
- Os mecanismos podem ser divididos em:
  - Mecanismos em série;
  - Mecanismos paralelos;

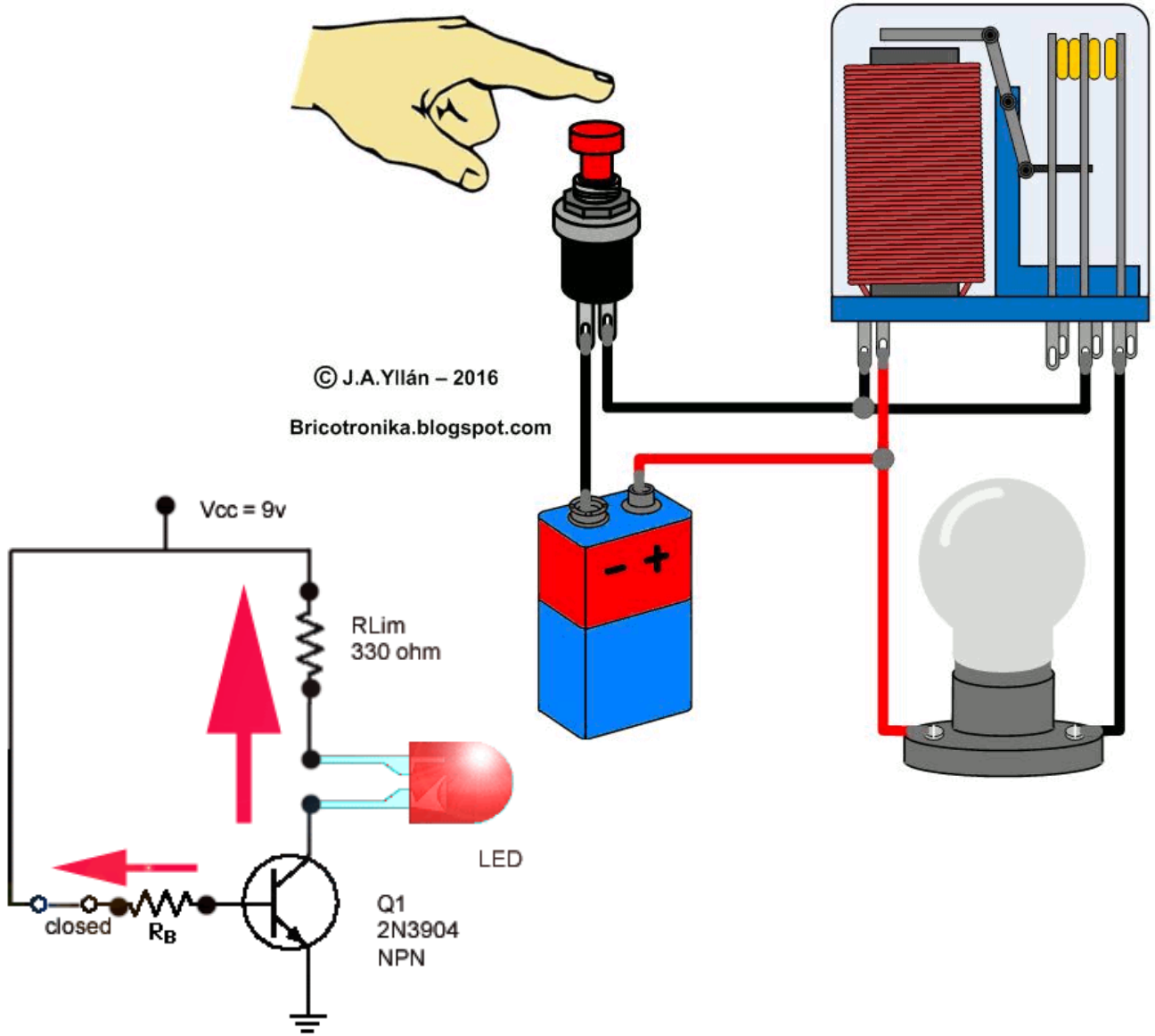






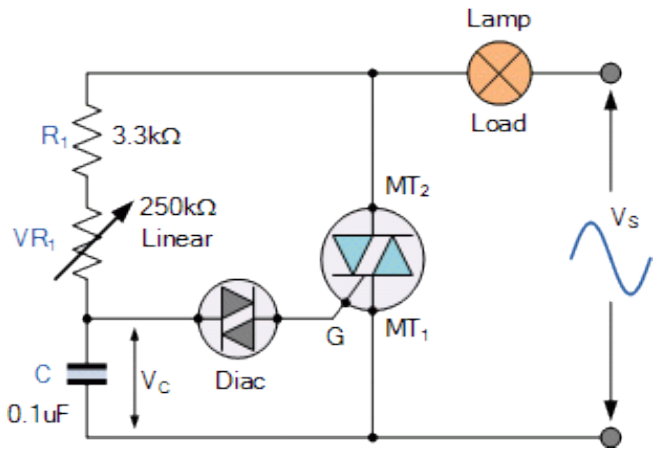
Fornecendo energia elétrica aos atuadores

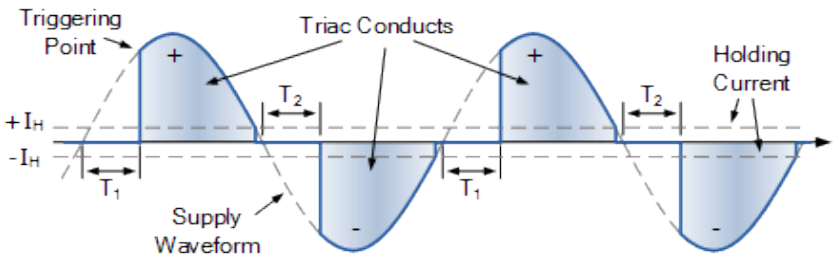
- Existem dois tipos de energia elétrica que podem ser fornecidas a atuadores:
  - Contínua;
  - Alternada.
- Em ambos os casos, podemos ativar ou desativar o atuador controlando o fornecimento ou não de energia através de um interruptor;
- Relês e transistores podem funcionar como interruptores que pode ser acionado remotamente!

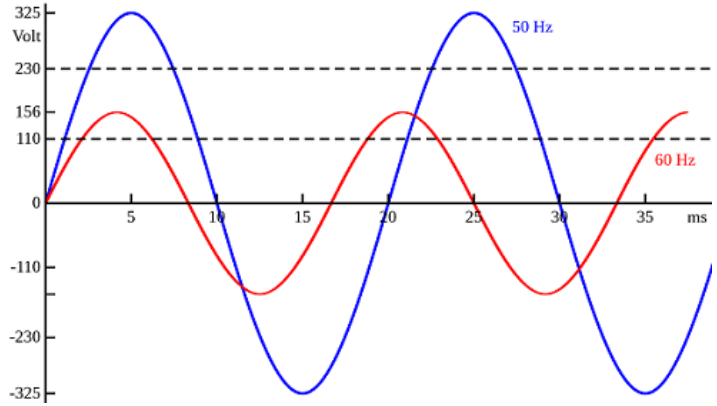


Variando o fornecimento em AC

- Existem dois modos principais de controlar o fornecimento de energia AC: i) Circuitos com Diacs e Triacs e ii) Inversores de frequência;
- No primeiro caso, somente parte da onda senoidal é aplicada a carga;
- No segundo caso, variamos a frequência e/ou amplitude da onda senoidal aplicada à carga;

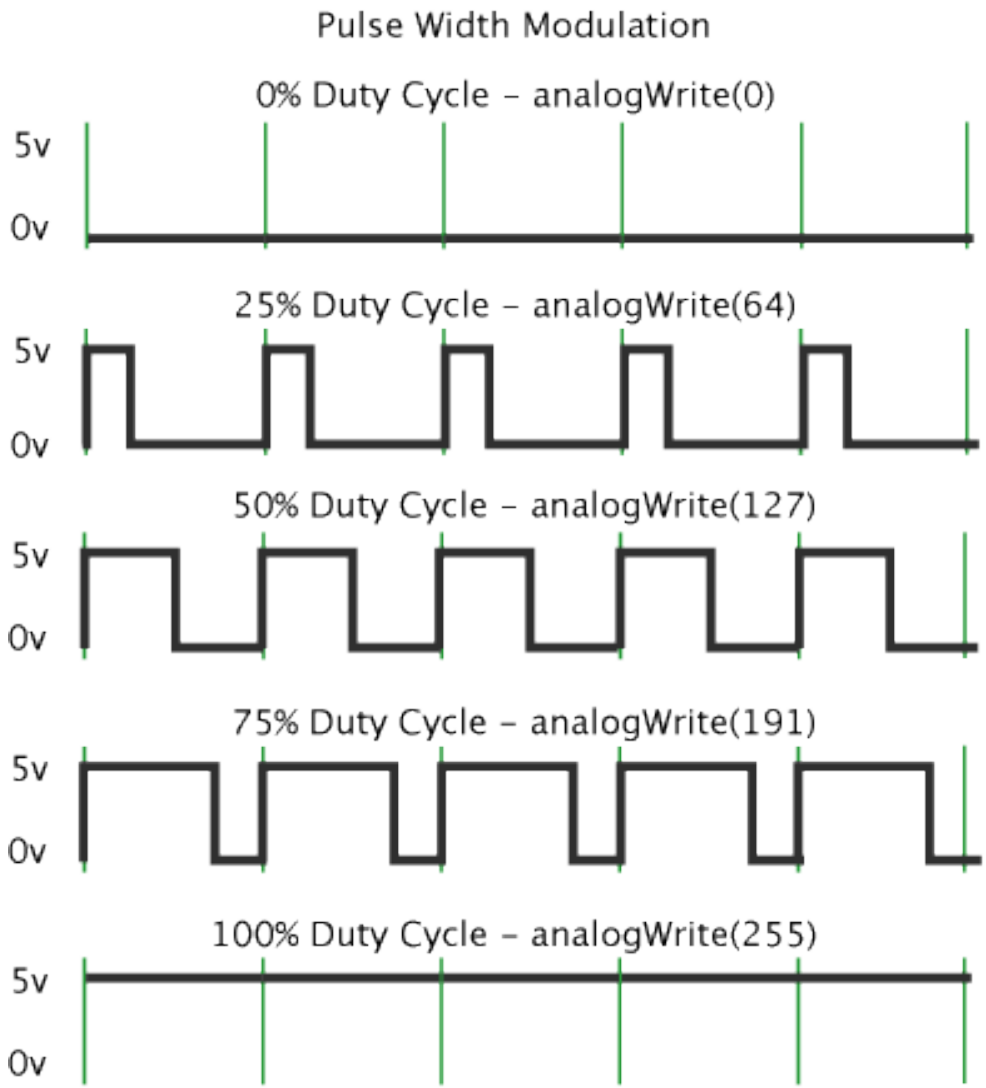






Variando o fornecimento em DC - PWM

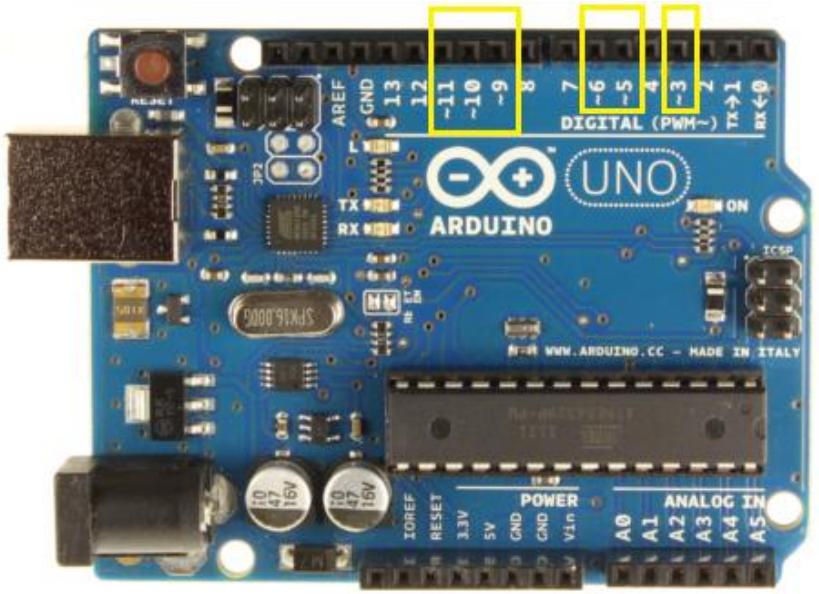
- O gráfico ao lado ilusta a influência da variação do duty cycle no valor analógico médio.
- PWM é a modulação da largura de ondas quadradas.
- PWM, do inglês Pulse Width Modulation, é uma técnica utilizada por sistemas digitais para variação do valor médio da voltagem, enquanto só se aplica na saída voltagens nula ou máxima. A técnica consiste em manter a frequência de uma onda quadrada fixa e variar o tempo que o sinal fica em nível lógico alto. Esse tempo é chamado de duty cycle.



PWM no Arduino

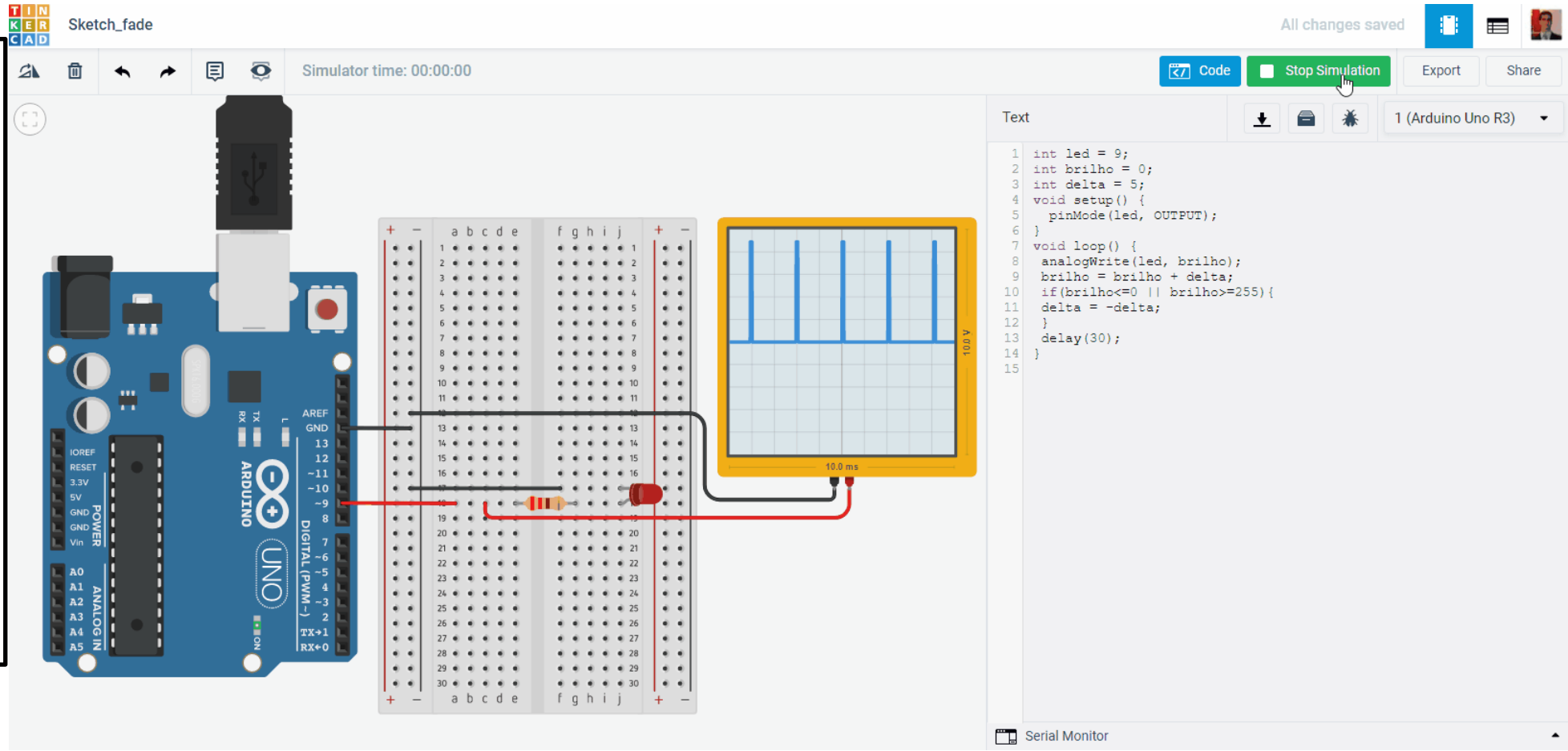
- No Arduino, somente os pinos com ~ possuem PWM por Hardware, sendo controlados pela função `analogWrite(pino,valor)`, no qual o pino corresponde ao pino que será gerado o sinal PWM (3, 5, 6, 9, 10 ou 11) e o valor corresponde ao duty cycle, variando de 0 a 255. Se for 0, a saída permanece sempre em nível baixo (0V) e 255 a saída permanece sempre em nível alto (5V).

A frequência do PWM no Arduino é:
- 1KHz nos pinos 5 e 6
- 500 Hz nos pinos 3,9,10,11



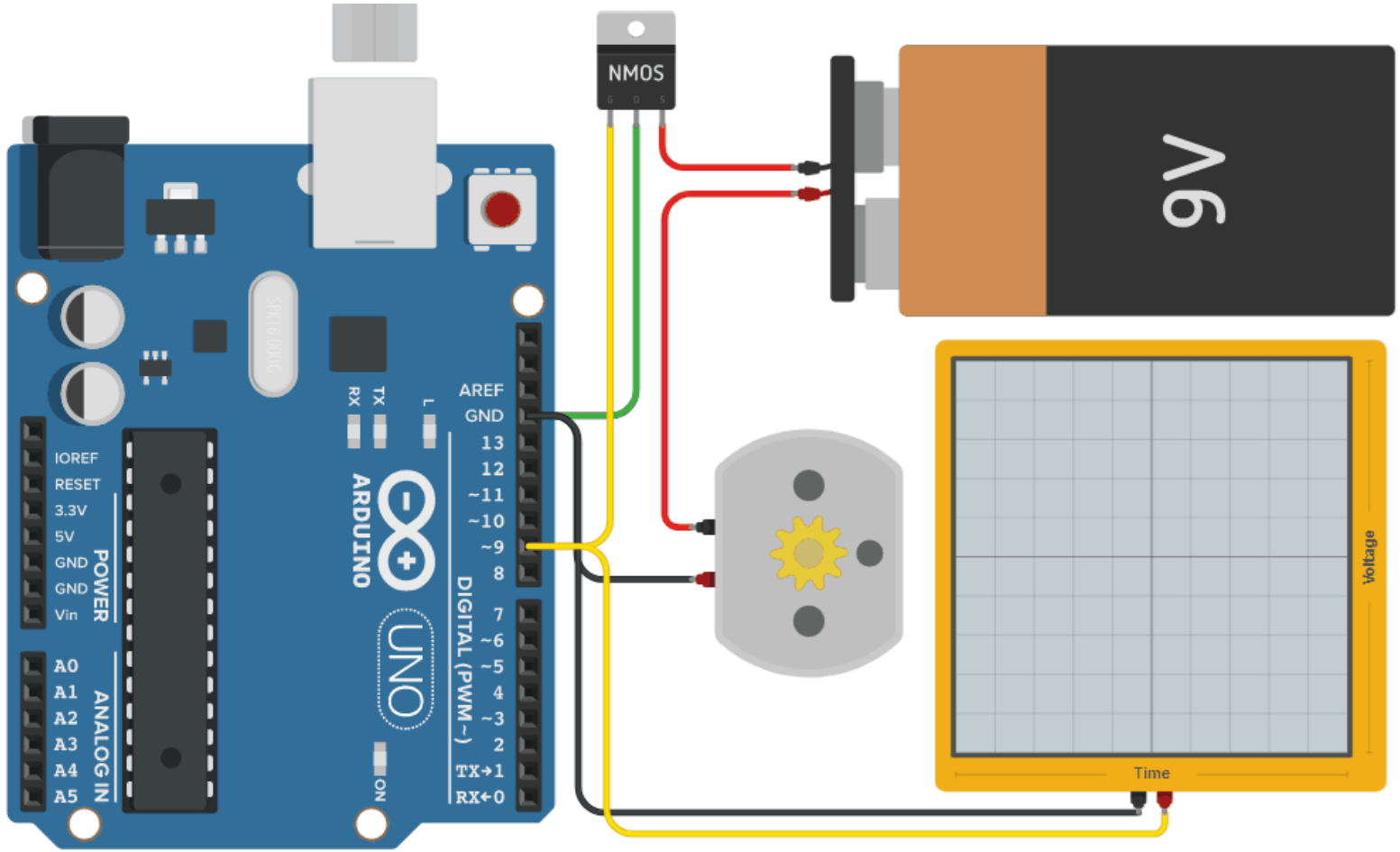
Exemplo 01

```
#define led 9
int delta = 5;
int brilho = 0;
void setup() {
  pinMode(led, OUTPUT);
}
void loop() {
 analogWrite(led, brilho);
 brilho = brilho + delta;
 if(brilho<=0 || brilho>=255){
 delta = -delta;
 }
 delay(30);
}
```



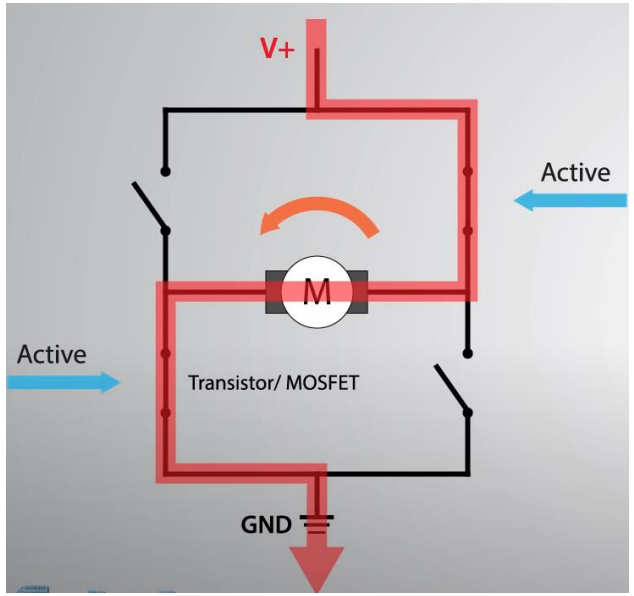
Variando a velocidade de um motor DC

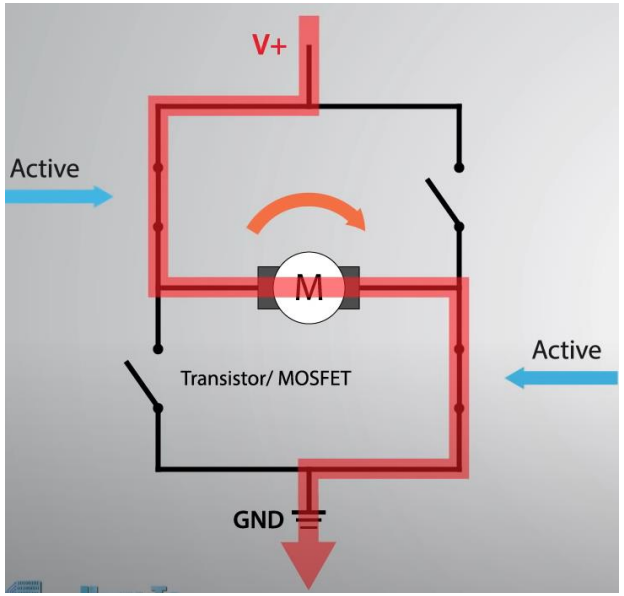
- Já no caso DC, utilizamos circuitos específicos ou microcontroladores que geram PWM;
- Como a saída destes circuitos é de baixa potência, precisamos de um driver para fornecer energia.
- Como exemplos de driver podemos ter um único transistor.

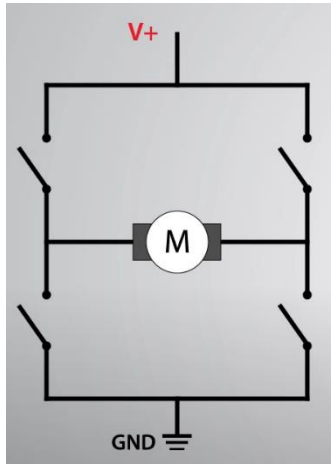


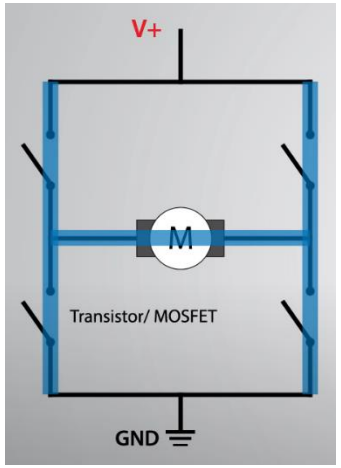
Para um motor DC

- Mas podemos querer ter um circuito que fornecem a energia desejada ao sistema, mas é capaz de inverter o sentido de rotação do motor.
- Este circuito pode ser um driver chamado ponte H.



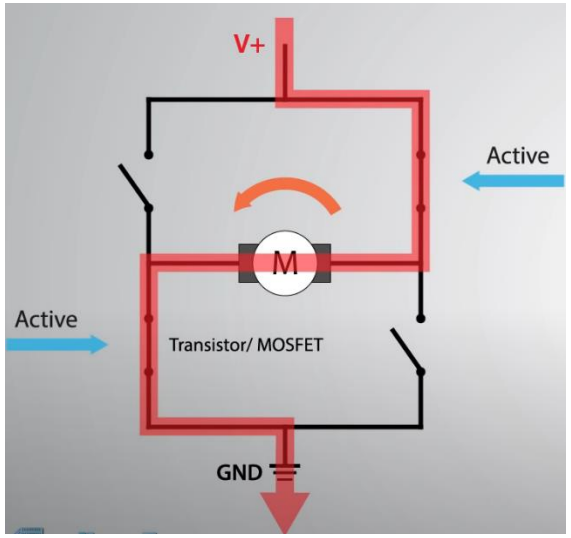


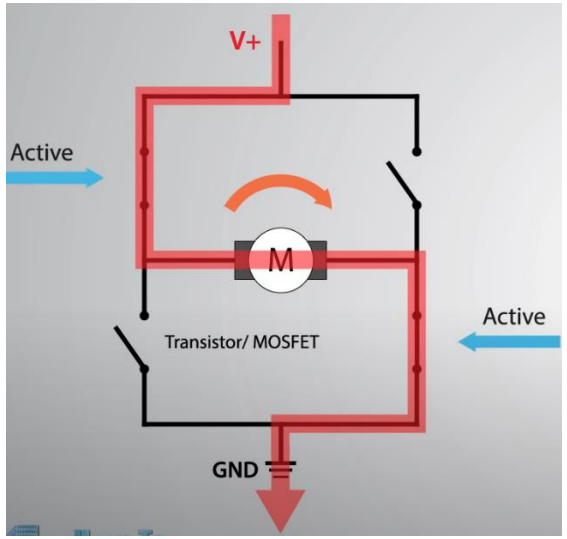


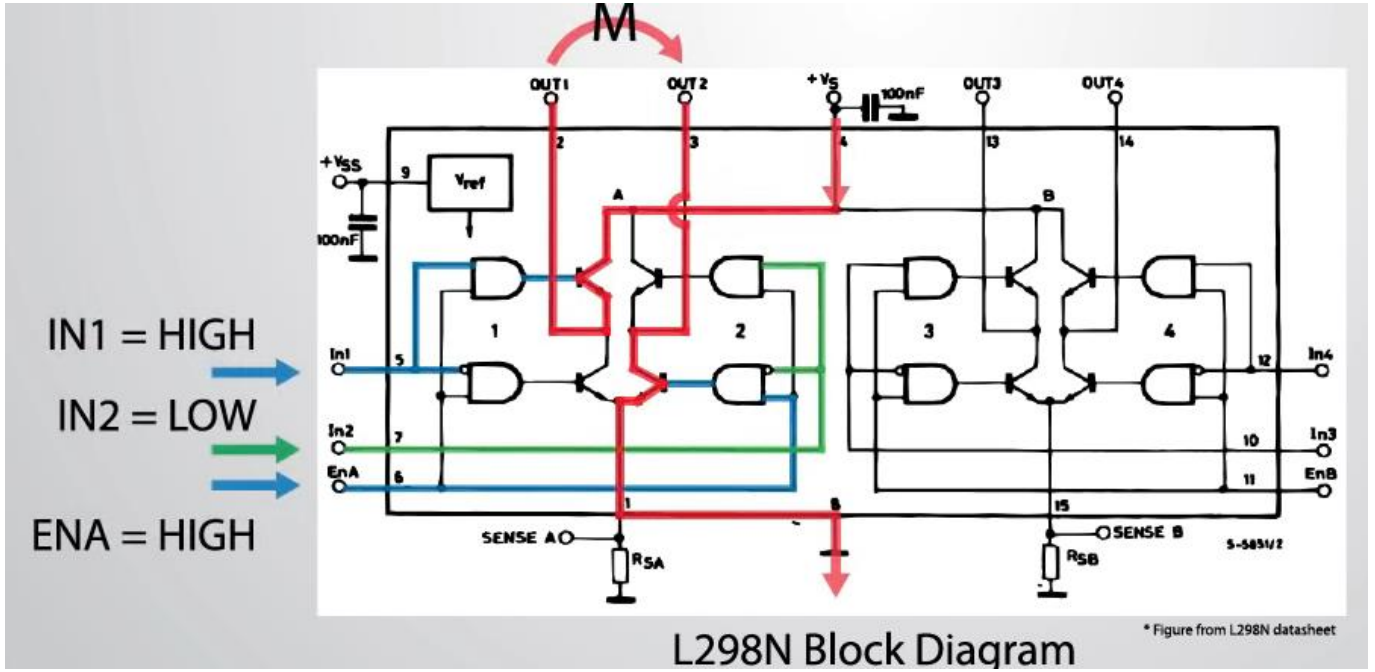


Para um motor DC

- Abaixo está o diagrama de blocos do L298N, que possui duas pontes-H.
- A velocidade do motor é controlada por um sinal PWM no pino ENABLE.
- Se colocarmos HIGH no pino IN1 e LOW no pino IN2, o motor gira para um lado.
- Se colocarmos LOW no pino IN1 e HIGH no pino IN2, o motor gira para o outro lado.

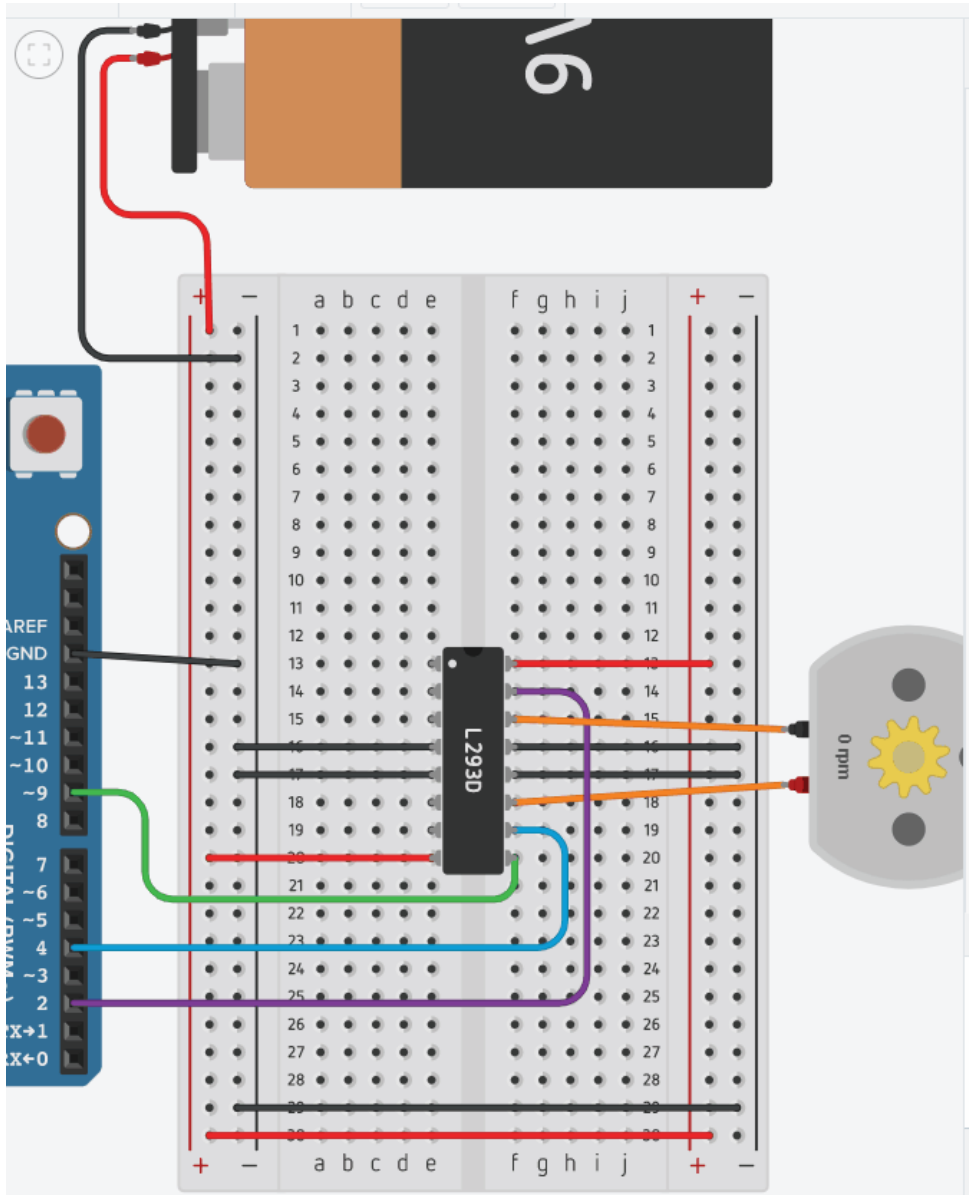






Para um motor DC - circuito

- O circuito ao lado usa a ponte H L293D, conforme explicado anteriormente;
- A fonte de energia é uma bateria que o driver passa ao motor é uma bateria;
- Comandos:
  - `w` acelera,
  - `s` desacelera,
  - `a` gira em um sentido,
  - `d` gira em outro sentido.



Para um motor DC – código parte 1

- A primeira parte do código contém:
  - define
  - declarações de variáveis globais
  - função `setup()` com:
  1. inicialização da comunicação serial em 9600kbps;
  2. definição de 3 pinos como saída (enable, IN1 e IN2);
  3. atribuição inicial de valores para as variáveis globais

```
#define Pino_Vel 9
#define delta 0

int velocidade;
int direcao;
char comando;

void setup() {
  Serial.begin(9600);
  pinMode(Pino_Vel, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(4, OUTPUT);
  velocidade = 0;
  direcao = 0;
comando = 'a';
}
```

Para um motor DC – código parte 2

- A função `loop()` começa verificando se há algo no buffer da porta serial;
- Caso exista:
  - Coloca-se o valor do buffer na variável `comando`.
  - Altera-se o valor da variável `direcao` caso o comando seja `a` ou `d`;
  - Altera-se o valor da variável `velocidade` caso o comando seja `w` ou `s`.

```
void loop() {
  if(Serial.available()){
       comando = Serial.read();
    Serial.print(comando);
    Serial.print("   ");
    if(comando=='a'){
      direcao = 0;
      }
    if(comando=='d'){
      direcao = 1;
      }
    if(comando=='w'){
      velocidade = velocidade + delta;
      }
    if(comando=='s'){
      velocidade = velocidade - delta;
      }
}
```

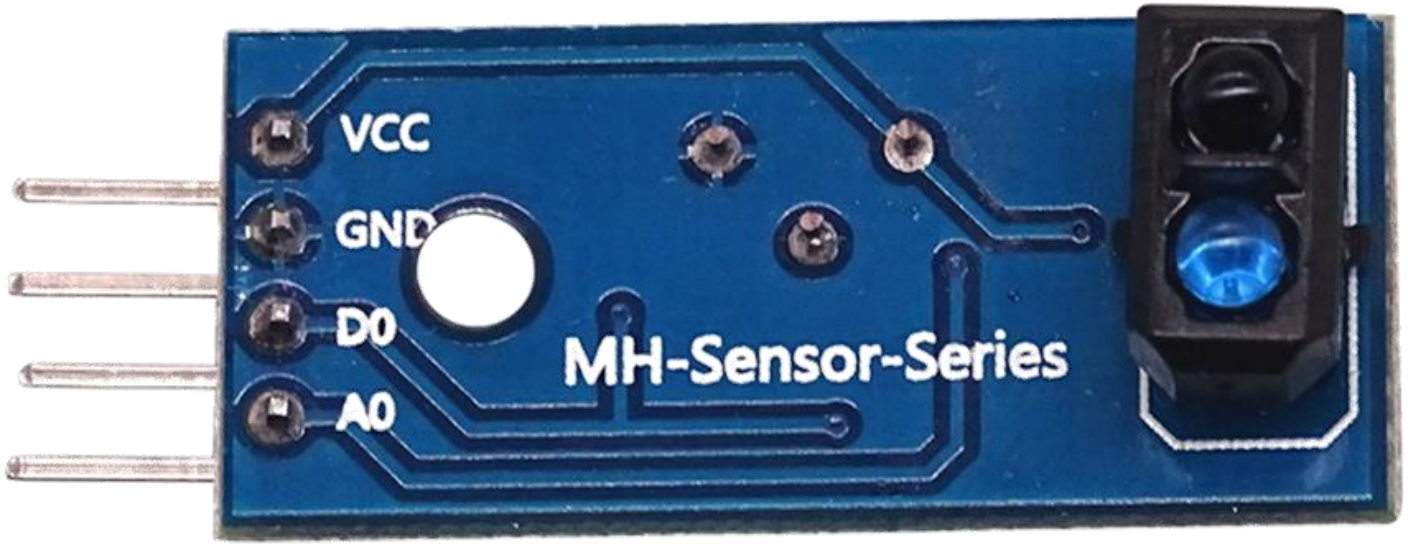
Para um motor DC – código parte 3

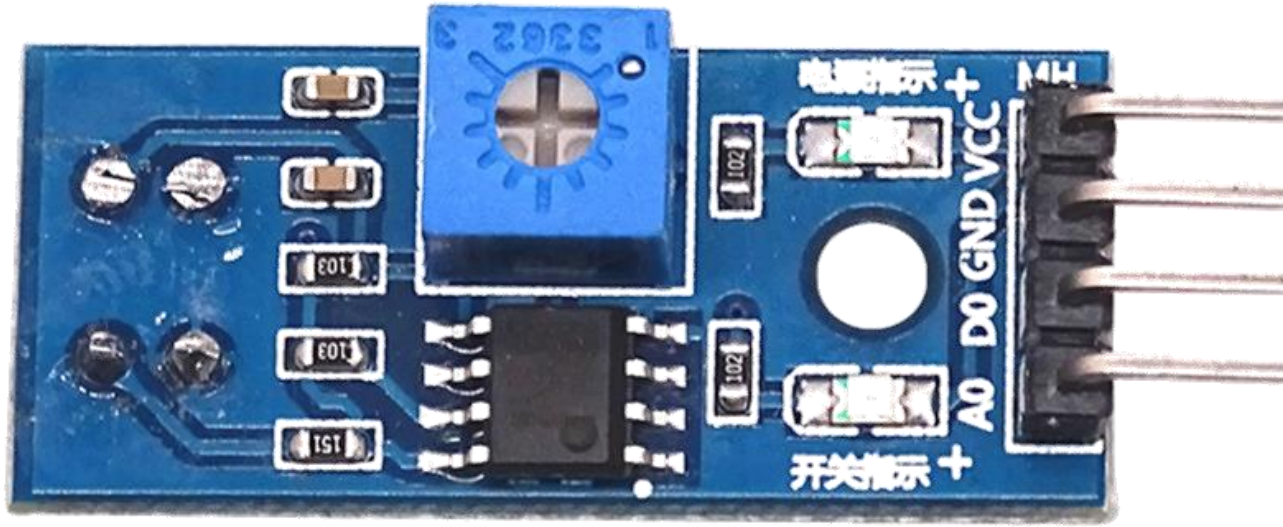
- Ainda na função `loop()` verificamos se os valores para o atuador estão saturados. Caso estejam, corrigimos os valores.

```
if(velocidade <= 0){
  velocidade = 0;
}
if(velocidade >= 255){
  velocidade = 255;
}
```

Sensor para seguidor de linha

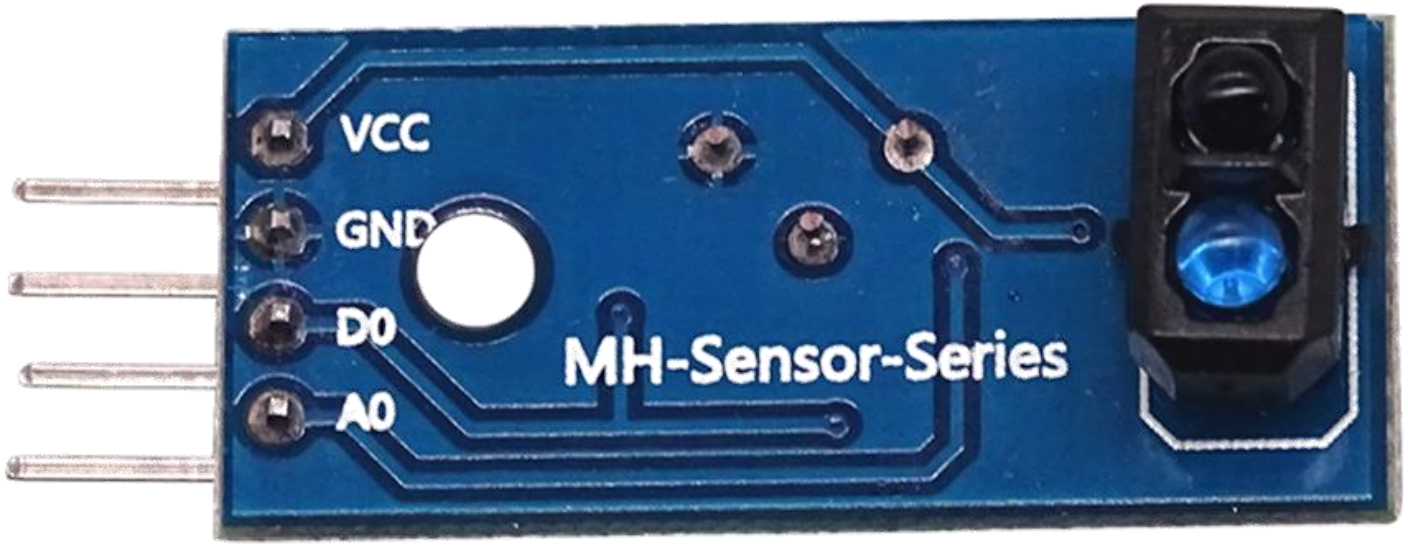
- A figura ao lado é o módulo sensor reflexivo infravermelho. Este é o sensor principal para desenvolver robôs seguidores de linha!
- O sensor conta com um transmissor e um receptor de infravermelho. O mesmo deve estar posicionado próximo ao chão e deve-se evitar que haja luz externa sendo captada pelo receptor.

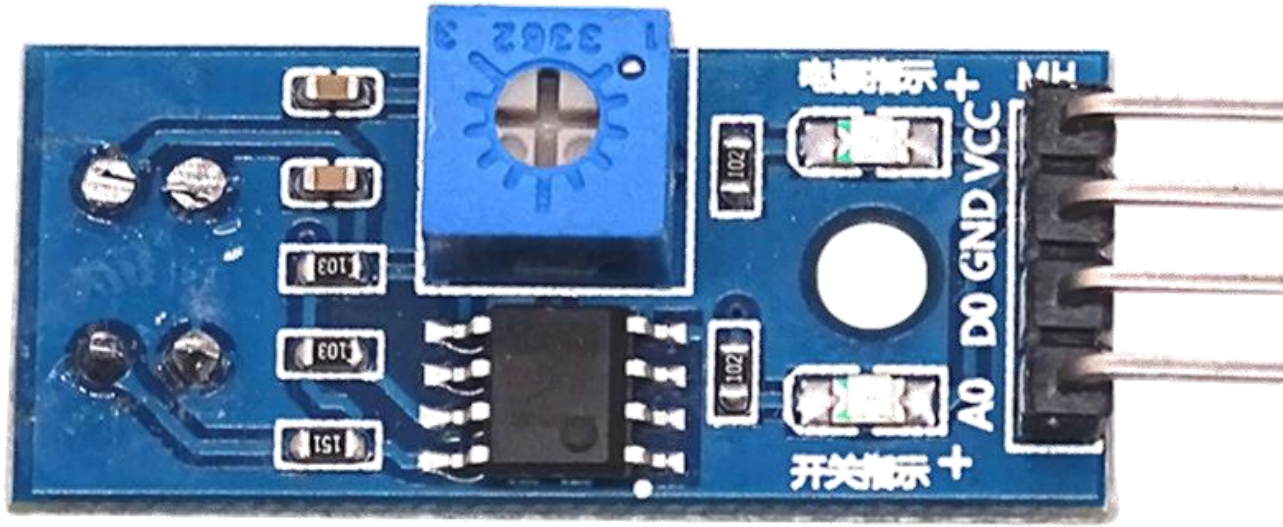




Sensor para seguidor de linha

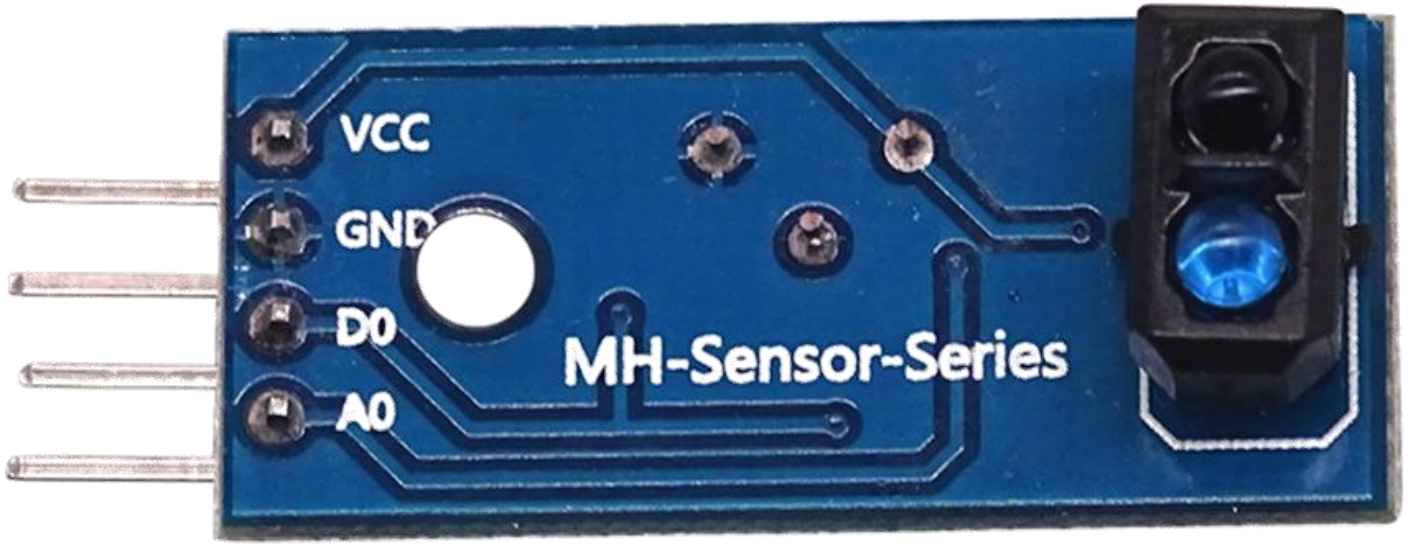
- Assim como a grande maioria dos módulos, sensores ou atuadores que iremos estudar na mecatrônica, este sensor possui uma entrada de terra (GND) e uma de alimentação VCC, nesse caso 5V.
- A quantidade de luz captada pelo receptor é traduzida em voltagem de 0 a 5V e no pino A0. Desta forma, deve ser colocado em um ADC do Arduino.

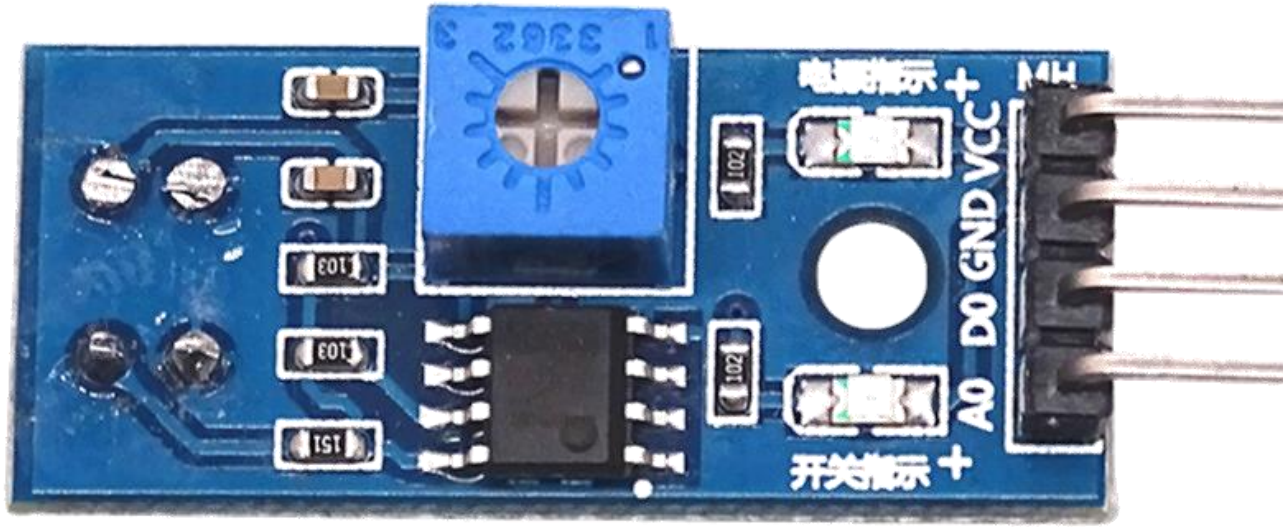




Sensor para seguidor de linha

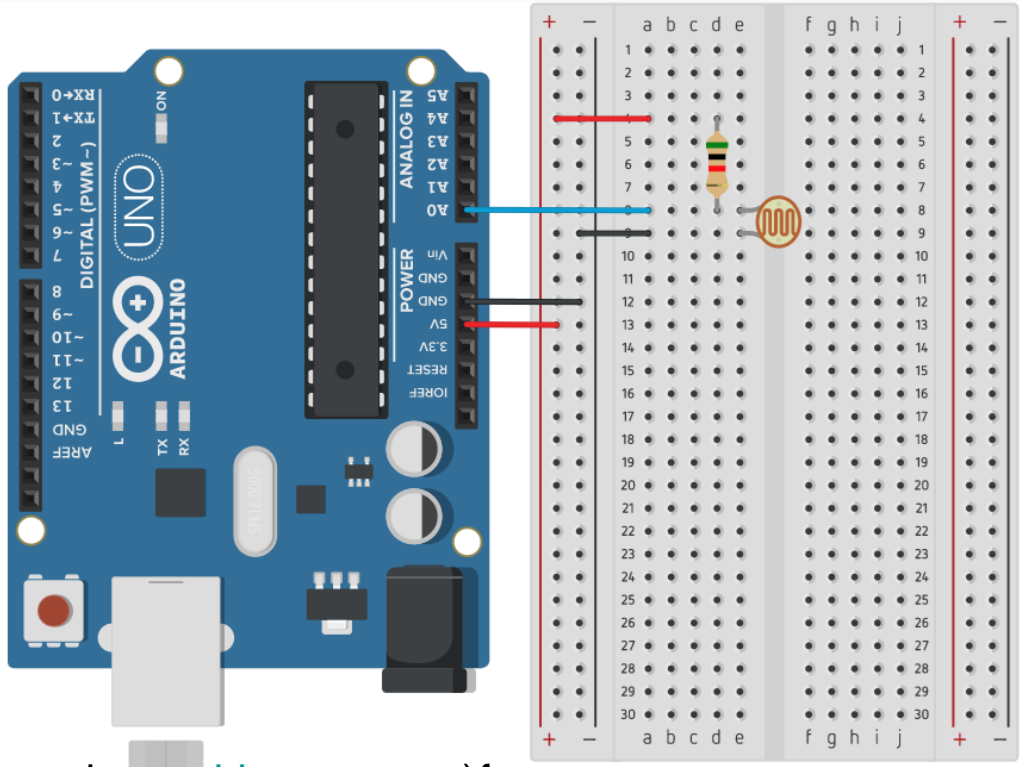
- O saída no pino D0 é reguada pelo usurário no potenciômetro (“*bloco azul com parafuso braco*”).
- Se o valor de voltagem for maior do que o regulado no potenciômetro, D0 é HIGH. Se for menor, é LOW.





Para simular em casa:
Simulando o sensor de linha com photoresistor R=5KΩ

```
int sensorPin = A0;
int sensorValue = 0;
double Voltagem = 0.0;
 void setup() {
 Serial.begin(9600);
}
void loop() {
  sensorValue = analogRead(sensorPin);
  Voltagem = interpolacao((double)sensorValue,0.0,1023.0,0.0,5.0);
 Serial.print("Nivel Logico ADC: ");
 Serial.print(sensorValue);
 Serial.print("  Voltagem: ");
 Serial.print(Voltagem);
 Serial.println("V");
}
double interpolacao(double valor, double old_min, double old_max, double new_min, double new_max){
  double derivada;
  double delta_old;
  double valor_new;
  derivada = (new_max-new_min)/(old_max-old_min);
  delta_old = valor - old_min;
  valor_new = new_min + derivada*delta_old;
  return(valor_new);
}
```

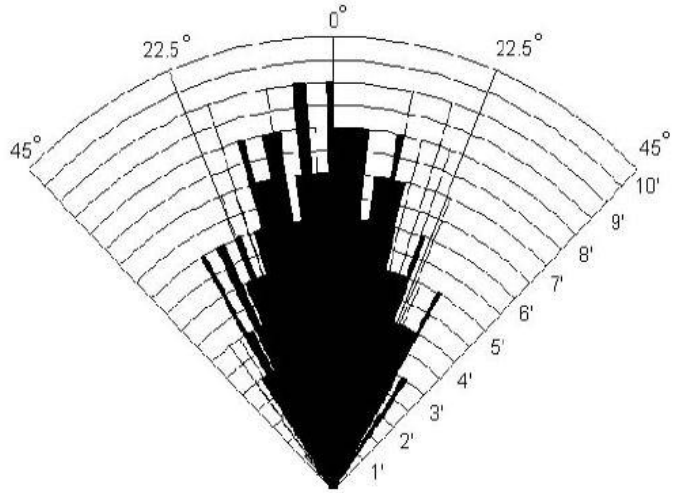


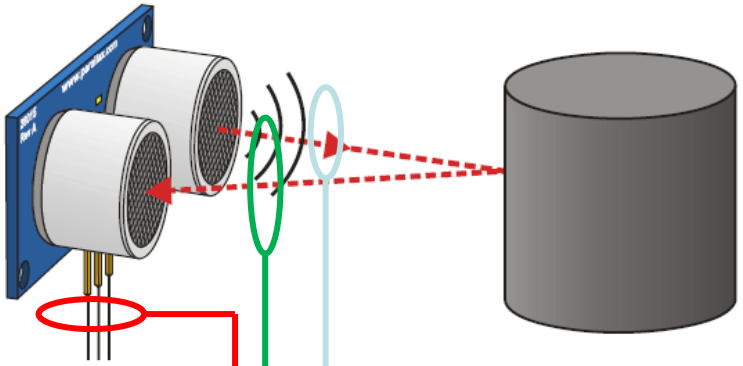
Sua vez

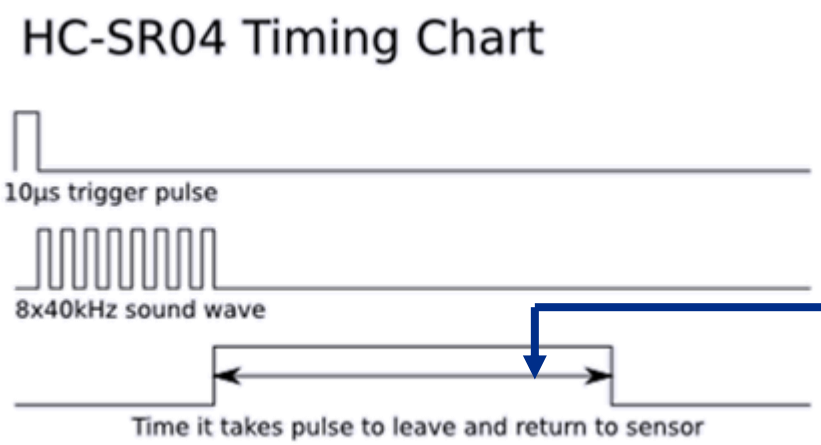
- Entre no moodle e baixe o exemplo de sensor de linha.
- Ligue o sensor de linha no shield de protobord, com VCC em 5V, GND em Gnd e A0 em A0.
- Vá em tools e abra o serial ploter;
- Carregue o código passe o sensor sobre a folha de teste.
- Avalie o gráfico em tempo real e a velocidade do motor.

Sensor de distância ultrassônico

- Sensor para medir distância pelo uso de ultrassom;
- Medidas entre 2cm e 400cm com precisão de 0,3cm;
- A leitura deve ser a cada 50ms ou mais (valor a ser programado no delay).





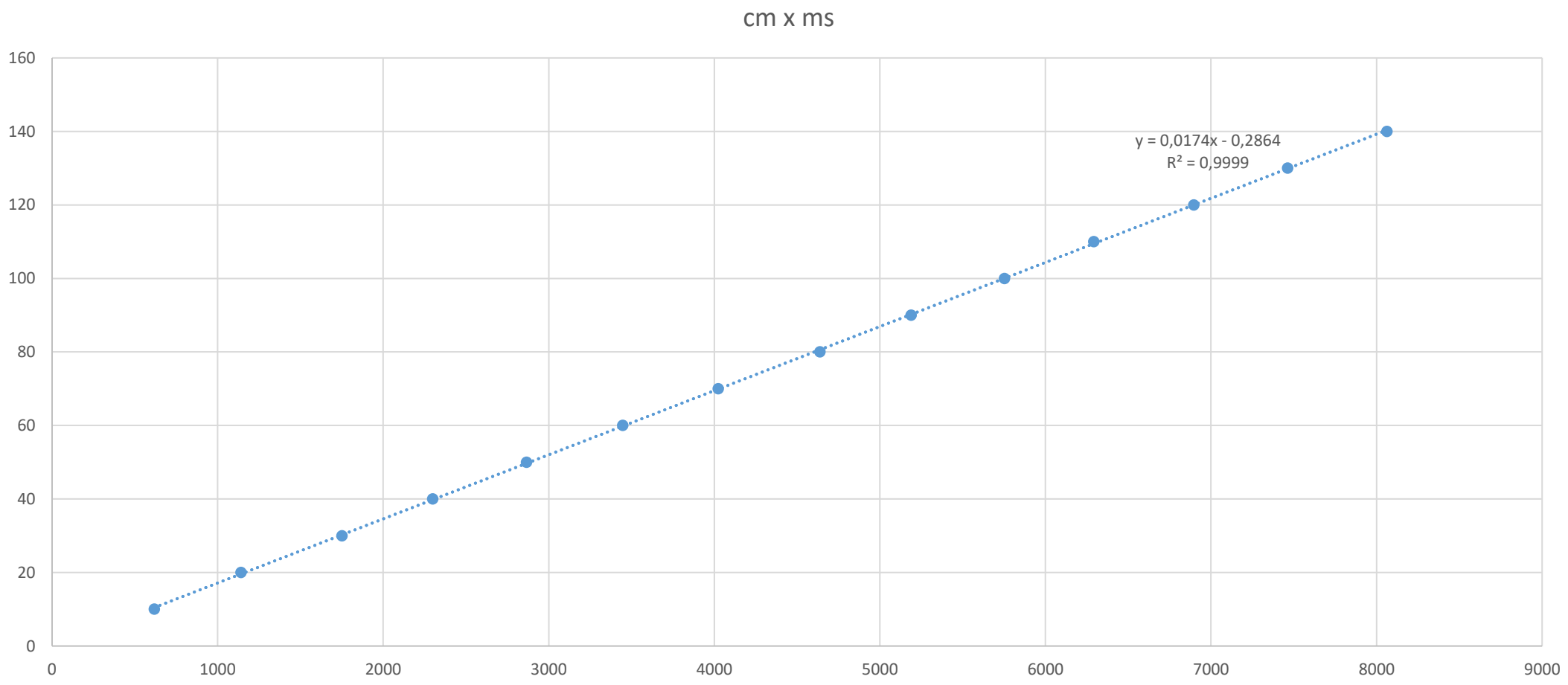


Medir com a função `pulseIn()`

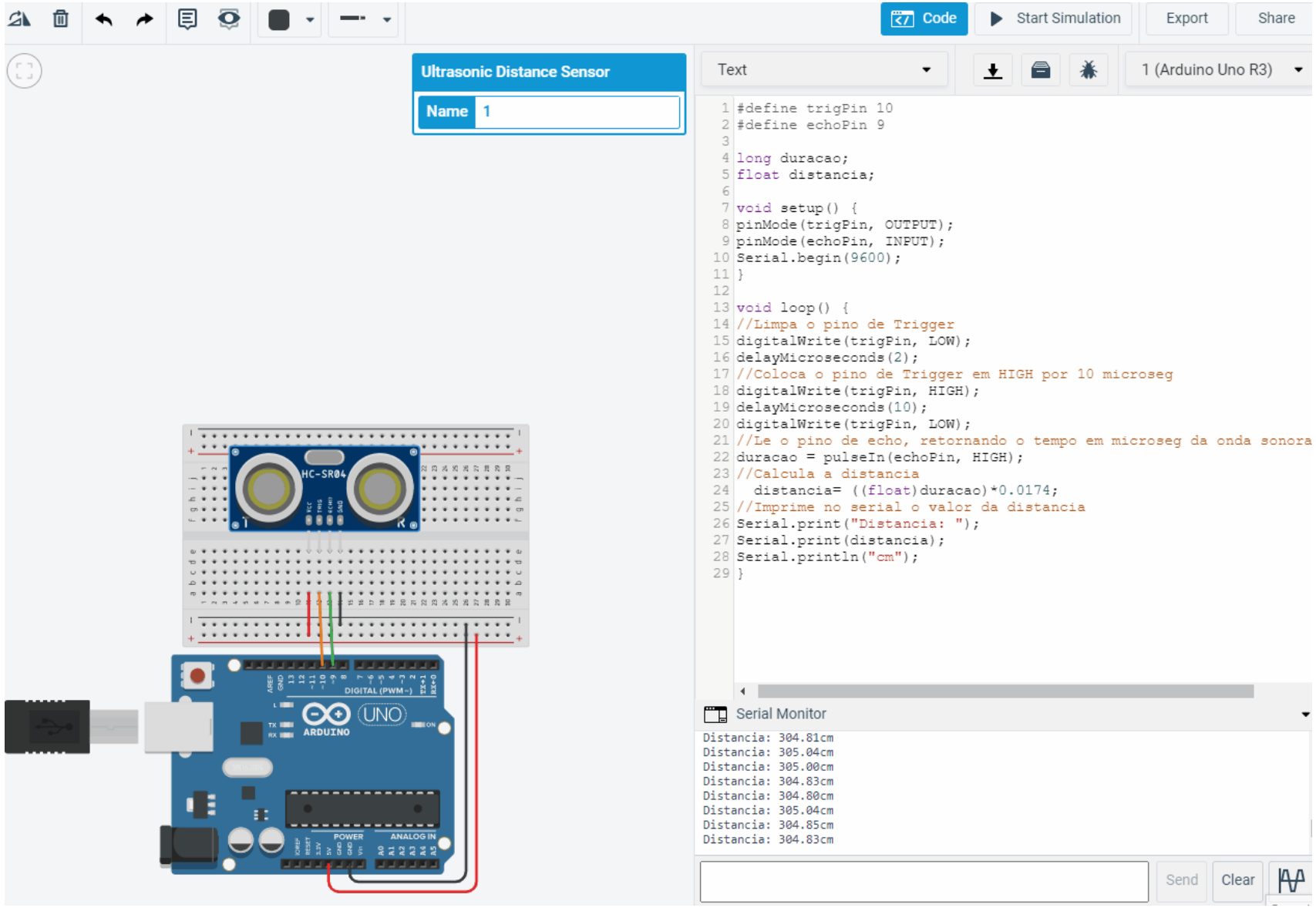
Sensor de distância ultrassônico

ms	cm
617	10
1141	20
1750	30
2299	40
2866	50
3446	60
4024	70
4637	80
5188	90
5753	100
6292	110
6896	120
7462	130
8063	140

Distância (cm) = 0,0174 * Tempo (ms)



Para simular em casa:
Sensor para seguidor de linha - circuito



Para simular em casa:
Sensor para seguidor de linha – códido parte 1

- A primeira parte do código contém:
  - define
  - declarações de variáveis globais
  - função `setup()` com:
  1. inicialização da comunicação serial em 9600kbps;
  2. definição do pino de trigger como saída;
  3. definição do pino de echo como entrada.

```
#define trigPin 10
#define echoPin 9

long duracao;
float distancia;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
```

Para simular em casa:
Sensor para seguidor de linha – códido parte 1

- Na função `loop()` temos:
  - Limpamos o pino de trigger colocando em LOW por 2 microssegundos;
  - Com o pino limpo, colocamos o pino de trigger em HIGH por 10 microssegundos. Neste momento são enviados 8 pulsos ultrassônicos;
  - Assim que os pulsos são enviados, o pino echo fica em HIGH até o momento quem que recebe um dos pulsos refletidos, indo para LOW. Medimos esse tempo com a função `pulseIN()`.

```
void loop() {
//Limpa o pino de Trigger
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
//Coloca o pino de Trigger em HIGH
// por 10 microseg
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
//Le o pino de echo, retornando o tempo
//em microseg da onda sonora ir e vir
  duracao = pulseIn(echoPin, HIGH);
```

Para simular em casa:
Sensor para seguidor de linha – códido parte 1

- Na função `loop()` temos ainda:
  - Cálculo da distância baseado na tebela anterior;
  - Envio da distância via comunicação serial;

```
//Calcula a distancia
  distancia= ((float)duracao)*0.0174;
//Imprime no serial o valor da
//distancia
  Serial.print("Distancia: ");
  Serial.print(distancia);
  Serial.println("cm");
} //fim da função loop
```

Sua vez

- Entre no moodle e baixe o exemplo de sensor de distância.
- Ligue o sensor de linha no shield de protobord, com VCC em 5V, GND em Gnd Echo no pino 9 e Trig no pino 10;
- Vá em tools e abra o serial ploter;
- Carregue o código e use a folha de teste como anteparo.
- Avalie o gráfico em tempo real e a velocidade do motor.