

UNIVERSIDADE
CORPORATIVA
TARZAN

1º Workshop de Robótica e Arduino

Gustavo Carlos

Thobias Santos

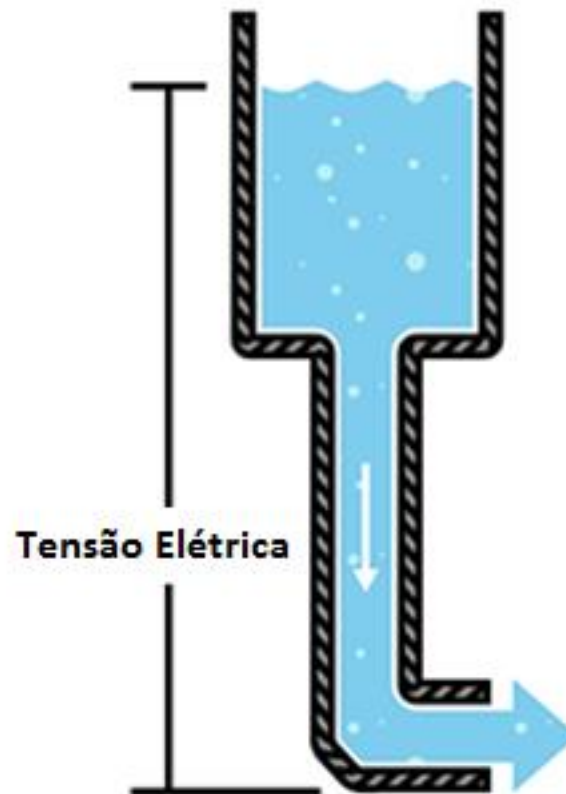
EMENTA DO CURSO

- Noções Básicas da eletrônica;
- Introdução;
- Microcontroladores;
- Ide Arduino e Programação;
- Sensores;
- Drive de potência;
- Montagem Prática do Robô segue faixa.

NOÇÕES BÁSICAS

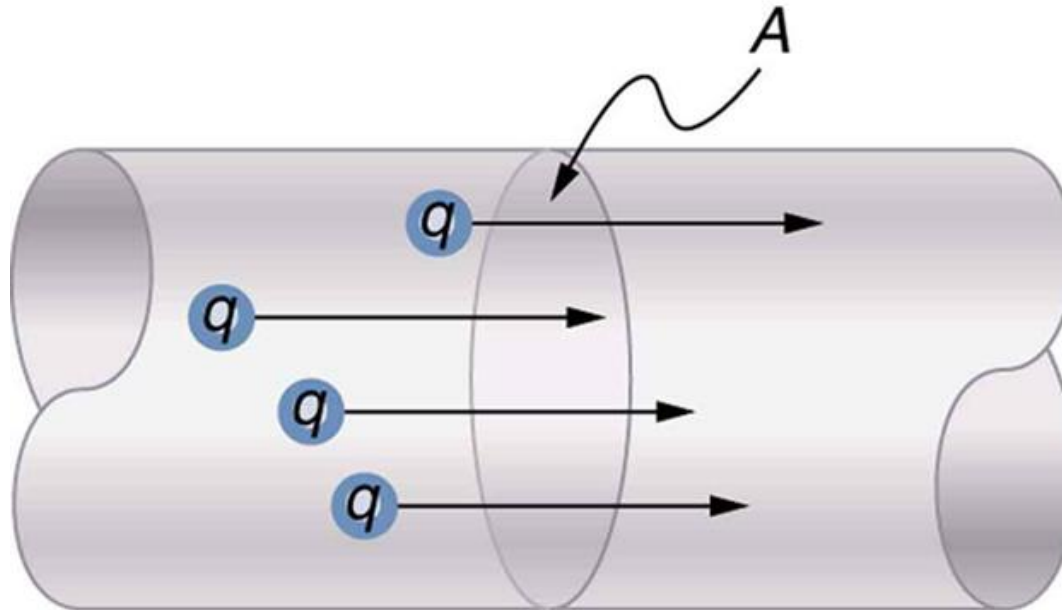
- O que é Tensão elétrica?
- O que é Corrente elétrica?
- O que é Resistência elétrica?

TENSÃO

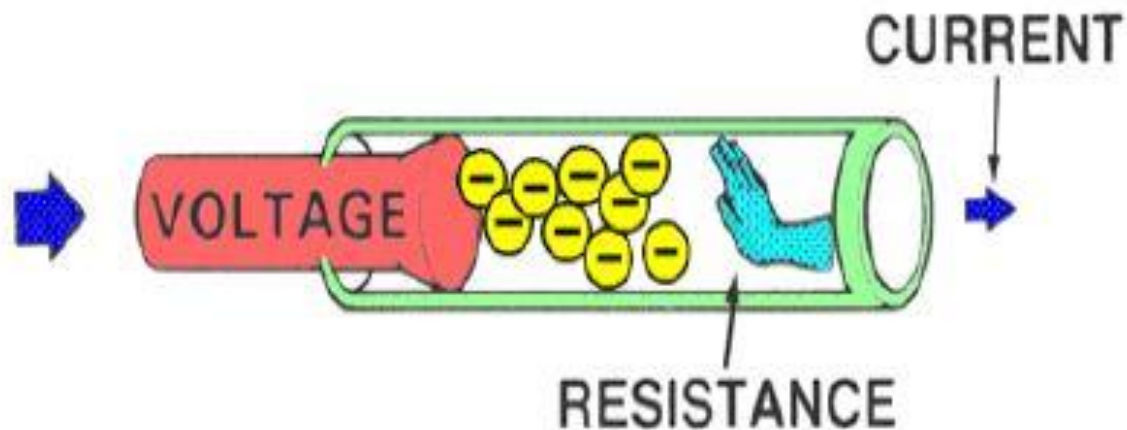


CORRENTE

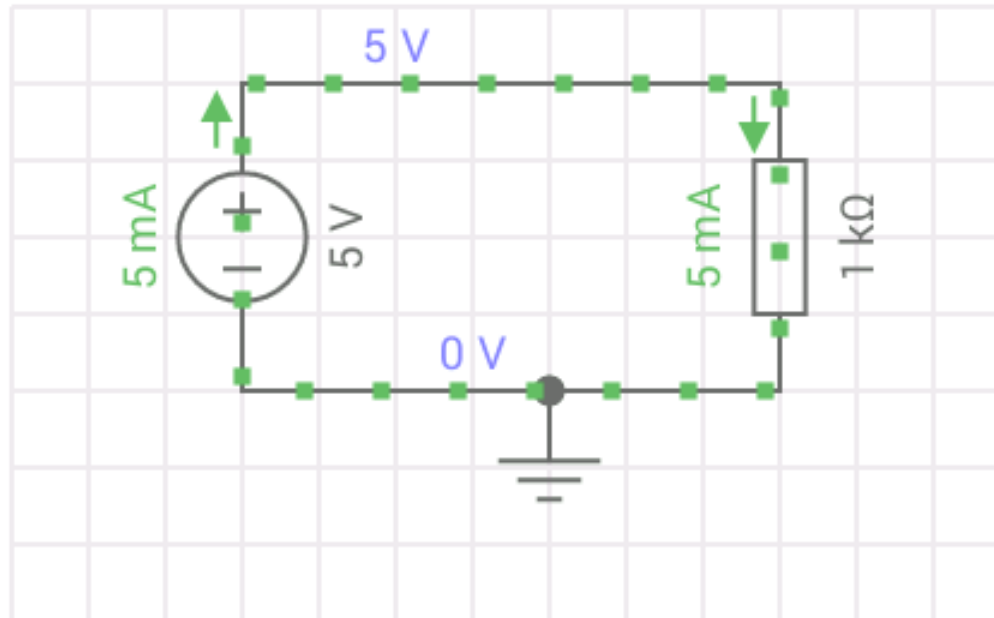
Current = flow of charge



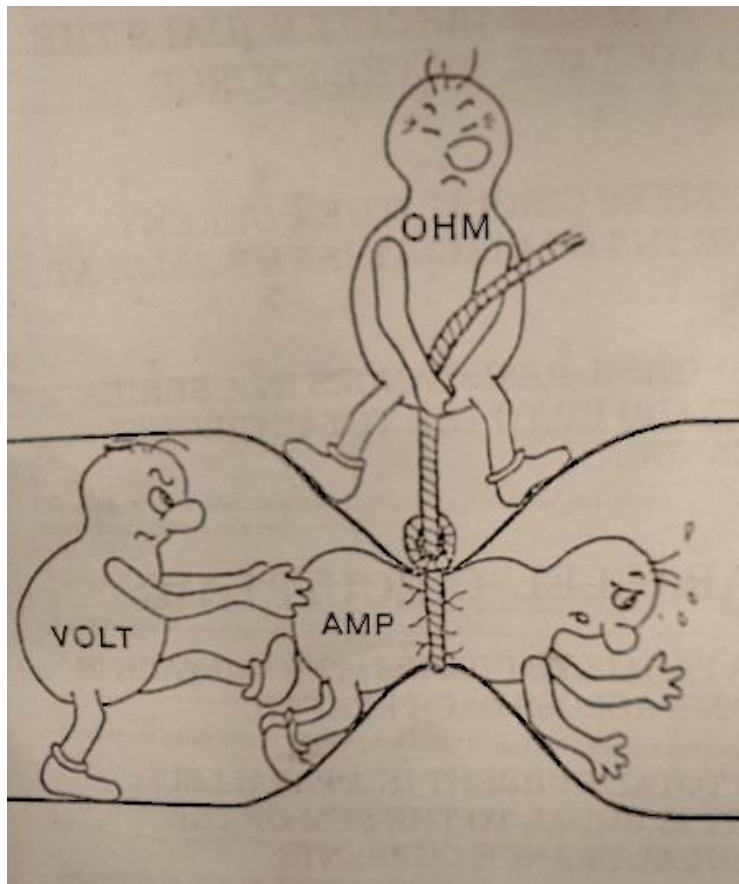
RESISTÊNCIA



LEI DE OHM

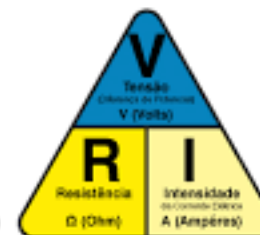


LEI DE OHM



Calcular a Intensidade da Corrente

$$I = \frac{V}{R}$$



Calcular a Tensão

$$V = R \times I$$



Calcular a Resistência

$$R = \frac{V}{I}$$

INTRODUÇÃO

O que é um robô?

INTRODUÇÃO

O que é um robô?

Aparelho capaz de agir de maneira automática numa dada função.

INTRODUÇÃO

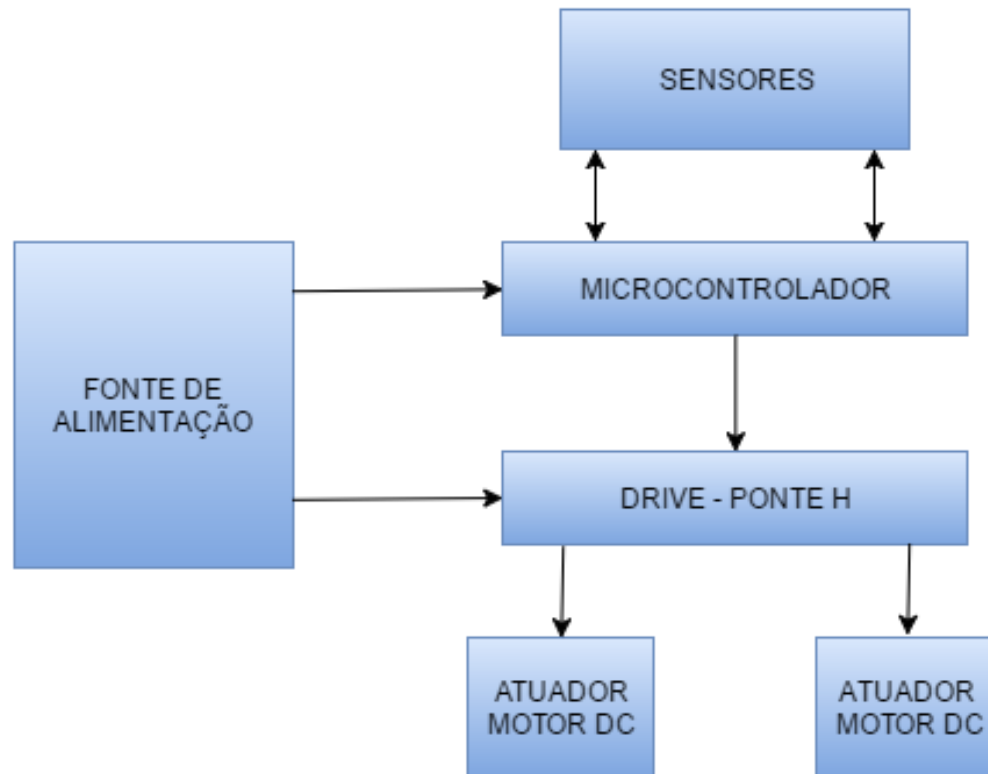
O que preciso para o meu Robô?

INTRODUÇÃO

LISTA DE COMPONENTES:

- Microcontrolador;
- Sensores;
- Drive de Motor DC (“Ponte H”);
- Bateria ;
- Motores.

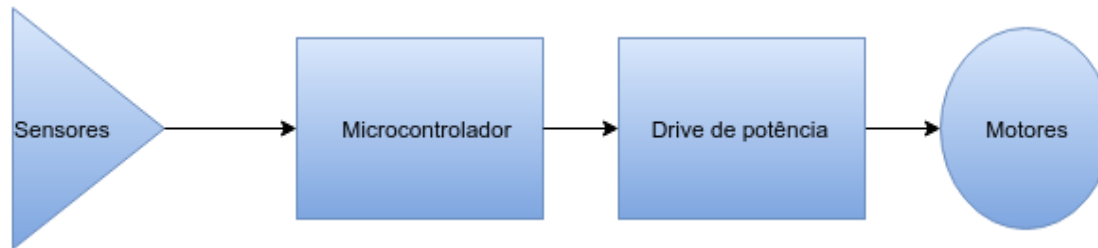
DIAGRAMA DOS COMPONENTES



Montagem

Como é a comunicação destes componentes?

DIAGRAMA DE BLOCOS



MICROCONTROLADOR



MICROCONTROLADOR

O que é um Microcontrolador?

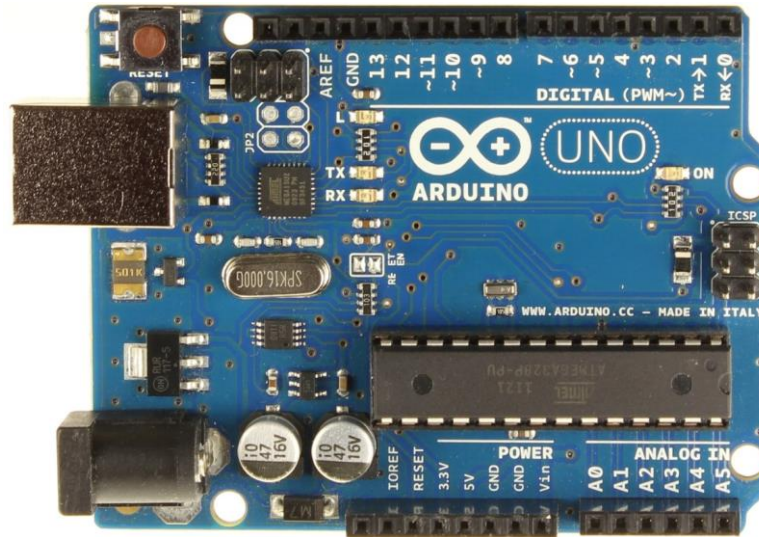
- Micro computador em um único circuito integrado;
- Contém um núcleo de processador;
- Contém periféricos de entrada e saída.

MICROCONTROLADORES

Plataformas disponíveis

MICROCONTROLADORES

- Arduino



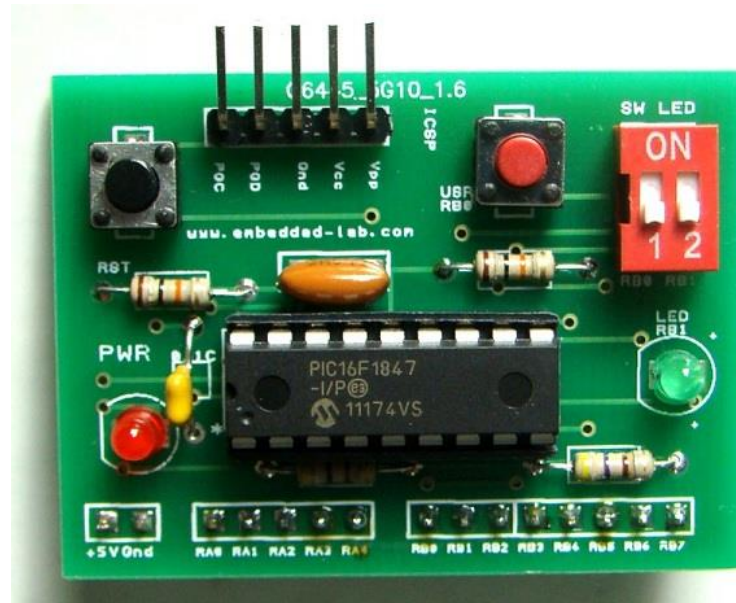
MICROCONTROLADORES

- MSP430 LaunchPad



MICROCONTROLADORES

- Família PIC



MONTAREMOS NOSSO ROBÔ COM ARDUINO

IDENTIFICANDO A PLATAFORMA

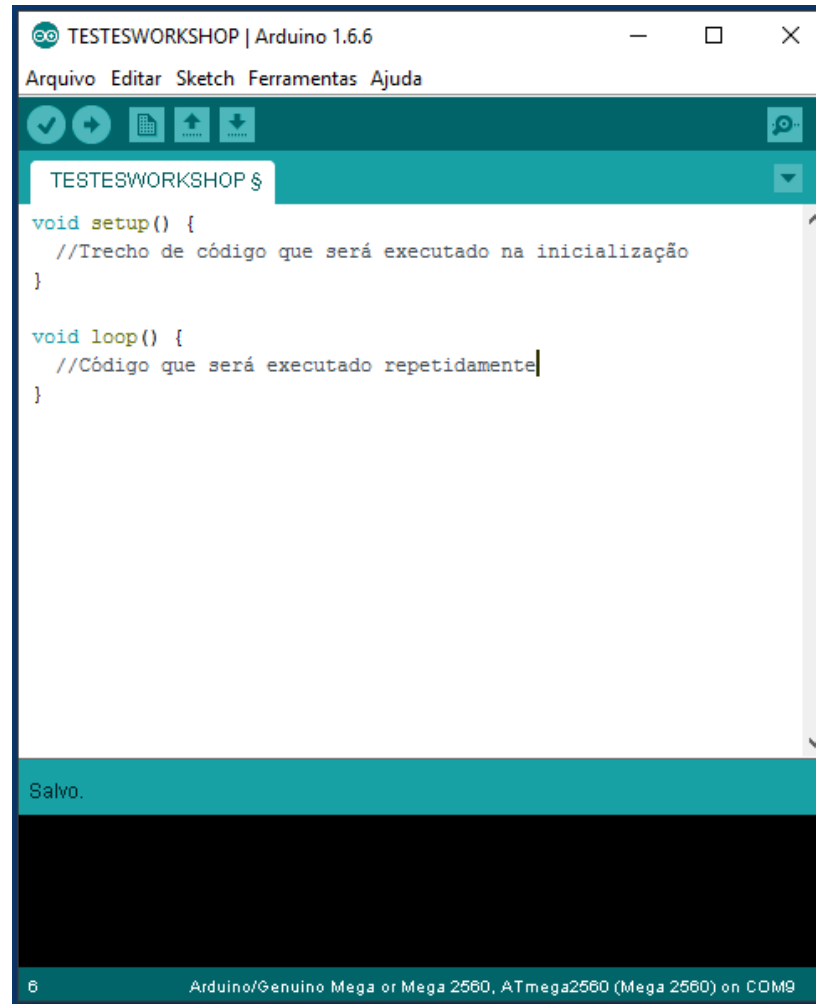
Arduino UNO



IDE ARDUINO



IDE ARDUINO

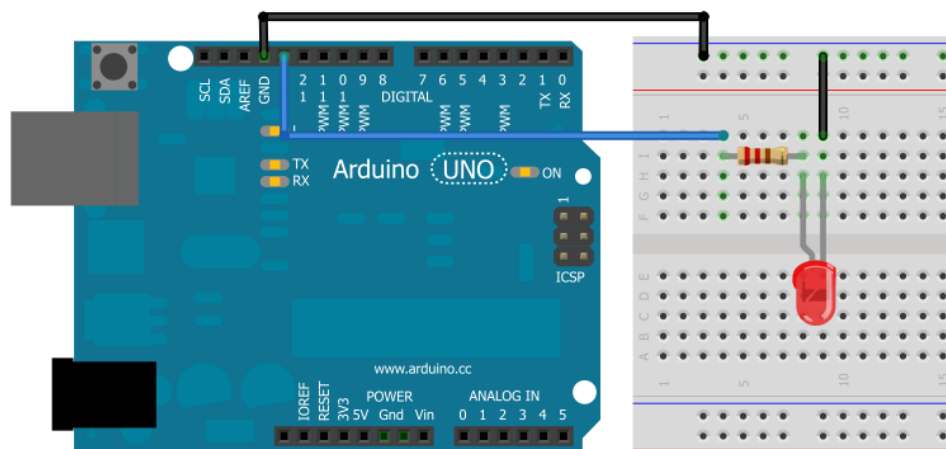


PERIFÉRICOS PROGRAMÁVEIS

Portas de entradas e saídas

- Cada um desses pinos pode ser configurado individualmente para funcionar como entrada e saída.

Exemplo de Portas de E/S



Exemplo de programação

Configurando as portas E/S

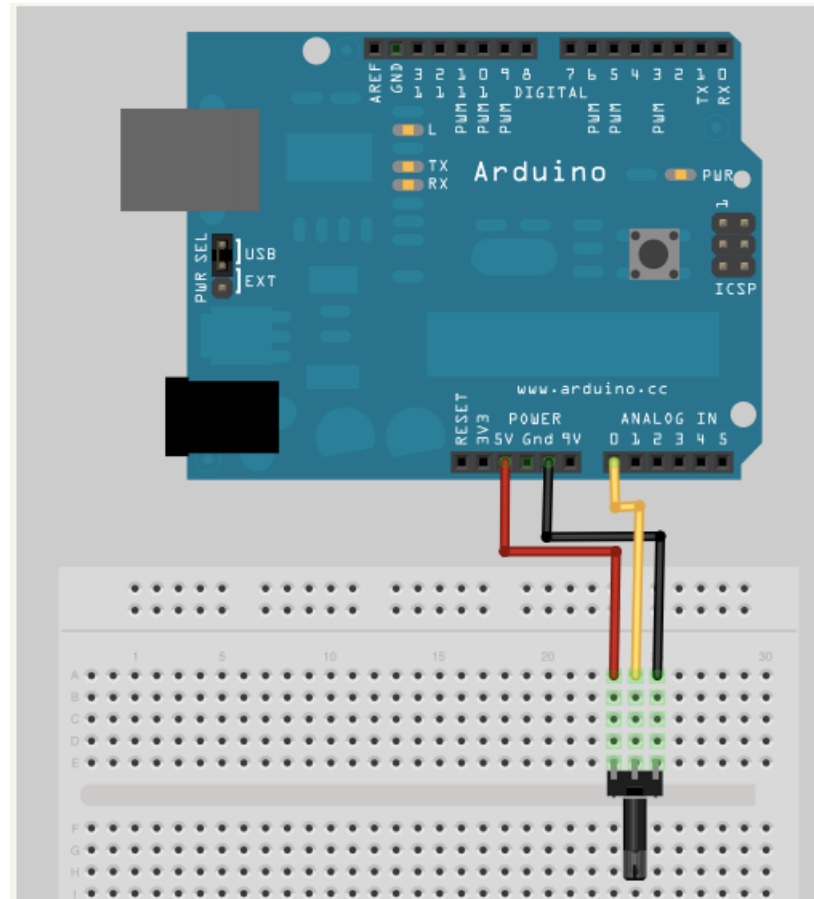
```
void setup() {  
    // inicializa o pino digital 13 como saída.  
    pinMode(13, OUTPUT);  
}  
// A função loop roda infinitamente.  
void loop() {  
    digitalWrite(13, HIGH); // Deixa o pino de saída em nível alto  
    delay(1000);           // Espera por um Segundo  
    digitalWrite(13, LOW);  // Deixa o pino de saída em nível baixo  
    delay(1000);           // Espera por um Segundo  
}
```

PERIFÉRICOS PROGRAMÁVEIS

Conversor AD (Analógico para Digital)

- Precisamos que o processador entenda digitalmente os sinais elétricos que estão chegando na porta entrada;
- Converte uma unidade física (normalmente Voltagem) em um número digital (Binário);
- Um conversor de 10 bits, faz uma leitura de números inteiros de 0 até 1023;
- Isso significa que ele irá mapear tensões de entrada entre 0 e 5 volts em valores inteiros entre 0 e 1.023.

Exemplo de Conversor AD



Exemplo de programação

```
int analogPin = 3; // Terminal do meio do potenciômetro ligado ao analógico 3
int val = 0;       // Variável para armazenar o valor lido
```

```
void setup(){
    Serial.begin(9600); // Configura a porta serial
}
void loop(){
    val = analogRead(analogPin); // Lê o valor do AD no pino de entrada
    Serial.println(val);         // Envia valor lido para a serial
}
```


PERIFÉRICOS PROGRAMÁVEIS

Modulação por largura de pulso (pulse width modulation) PWM

- É um controle digital usado para criar uma onda quadrada, onde sinal fica alternado entre on e off;
- Este padrão de on-off, é um simples modo de alternar a tensão entre 5V e 0V por um determinado período, isto que define a largura do pulso;
- Portanto modulando a largura do pulso, o resultado é, como se o sinal é uma tensão constante que varia entre 0 e 5V;

Pulse Width Modulation

0% Duty Cycle - `analogWrite(0)`



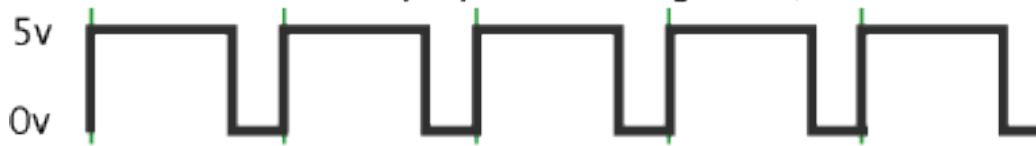
25% Duty Cycle - `analogWrite(64)`



50% Duty Cycle - `analogWrite(127)`



75% Duty Cycle - `analogWrite(191)`



100% Duty Cycle - `analogWrite(255)`



Demo

Exemplo de programação

```
int Ledpin=9;      // Pino para a conexão do LED
int analogpin=A0; // Canal analógico que será lido
int value=0;       // Variável para armazenar o valor lido do AD
int x;             // Variável que indicara o valor do duty cycle PWM

void setup() {
  pinMode(Ledpin, OUTPUT); // coloca pino do Led como saída
}
void loop() {
  value=analogRead(analogpin); // realiza a leitura do AD
  x=map(value,0,1023,0,255); // realiza a conversão do AD para PWM
  analogWrite(Ledpin,x); // atualiza duty cycle no pino do LED
  delay(100); // aguarda 100 milissegundos
}
```

SENSORES

Sensor Infra vermelho (IR), Exemplo o modulo P-12 da GBK.



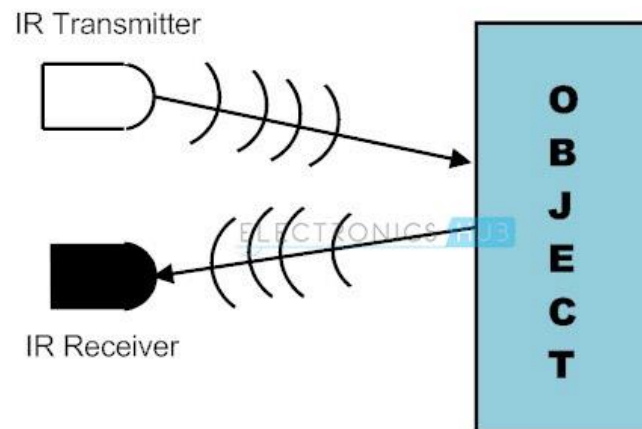
SENSORES

Características

- LED Emissor Infravermelho emite o sinal, e o LED receptor recebe os sinais devolvidos após bater em algum objeto;
- São utilizados em sensores de faixa, por causa de sua reflexão em cores;
- São utilizados em robôs seguidores de linhas e também impressoras.

SENSORES

Princípio de funcionamento do sensor de obstáculo (IR)



SENSORES

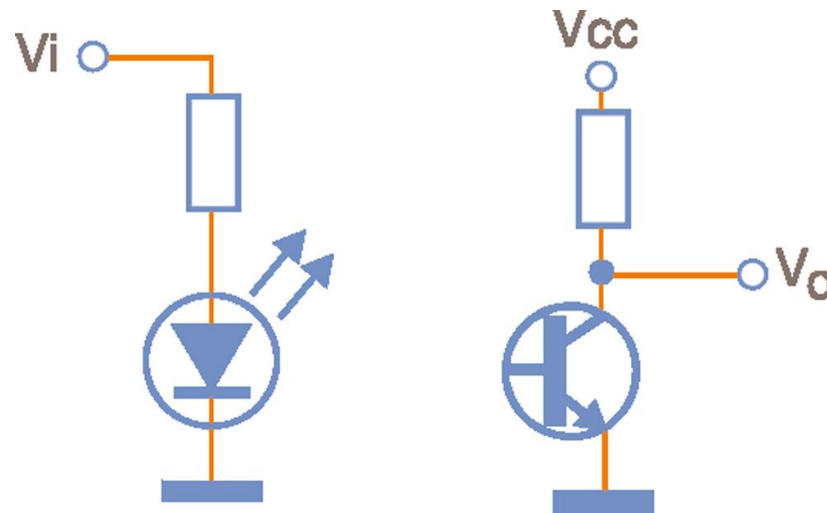
Exemplo de Ligação com modulo P-12

Arduino – Modulo

GND	GND
PINO A0	A0
VCC	VCC

SENSORES

Diagrama elétrico sensor Infra vermelho (IR)



Exemplo de programação

```
int sensor;    //Declarando a variável para armazenar valores do AD
int Led = 13;  //Declarando a variável para o led no pino 13
void setup() {
    Serial.begin(9600); // Inicializa comunicação serial a 9600 bits por segundo
    pinMode(Led, OUTPUT); // Configurando o led como saída
}
void loop() {
    sensor= analogRead(A0); //Realiza a leitura do AD no canal A0
    if(sensor<500){ //Se o valor da leitura do pino A0 for menor que 500
        digitalWrite(Led,HIGH); //acende o Led
    }
    else{ //Caso contrario
        digitalWrite(Led,LOW); //apaga o Led
    }
}
```

SENSORES

Sensor Ultrassônico



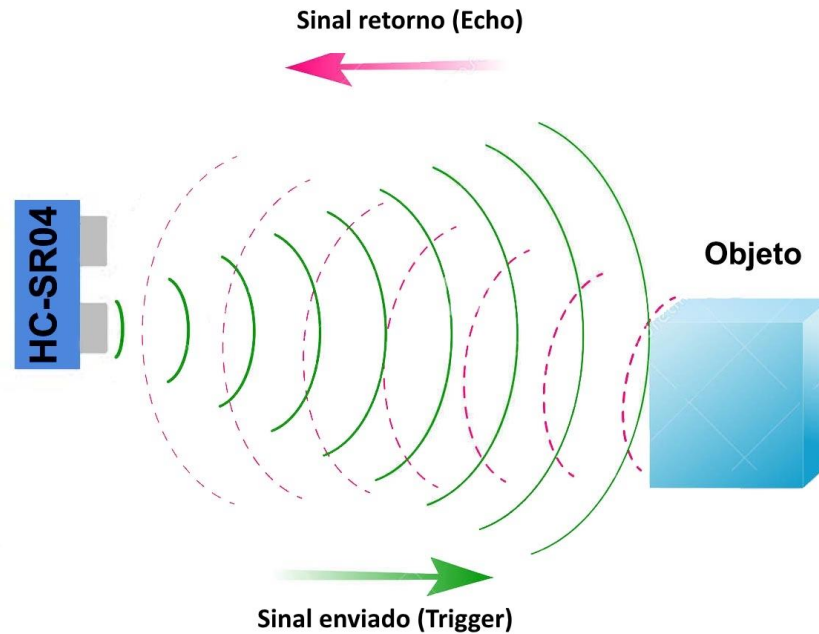
SENSORES

Características

- São similares com a tecnologia do sonar;
- Onde o sensor emissor gera um pulso sonoro ultrassônico ao detectar um objeto reflete e irá retorna até o sensor receptor, quando o sensor intercepta a onda calculará o período de frequência, e através do tempo de emissão e recepção se obtém o distancia do objeto;
- Frequência 40Khz , superior ao limite Maximo do ser humano 20khz.

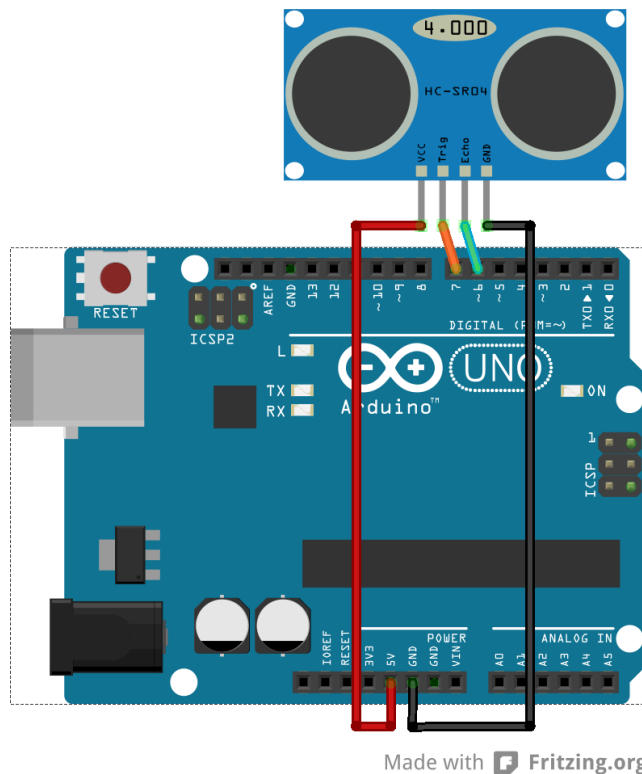
SENSORES

Princípio de funcionamento



SENSORES

Diagrama de Conexão



Exemplo de programação

```
// Definindo o pinos de entrada e saída
#define echoPin 7
#define trigPin 6
#define LEDPin 13
long duration, cm;

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

Continuação do programa...

```
void loop(){  
  //sequencia de nível alto e baixo e gerar a frequência do sinal sonoro  
  digitalWrite(trigPin, LOW);  
  delayMicroseconds(2);  
  digitalWrite(trigPin, HIGH);  
  delayMicroseconds(5);  
  digitalWrite(trigPin, LOW);  
  //Faz a leitura do tempo do pulso no pino echo  
  duration = pulseIn(echoPin, HIGH);  
  //calcula a distancia  
  cm = microsecondsToCentimeters(duration);  
  //Envia valores para a serial  
  Serial.print ("Milisegundos: ");  
  Serial.print(duration);  
  Serial.print(" Distancia estimada: ");  
  Serial.print(cm);  Serial.println(" cm");  
  delay(1000);  
}
```


Continuação do programa...

```
long microsecondsToCentimeters(long microseconds){  
    // A velocidade do som é 340 m/s  
    // 29 microsegundos por centrimetro.  
    // o Sinal vai e volta  
    // a distancia é a metade da distancia percorrida  
    return microseconds / 29 / 2 ;  
}
```

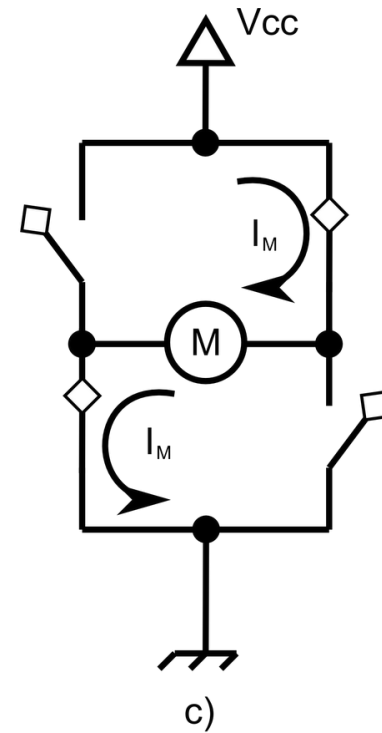
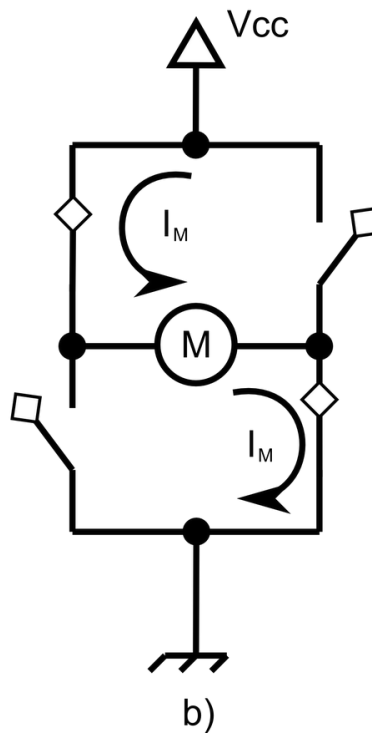
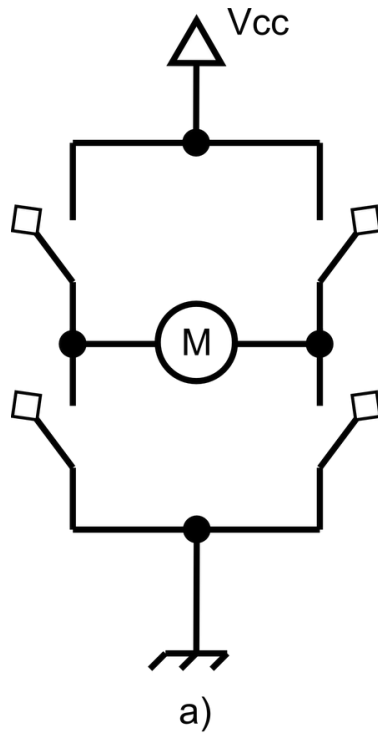
Drive de motor

- É um circuito que faz controle de velocidade , e inversão de sentidos do Motor;
- Controlar um motor requer correntes mais altas do que as que o Arduino pode produzir ;
- É também uma forma de proteger o Microcontrolador.

O que é? Qual utilidade?

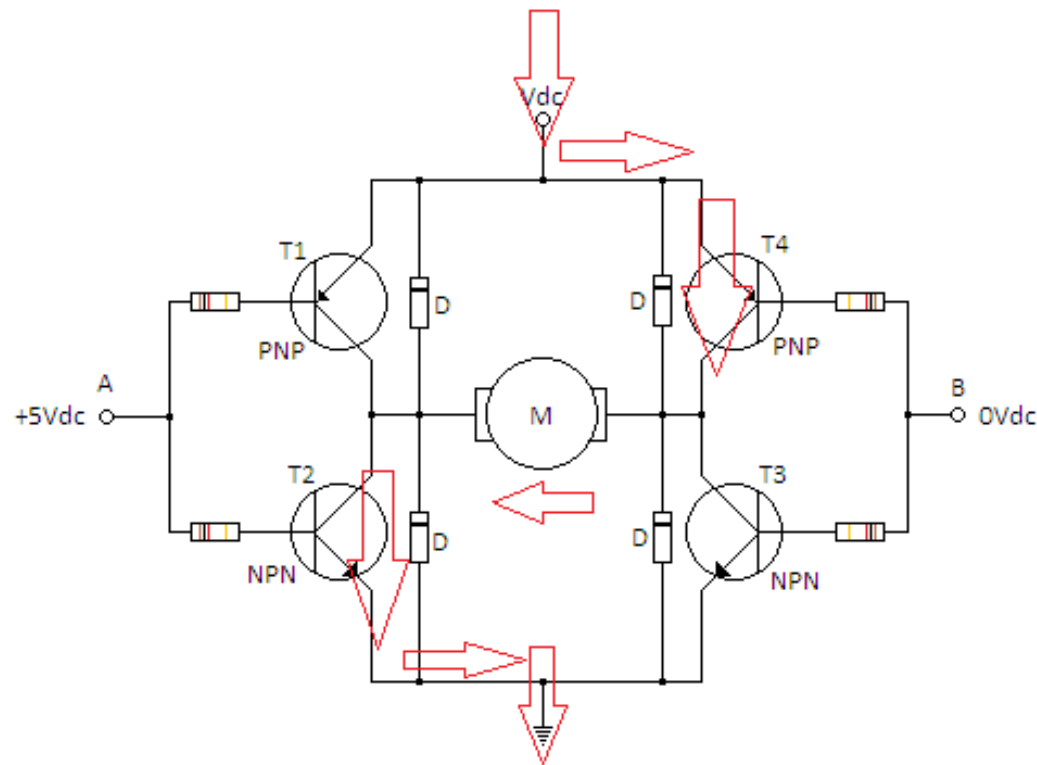
Drive de motor

É a famosa Ponte H.

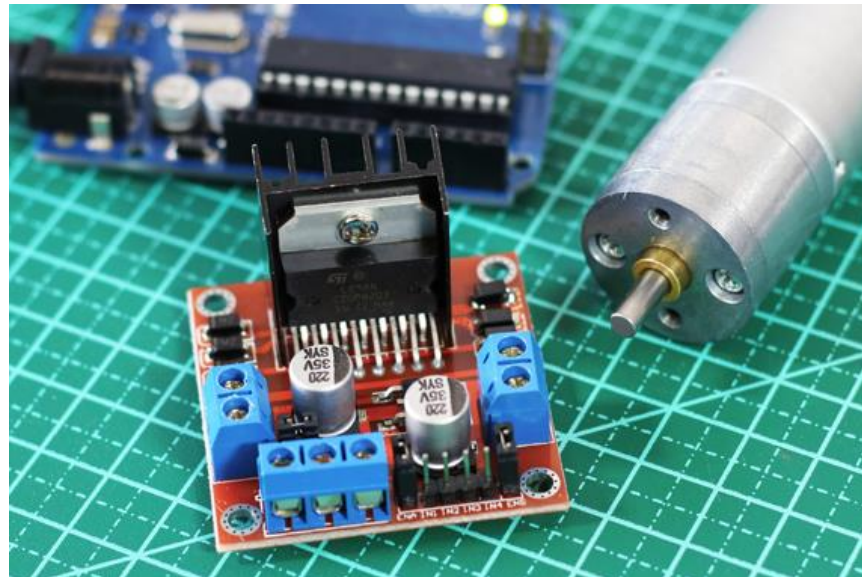


Drive de motor

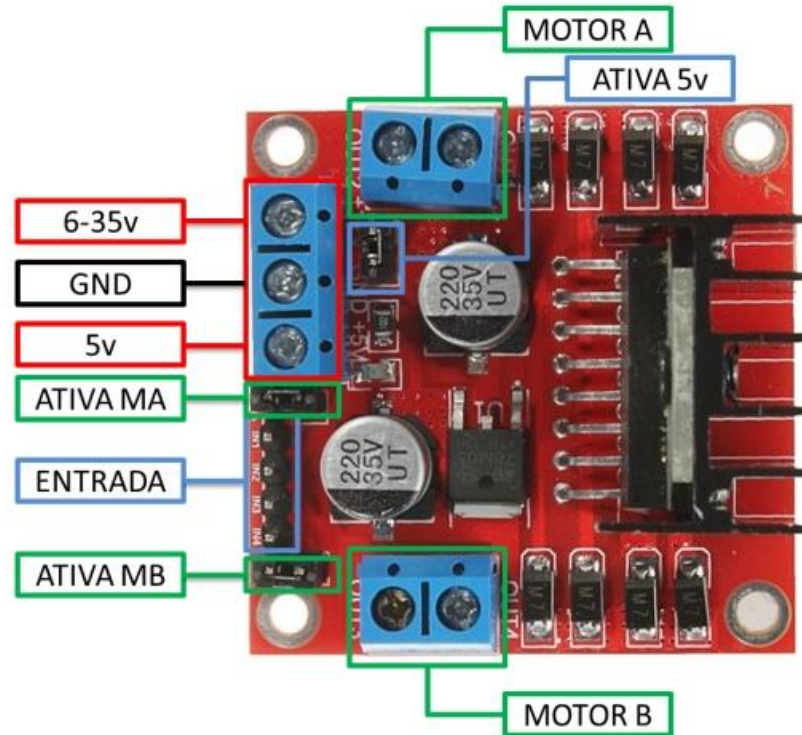
Representação com transistores



Drive de Motor L298D



Drive de Motor L298D



Drive de Motor L298D

Especificações

- Tensão de Operação: 4~35v
- CI: ST L298N ([Datasheet](#))
- Controle de 2 motores DC ou 1 motor de passo
- Corrente de Operação máxima: 2A por canal ou 4A max
- Tensão lógica: 5v
- Corrente lógica: 0~36mA
- Limites de Temperatura: -20 a +135°C
- Potência Máxima: 25W

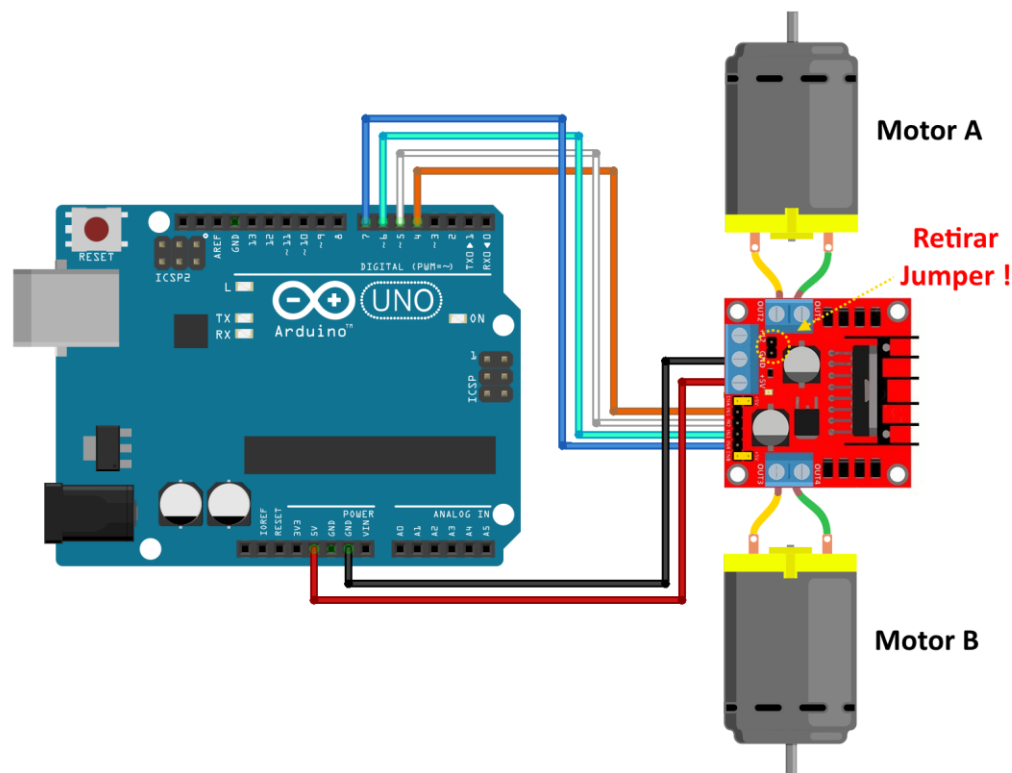
Drive de Motor L298D

A tabela abaixo mostra a ordem de ativação do **Motor A** através dos pinos **IN1** e **IN2**. O mesmo esquema pode ser aplicado aos pinos **IN3** e **IN4**, que controlam o **Motor B**.

MOTOR	IN1	IN2
HORÁRIO	5v	GND
ANTI-HORÁRIO	GND	5v
PONTO MORTO	GND	GND
FREIO	5v	5v

Drive de Motor L298D

Diagrama de Conexão



Exemplo de programação

```
//Definições pinos do Arduino ligados a entrada da Ponte H
int IN1 = 4;
int IN2 = 5;
int IN3 = 6;
int IN4 = 7;
void setup() {
    //Define os pinos como saída
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);
}
```

Continuação do programa...

```
void loop() {  
    //Gira o Motor A no sentido horário  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, LOW);  
    delay(3000);  
    //Para o motor A  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, HIGH);  
    delay(1000); //Gira o Motor A no sentido anti-horário  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, HIGH);  
    delay(3000);  
    //Para o motor A  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, HIGH);  
    delay(1000);  
}
```

Vídeo

Simulação de uma ponte H

FIM.

Obrigado!

Contatos

Gustavo Carlos

Email: gustavo_carlos@ymail.com

Github: <https://github.com/GustavoCarlos>

LinkedIn: <https://br.linkedin.com/in/gustavo-carlos-48a12624>

Thobias Santos

Email: Professor@tarzancomponentes.com.br