

Robótica III

Professores

Antonio Fernando Traina – Professor da FATEC – Franca
Doutor em Física Aplicada Computacional - IFSC-USP,
aftraina@gmail.com



Roseli Aparecida Romero – Coordenadora do Curso
Professora ICMC-USP,
rafrance@icmc.usp.br



Março -2019

Agenda do curso

Introdução

- Conceitos Iniciais
- A Olimpíada Brasileira de Robotica - OBR
- Conceitos de Arduino
- Conceitos de Sensores e atuadores
- Apresentação do Kit básico
- Plataforma e simulação

Programação para arduino

- Apresentação das Regras da Olimpíada Brasileira de Robotica
- Estrutura da Linguagem do Arduino - linguagem C
- As portas de E/S do Arduino e suas funções em C
- Programando os sensores e atuadores



Aula de Hoje

Desenvolvendo Programação robô seguidor de linha

- Motores
- Sensores claro/escuro
- Sensores de cor
- Sensores de distância
- Aprimoramentos e melhorias

Competição entre equipes – Organizar.

Material para a aula de hoje

Item	Descrição	Preço	Quantidade
1	Kit Robótica Robô Segue Linhas Chassi 2 Rodas	53.45	1
2	Placa de desenvolvimento	26.00	1
3	Cabo USB A-B	9.90	1
4	jumper macho-femea (10 pares)	15.00	1
5	Bateria 9 volts + cabo	25.25	1
6	Motor Shield Driver Ponte H L298N	23.00	1
7	Mini Protoboard 170 Pontos	9.50	1
8	Sensor De Linha Segue Faixa Infravermelho Tcrt5000	9,00	2
	TOTAL	162.10	9

Robótica III

Aula II

Março -30/2019

Agenda de hoje

Conceitos básicos de Física

- Conhecendo os sensores e atuadores
 - Motores e Ponte H
 - Sensores de Linha



Montagem de um exemplo de carrinho segue linha

- A importância do robô seguidor de linha
- Como o carrinho de um robô móvel funciona?
- Motor DC com Driver Ponte H L298N
- Sensor segue linha TCRT5000

Agenda de hoje

Estrutura da Linguagem do Arduino (linguagem C)

As portas de E/S do Arduino e suas funções em C

Conceitos básicos de Física

- Tensão, corrente & resistência
- Componentes básicos
- Conhecendo os sensores e atuadores
 - Motores e Ponte H
 - Sensores de Linha



Montagem de um exemplo de carrinho segue linha

- A importância do robô seguidor de linha
- Como o carrinho de um robô móvel funciona?
- Motor DC com Driver Ponte H L298N
- Sensor segue linha TCRT5000

Conhecendo os sensores e atuadores

Motores e Ponte H
Sensores de Linha

Sensores de Linha

Funcionamento do modulo infravermelho

O módulo sensor de obstáculo infravermelho IR emite luz infravermelha por um LED negro e capta o reflexo com o LED receptor (LED claro).

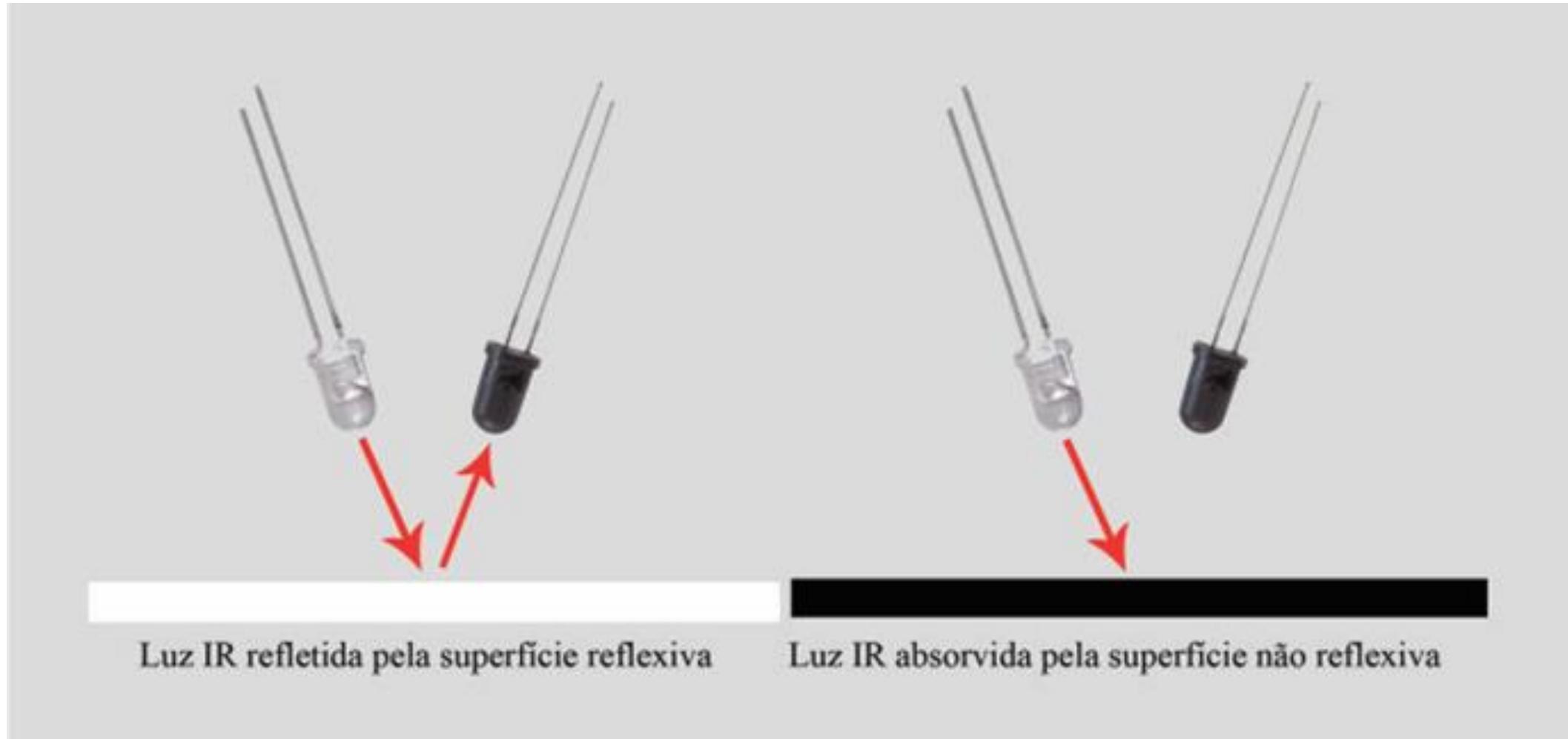
A luz reflete em superfícies claras e é absorvida em superfícies negras.

O LED receptor irá detectar a luz infravermelha no branco e não detectar no preto.

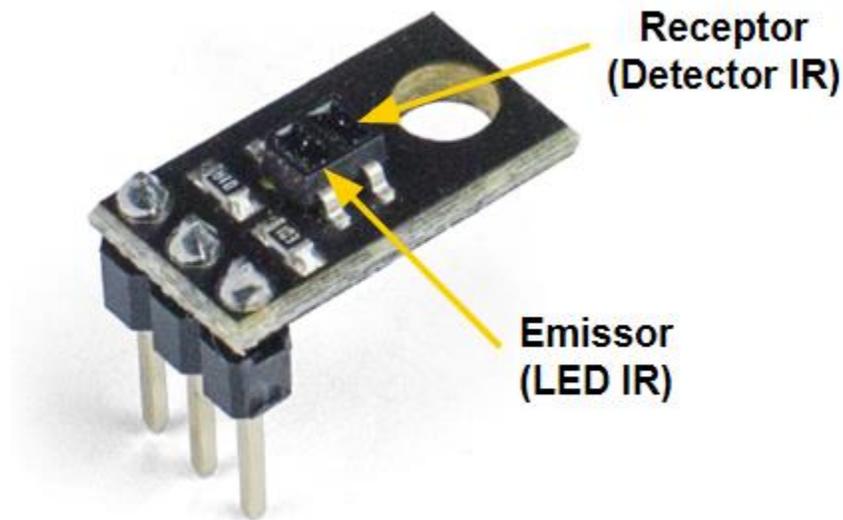
Para a eficácia do sensor, a superfície em contraste com a faixa preta deve ser branca.

Para ajustar a sensibilidade ao contraste, o modulo possui um potenciômetro de calibração.

1.3 Funcionamento do sensor infravermelho



Sensor de linha



Os sensores de linha funcionam melhor quando estão perto da superfície abaixo do robô.

O sensor deve estar cerca de 3mm acima do solo.

Essa é uma ótima distância para que o emissor IR ilumine a superfície e realize a leitura da luz refletida.

Agenda de hoje

Estrutura da Linguagem do Arduino (linguagem C)

As portas de E/S do Arduino e suas funções em C

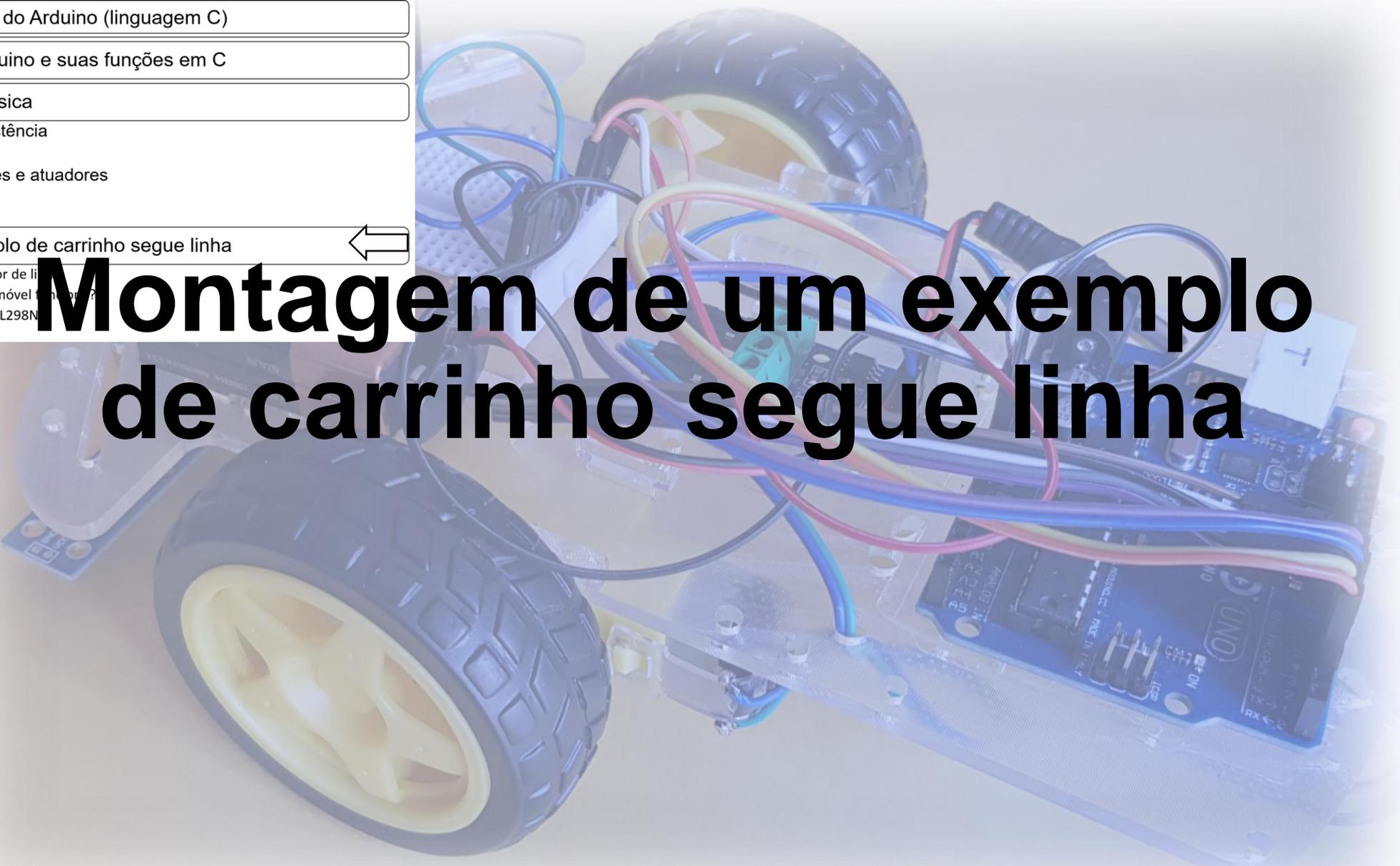
Conceitos básicos de Física

- Tensão, corrente & resistência
- Componentes básicos
- Conhecendo os sensores e atuadores
 - Motores e Ponte H
 - Sensores de Linha

Montagem de um exemplo de carrinho segue linha ←

- A importância do robô seguidor de linha
- Como o carrinho de um robô móvel funciona?
- Motor DC com Driver Ponte H L298N
- Sensor segue linha TCRT5000

Montagem de um exemplo de carrinho segue linha



Tópicos

Introdução ao
Robô seguidor
de linha

A importância do robô seguidor de linha
Como o carrinho de um robô móvel funciona?

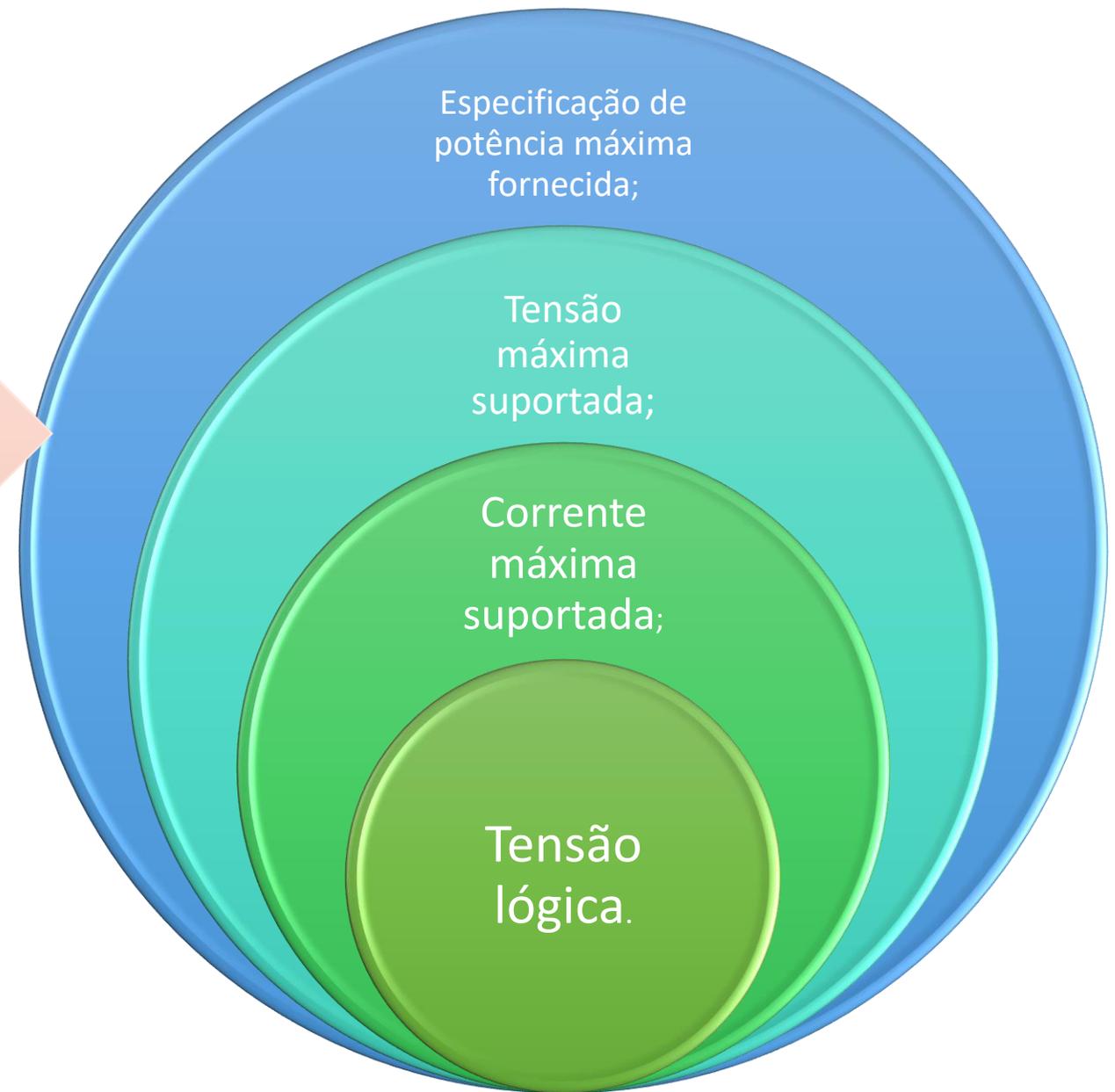
Prática –
Desenvolvendo
o robô
seguidor de
linha

Motor DC com Driver Ponte H L298N
Sensor segue linha
Colocando para funcionar

Motor DC com Driver Ponte H L298N

Ponte H

Existem varias opções disponíveis no mercado, com tamanhos e especificações diferentes. Algumas especificações são importantes ao escolher seu módulo, são elas:

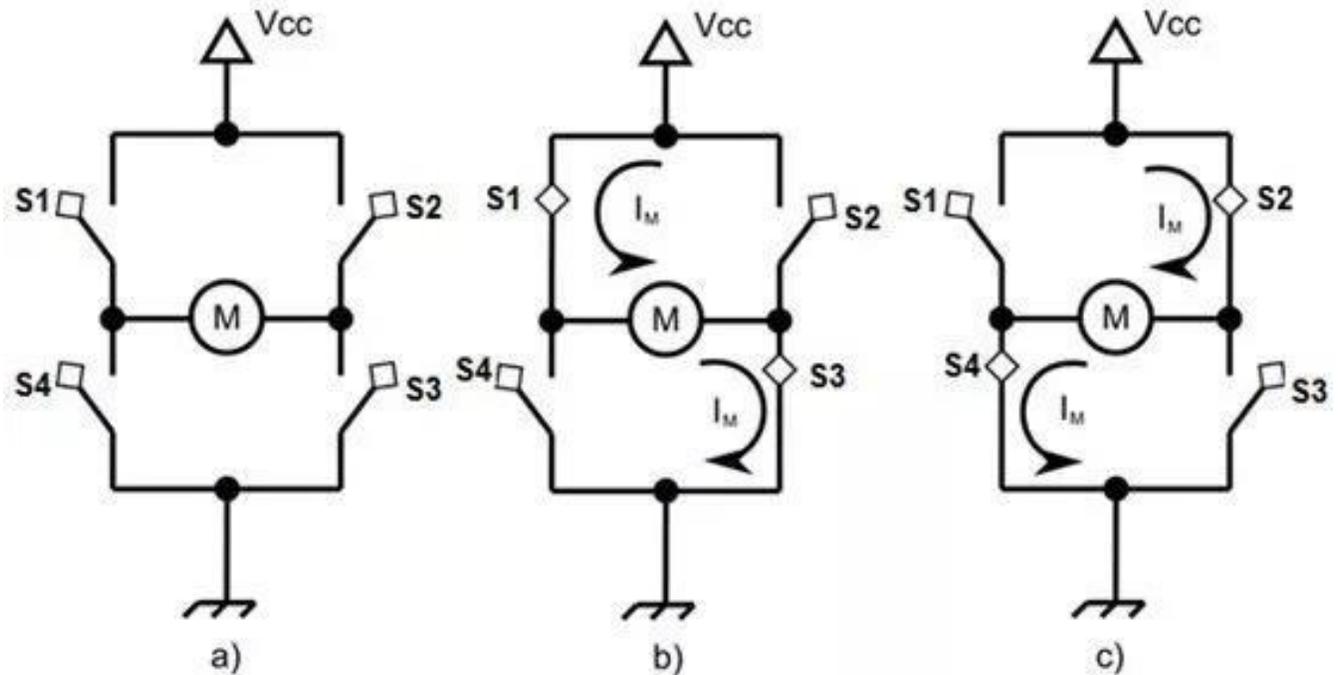


Mas como funciona a Ponte H? Porque este nome?

As pontes H em possuem este nome devido ao formato que é montado o circuito, semelhante a letra H.

O circuito utiliza quatro chaves (S1, S2, S3 e S4) que são acionadas de forma alternada, ou seja, (S1-S3) ou (S2-S4).

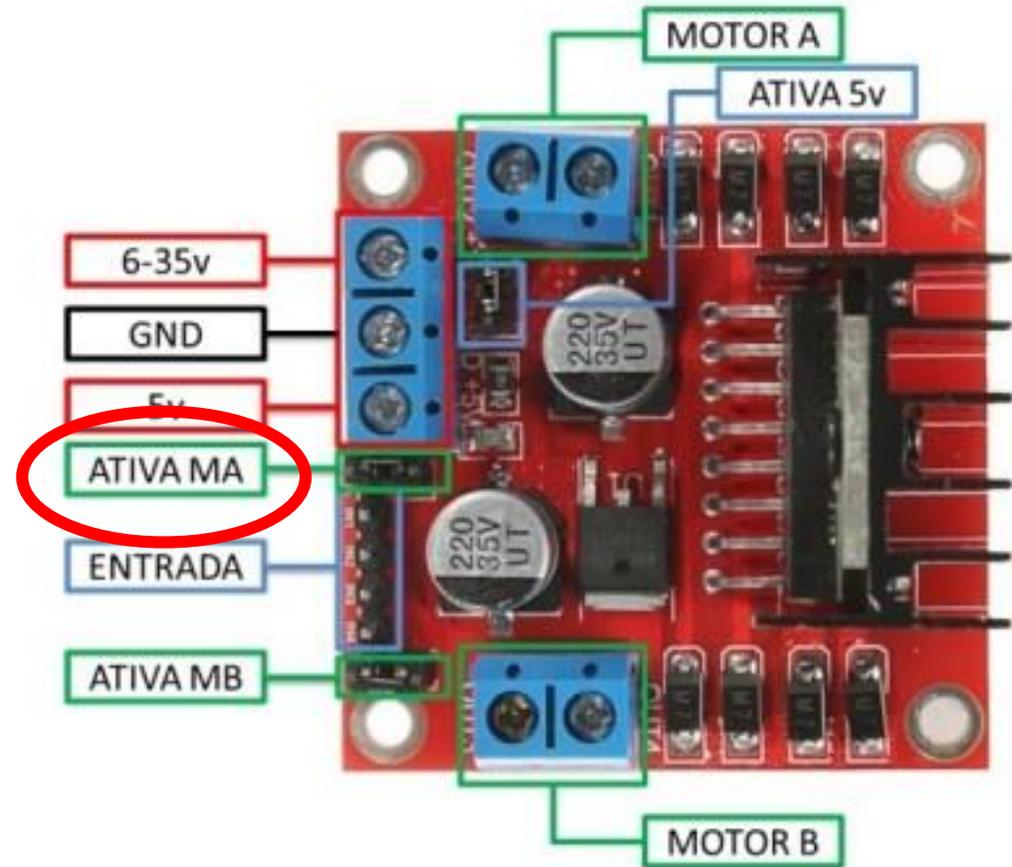
Dependendo da configuração entre as chaves teremos a corrente percorrendo o motor hora por um sentido, hora por outro.



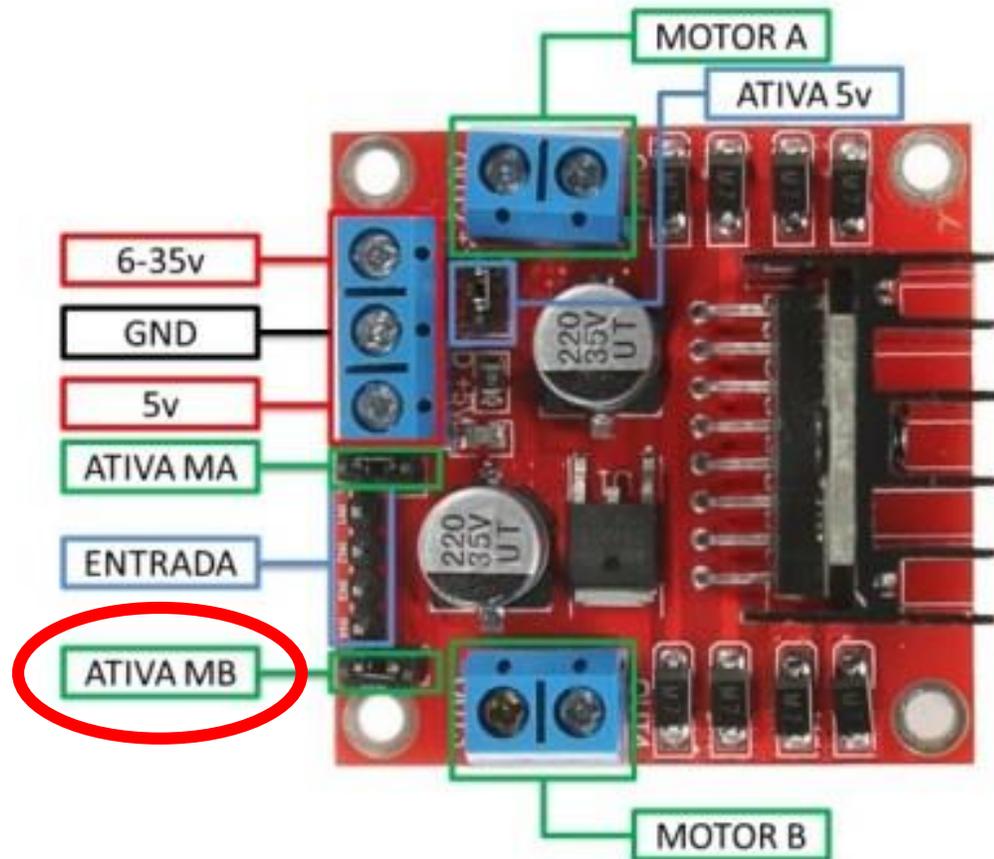
Ativa MA:

Quando jumpeado aciona o motor A com velocidade máxima.

Para controlar a velocidade do motor A basta remover o jumper e alimentar o pino com uma tensão entre 0 e 5v, onde 0V é a velocidade mínima (parado) e 5V a velocidade máxima.



Ativa MB



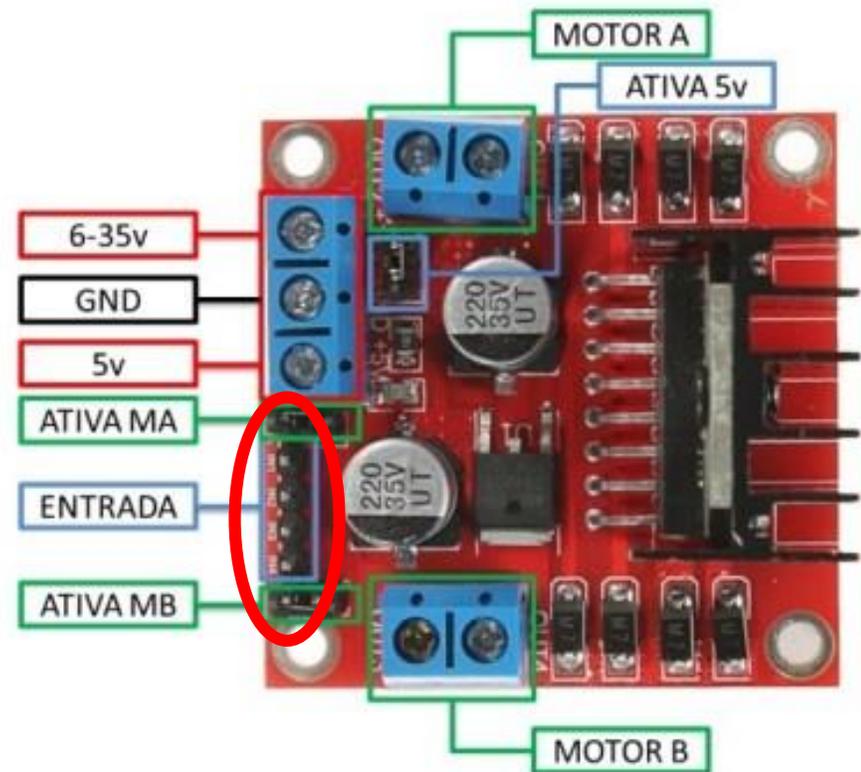
Quando jumpeado aciona o motor B com velocidade máxima.

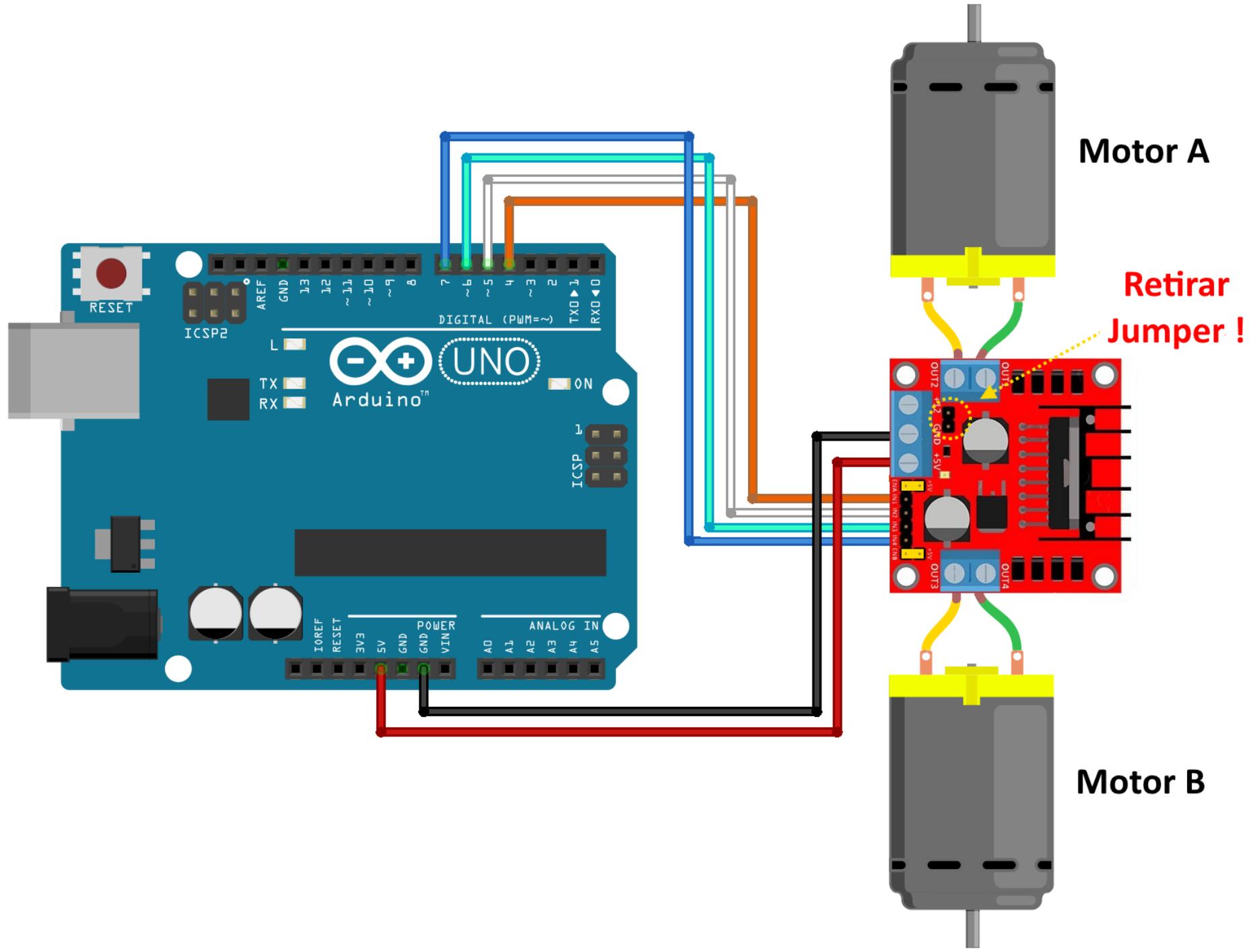
Para controlar a velocidade do motor B basta remover o jumper e alimentar o pino com uma tensão entre 0 e 5v, onde 0V é a velocidade mínima (parado) e 5V a velocidade máxima.

IN 1, 2, 3, 4

IN1 e IN2: são utilizados para controlar o sentido do motor A;

IN3 e IN4: são utilizados para controlar o sentido do motor B;





Código teste 1

Programando Ponte H com Arduino: teste

Teste o seu módulo carregando o programa abaixo, que vai servir para os mostrar o funcionamento do motor com ponte H.

O programa gira o motor A no sentido horário, depois desliga esse motor e gira o motor B no mesmo sentido. Depois, repete esse procedimento no sentido anti-horário.

Definições de variáveis:

```
//Programa : Controle 2 motores DC usando Ponte H L298N - Versão 1.0  
//Dando voltas para esquerda e para direita  
//Definicoes pinos Arduino ligados a entrada da Ponte H  
int IN1 = 4;  
int IN2 = 5;  
int IN3 = 6;  
int IN4 = 7;
```

Código setup()

```
void setup()
{
  //Define os pinos como saida
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
}
```

Código loop() 1/2

```
void loop() {  
  //Gira o Motor A no sentido horario  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, LOW);  
    delay(2000);  
  //Para o motor A  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, HIGH);  
    delay(500);  
}
```

```
//Gira o Motor B no sentido horario  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, LOW);  
    delay(2000);  
//Para o motor B  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, HIGH);  
    delay(500);
```

Código loop() 2/2

```
//Gira o Motor A no sentido anti-  
horario
```

```
digitalWrite(IN1, LOW);  
digitalWrite(IN2, HIGH);  
delay(2000);
```

```
//Para o motor A
```

```
digitalWrite(IN1, HIGH);  
digitalWrite(IN2, HIGH);  
delay(500);
```

```
//Gira o Motor B no sentido horario
```

```
digitalWrite(IN3, HIGH);  
digitalWrite(IN4, LOW);  
delay(2000);
```

```
//Para o motor B
```

```
digitalWrite(IN3, HIGH);  
digitalWrite(IN4, HIGH);  
delay(500);
```

Código teste 2

Programando Ponte H com Arduino: teste

Outro teste que serve para os mostrar o funcionamento do motor com ponte H.

O programa faz o carrinho avançar por 2 segundos, para e depois retrocede por 2 segundos. Depois, repete esse procedimento no sentido anti-horário.

Definições de variáveis:

```
//Programa : Controle 2 motores DC usando Ponte H L298N - Versão 2.0  
//Para frente e para trás  
//Definicoes pinos Arduino ligados a entrada da Ponte H  
int IN1 = 4;  
int IN2 = 5;  
int IN3 = 6;  
int IN4 = 7;
```

Código setup()

```
void setup()
{
  //Define os pinos como saida
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
}
```

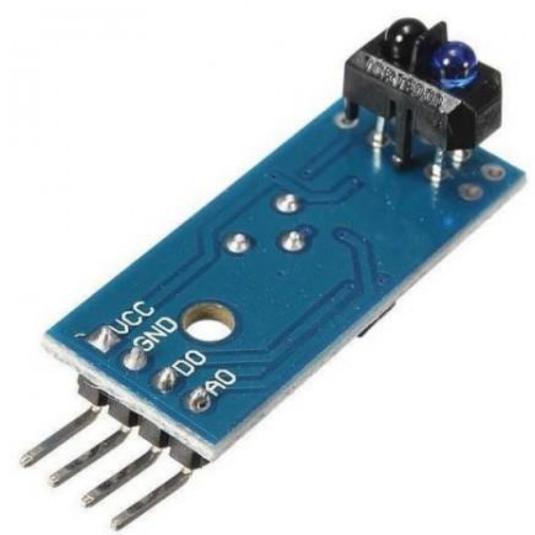
Código loop() 1/2

```
void loop() {  
  //Gira o Motor A e B para frente  
  (no sentido horario)  
  digitalWrite(IN1, HIGH);  
  digitalWrite(IN2, LOW);  
  digitalWrite(IN3, HIGH);  
  digitalWrite(IN4, LOW);  
  delay(2000);  
}
```

```
//Para o motor A e B  
  digitalWrite(IN1, HIGH);  
  digitalWrite(IN2, HIGH);  
  digitalWrite(IN3, HIGH);  
  digitalWrite(IN4, HIGH);  
  delay(2000);
```

Código loop() 2/2

```
//Gira o Motor A e B no sentido anti-horario
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  delay(2000);
}
```

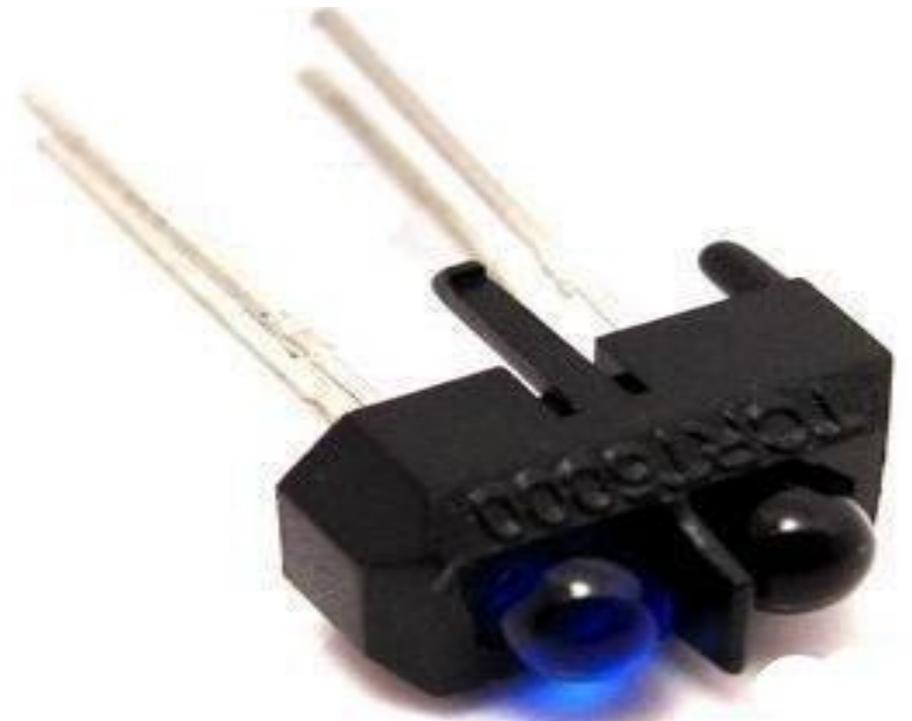


Sensor segue linha TCRT5000

Infravermelho

Especificações e características:

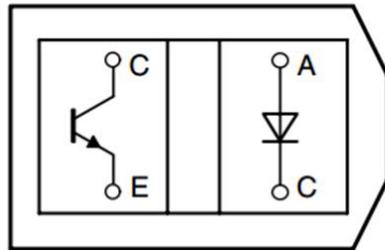
- Tensão de operação: 5VDC
- Corrente máxima: 60 mA
- Comprimento de onda: 950nm
- Distância de detecção: 1mm ~ 8 mm;
- Potenciômetro de ajuste de sensibilidade;



Sensor infravermelho TCRT5000



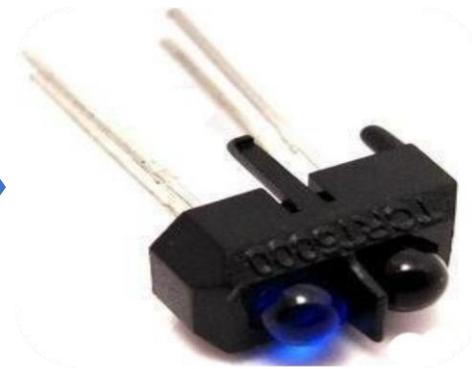
19156_2



Este sensor é composto basicamente por um LED emissor de infravermelho (Luz que nesta frequência [Hz] não é visível a olho nu) e um fototransistor responsável por detectar a luz infravermelha.

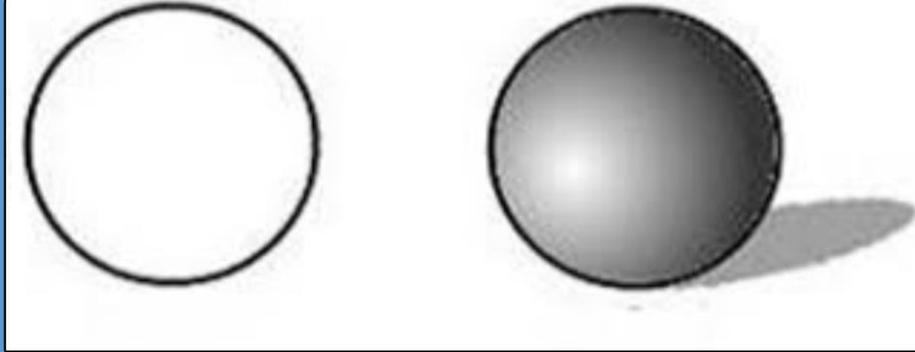
Sensor infravermelho TCRT5000

Nesse módulo, o sensor infravermelho TCRT5000 emite continuamente luz infravermelha, através de seu diodo emissor, quando a luz não é refletida ou é refletida de volta, mas a intensidade não é suficientemente forte, o fototransistor não conduz (desligado), dessa forma, a saída do módulo é baixo.



Se existir objetos na área de detecção, e a intensidade dos raios infravermelhos refletidos é forte o suficiente para saturar o fototransistor, a saída do módulo é alta.

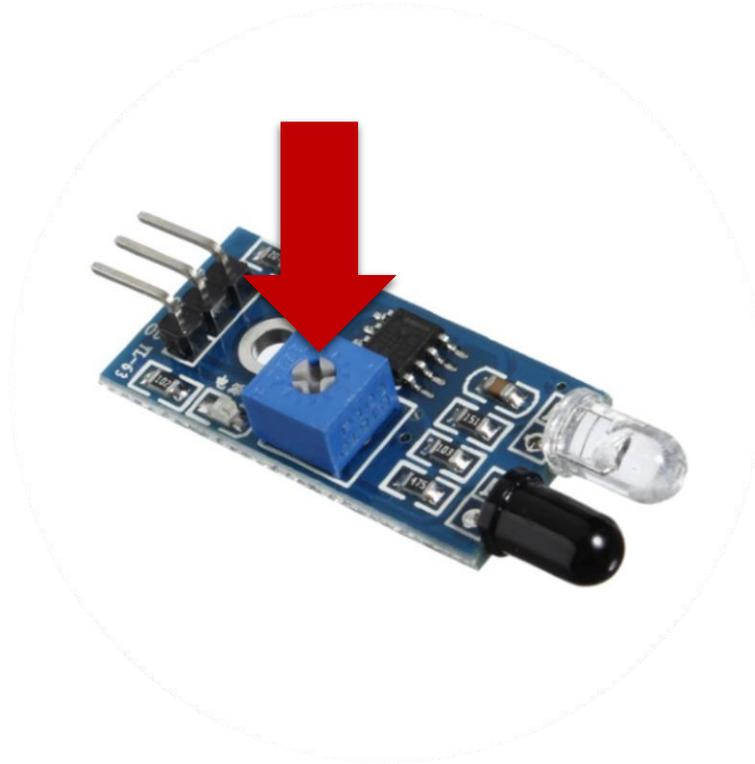
Sensor infravermelho TCRT5000



Com esse tipo de sensor, podemos detectar se uma superfície é escura ou clara, já que superfícies escuras quase não refletem a luz e as claras refletem quase toda a luz incidente.

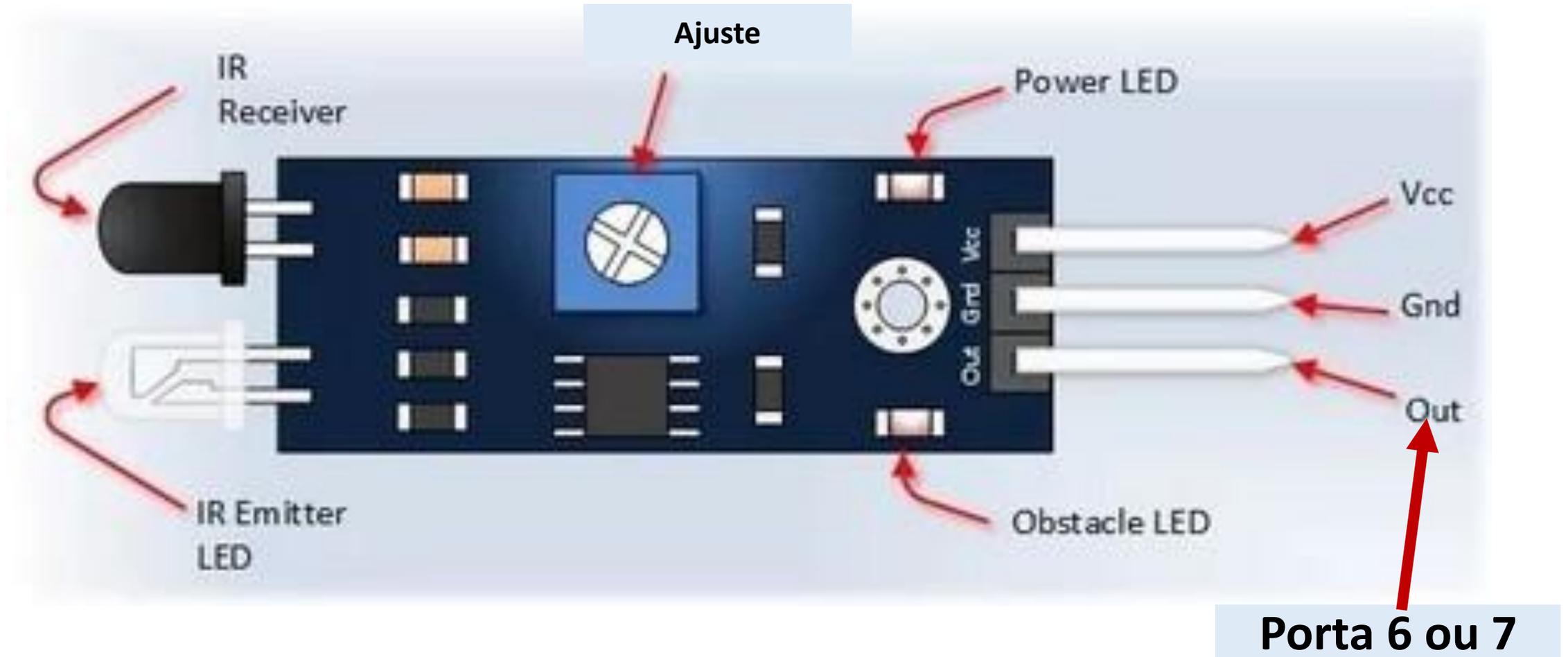


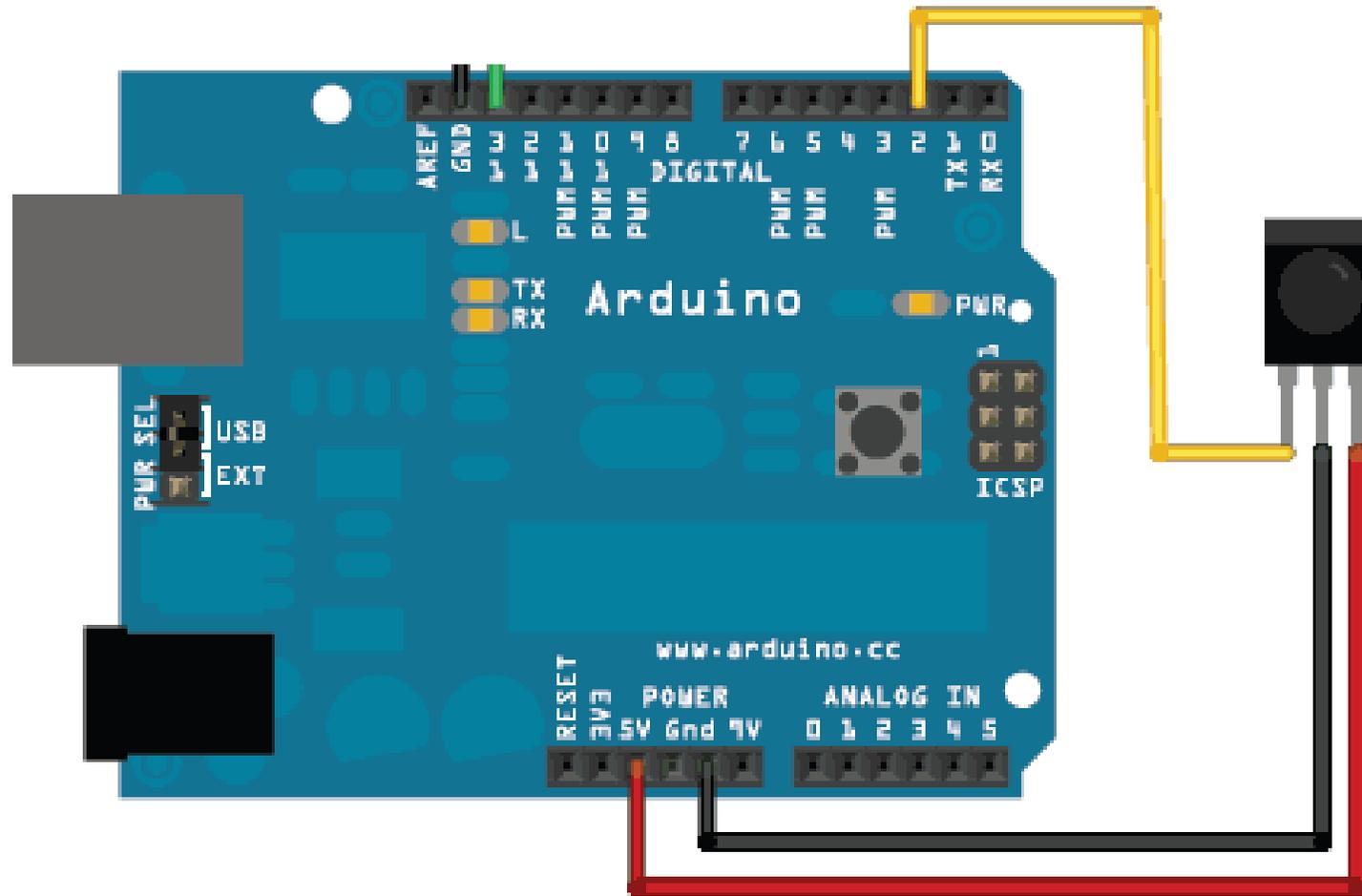
Sensor infravermelho TCRT5000



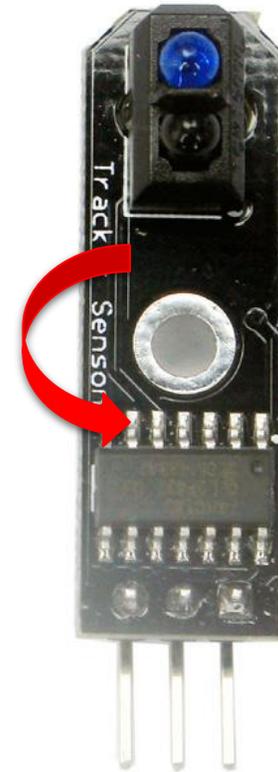
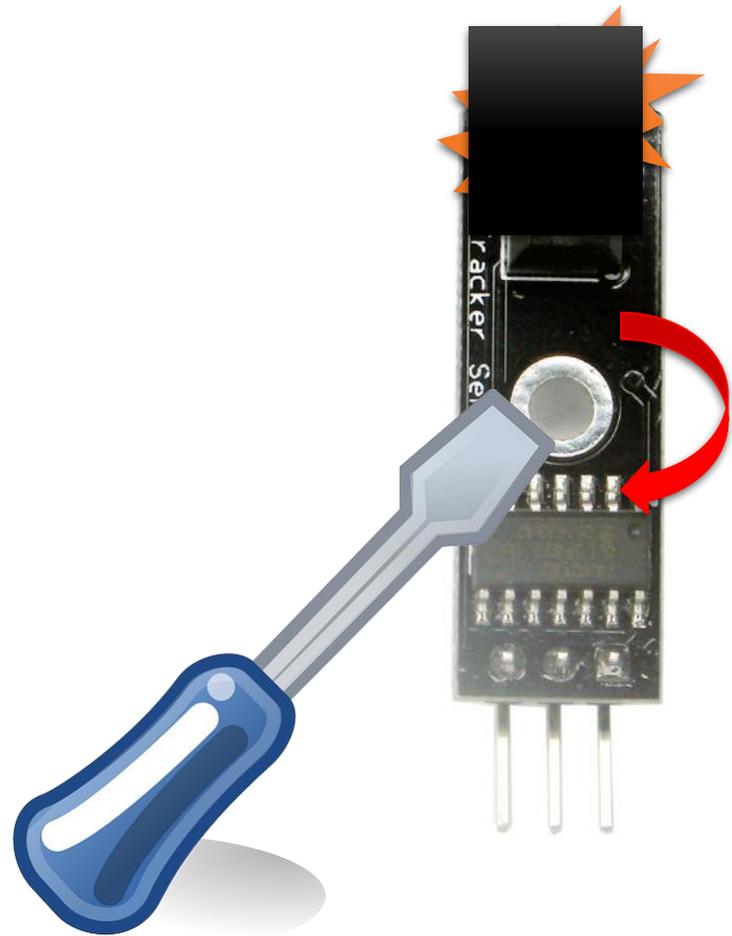
Este módulo conta ainda com um potenciômetro para ajuste da sensibilidade do sensor, não há a necessidade de implementar este ajuste no software do microcontrolador.

Como ligar





Calibragem



Código teste

Programando o sensor infravermelho TCRT5000 Arduino: teste

Agora vamos testar o funcionamento do sensor infravermelho.

Devemos usar esse programa para o ajuste do sensor através de seu potenciômetro como indicado anteriormente.

Definições de variáveis:

```
//Programa : Controle de dois sensores infravermelhos TCRT5000 -  
//Versão 1.0  
//Definições pinos Arduino ligados ao sensor  
    #define Sensor_direita 6  
    #define Sensor_esquerda 7
```

Código setup()

```
void setup()
{
    Serial.begin(9600);
    pinMode(Sensor_direita, INPUT);
    pinMode(Sensor_esquerda, INPUT);
}
```

Código loop()

```
void loop()
{
  //Leituras dos Sensores
  direita = digitalRead(Sensor_direita);
  esquerda = digitalRead(Sensor_esquerda);
  Serial.print(direita);
  Serial.print(" || ");
  Serial.println(esquerda);
}
```

Referencias

Motor DC com Driver Ponte H L298N - FelipeFlop

- <https://www.filipeflop.com/blog/motor-dc-arduino-ponte-h-l298n/>

Robô seguidor de linha com sensor infravermelho e PWM - Vida de silício)

- <https://portal.vidadesilicio.com.br/robo-seguidor-de-linha-sensor-infravermelho-e-pwm/>

Fim da Aula III