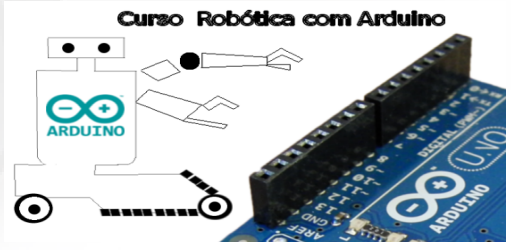
Robótica com Arduino

Tiago Ribeiro Santos

ARDUINO ES

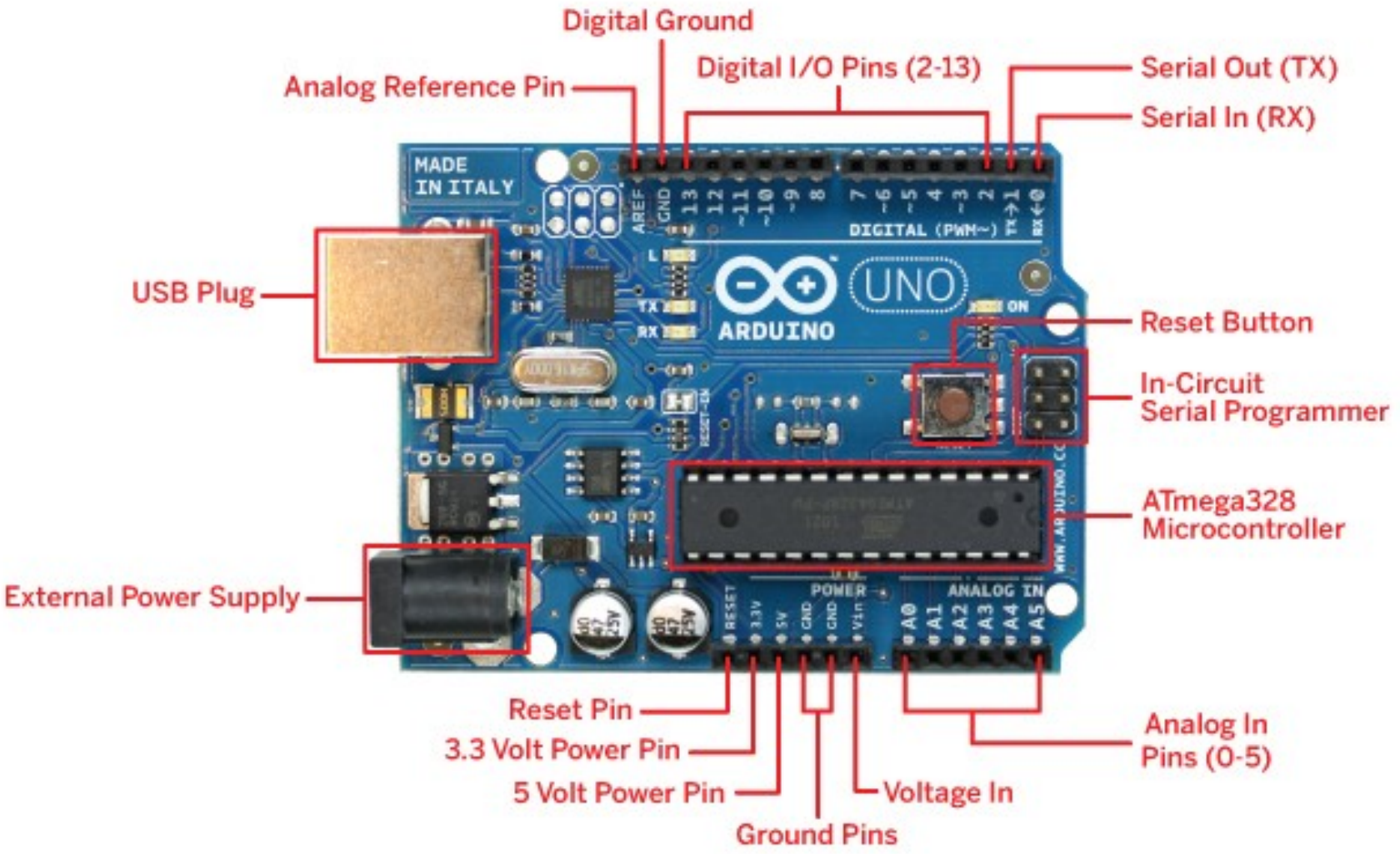
www.arduinoes.com.br



Who’s this guy? Tiago??

- **Programador** Python,JAVA,C,PHP, Arduino e Raspberry.Desenvolvo projetos com hardware livre para empresas e clientes físicos.
- **Fundador** do Grupo Arduino EspíritoSanto www.arduinoes.com.br
- **Pesquisador** em soluções com Hardware Livre/projetista(IFES/ D.I.Y)
- **Monitor de lógica/programação** no IFES – Instituto Federal do ES – Campus Santa Teresa (Acadêmico do Curso Análise de Sistemas)
- **Instrutor de robótica educacional** e atuo como desenvolvedor com interesses em visão computacional,microcontroladores e eletrônica.

Hardware Arduino();



Especificações Técnicas(UNO)

- 28 pinos ,podendo ser 23 pra I/O.
- 5 pinos analógicos (Analog I/O) podendo ser usado como saida e entrada digital.
- 32 KB de memória Flash (Sketch))
- CPU ATmega328
- Pinos especiais para PWM ,Interrupção e Comunicação Serial
- Fornece alimentação e saída (3.3V,GND,5V)

Entrada/Saída ARDUINO(INPUT)

- **Entrada/Saida analógica** → Informa qual nível de tensão aplicada em seu pino, conforme o tempo,temperaturae diversas outras grandezas, convertendo em algum valor lido.

  Fornece entrada/saída de valores entre 0 a 1023.

- **Entradas/Saída digital** → Assume apenas 2 valores binários (0 e 1) de “tensão aplicada” ao pino que está sendo lido. São dois estados lógicos:

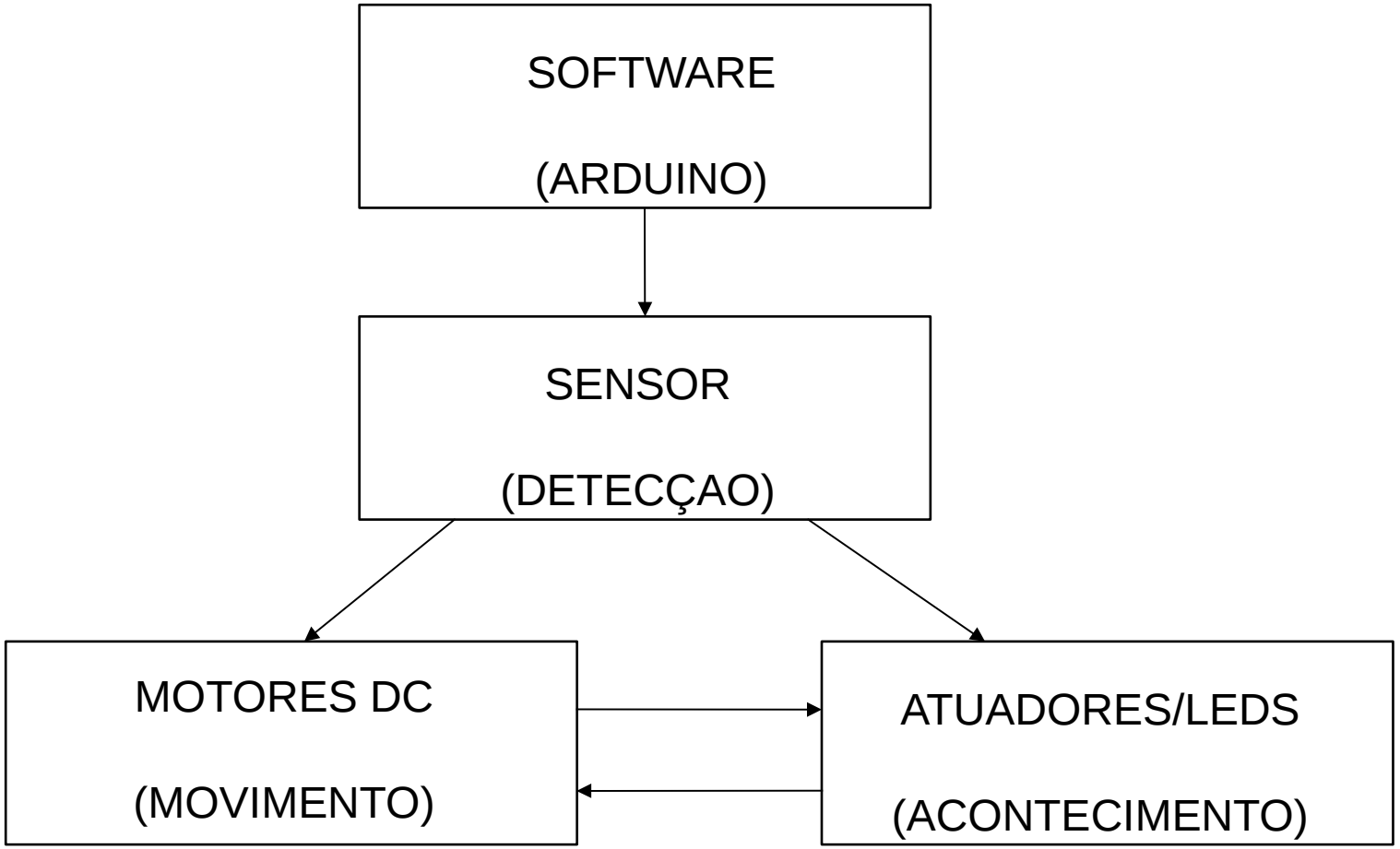
  HIGH – 5 Volts(Estado binário **→ 1)**
  LOW – 0 Volts (Estado binário **→ 0)**

Funções Internas (ARDUINO)

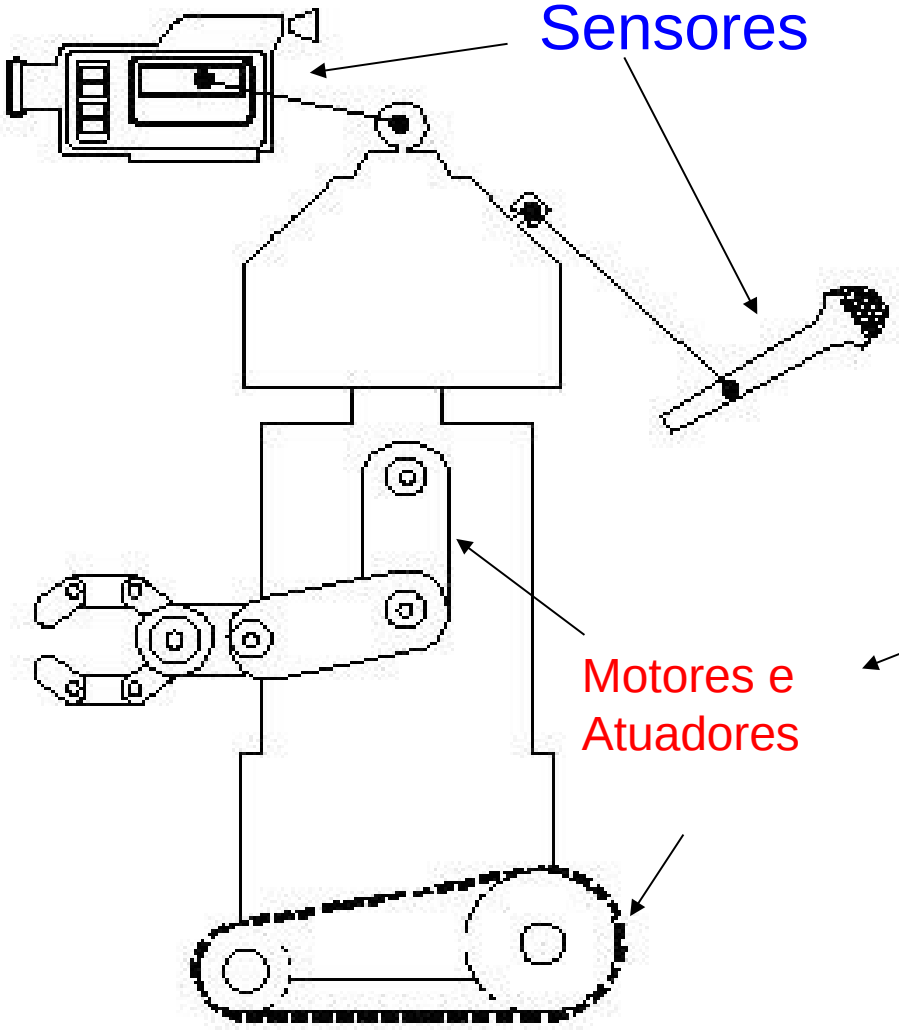
- **digitalWrite(num_pino,valor)** → Escreve na saída digital os valores : HIGH(5 V) e LOW (0 V) usados para ligar ou desligar um pino digital.
- **pinMode(num_pino,mode)** → Designa o modo de configuração do pino, se é saída (OUTPUT) ou entrada (INPUT).
- **delay(tempo_ms)** → Espera por um determinado tempo em milissegundos. 1 segundo = 1000ms.
- **analogWrite(pino,valor)** → Escreve valor de saída analógica entre 0 a 255.

Diagrama do Robô

- A representação de nosso esquema geral do robô pode ser mostrada logo abaixo:



Sensores e Motores(Projeção)



Captam o estado atual do robô em relação ao ambiente externo.

Modificam o estado do mundo (movem o robô)

Estrutura do Robô

- Para o corpo do robô , você pode utilizar de qualquer material reaproveitado com motores DC. Caso tenha,use um carrinho de brinquedo.

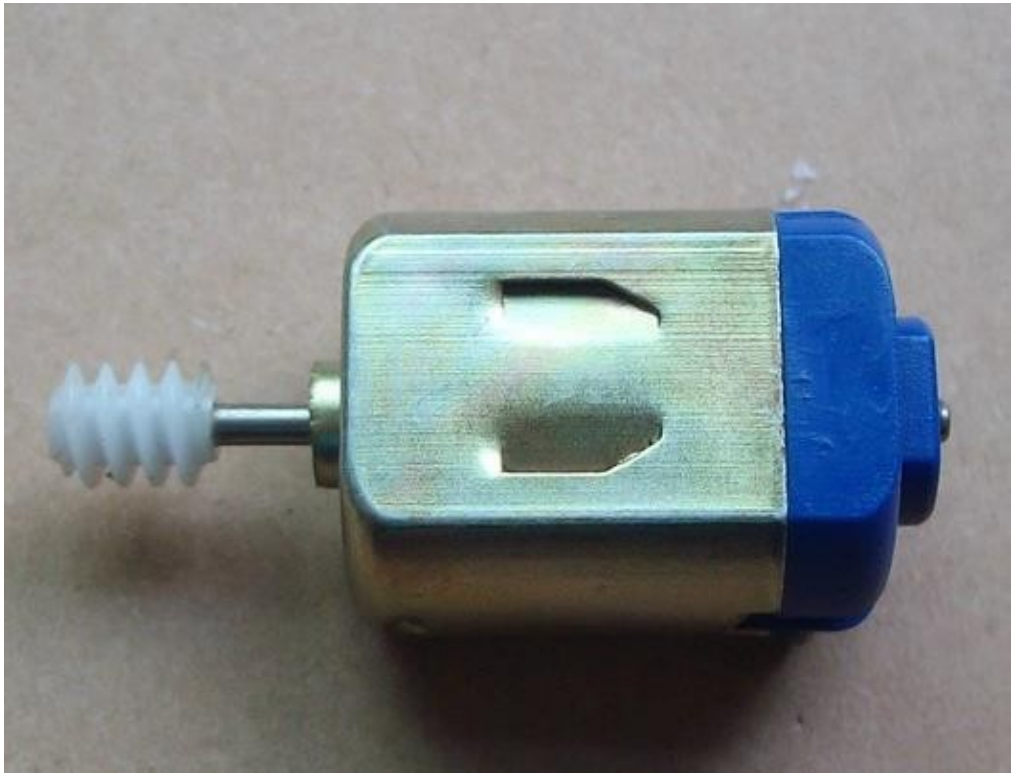
- Podemos também usar engrenagens e motores que compõem drivers de CD/DVD para movimentar rodas.Eixos,roldanas,rodas e chassi,tudo isso é reutilizável para montar nosso robô.

Motores DC

- São utilizados como atuadores para movimentação de robôs.Quando ligamos um **motor DC**(corrente contínua) com uma bateria,observamos que ele gira a uma velocidade constante e única direção.

- Para alterarmos o sentido de rotação de nosso motor,simplesmente invertemos os terminais.

- Mas, porque trocar os terminais do motor todas as vezes que precisamos inverter a direção?

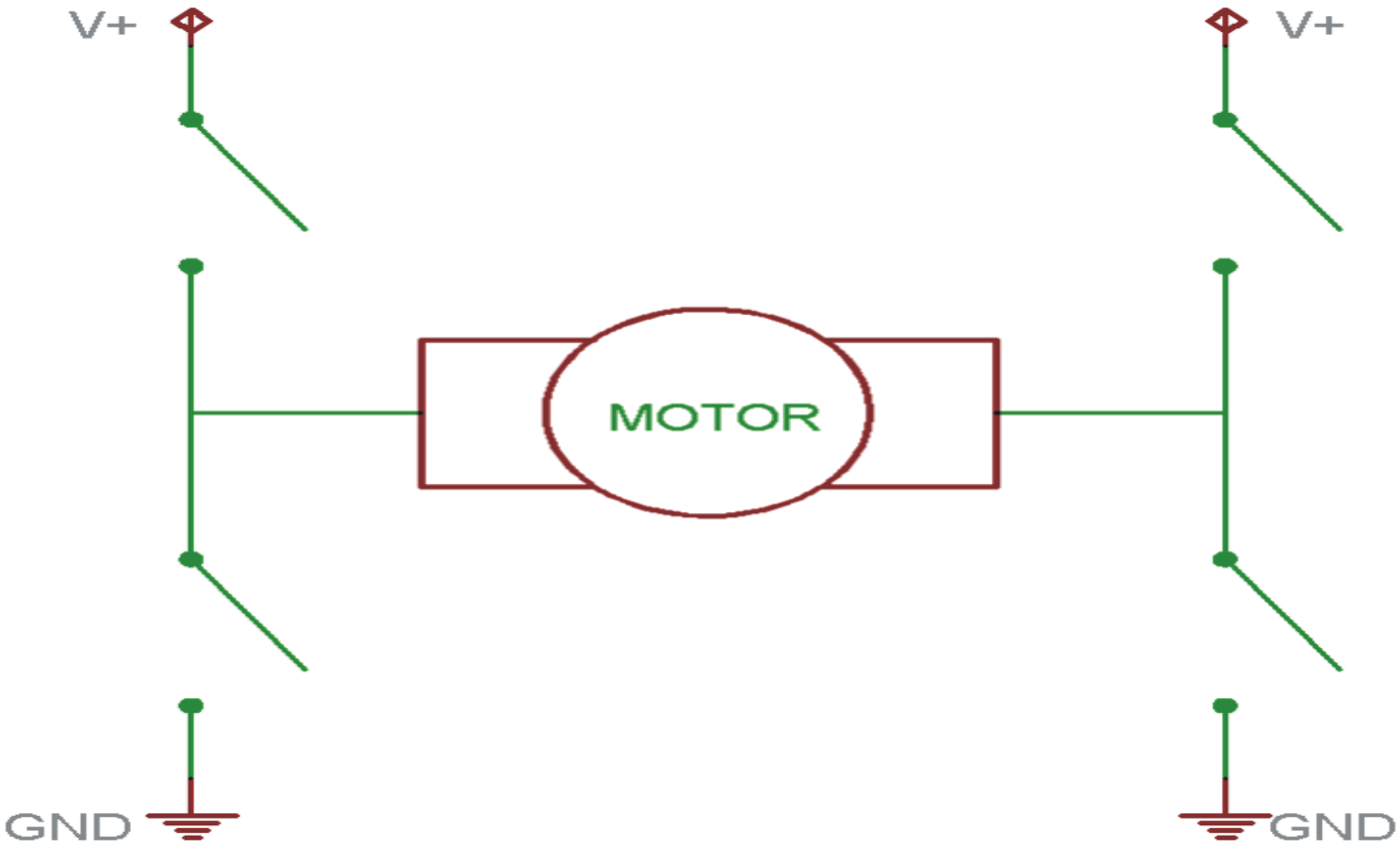


Observações Motores

- A corrente fornecida pelo o **Arduino de 40 mA(milamperes) é insuficiente** para acionar diretamente um motor elétrico,mas ainda sim, coseguimos ligar um motor em uma porta Arduino.
- Portanto,há um risco a correr. Motores DC podem danificar um pino do Arduino .Para isso , usamos um **transistor ou um CI conhecido como L293N** a qual explicaremos melhor a seguir.

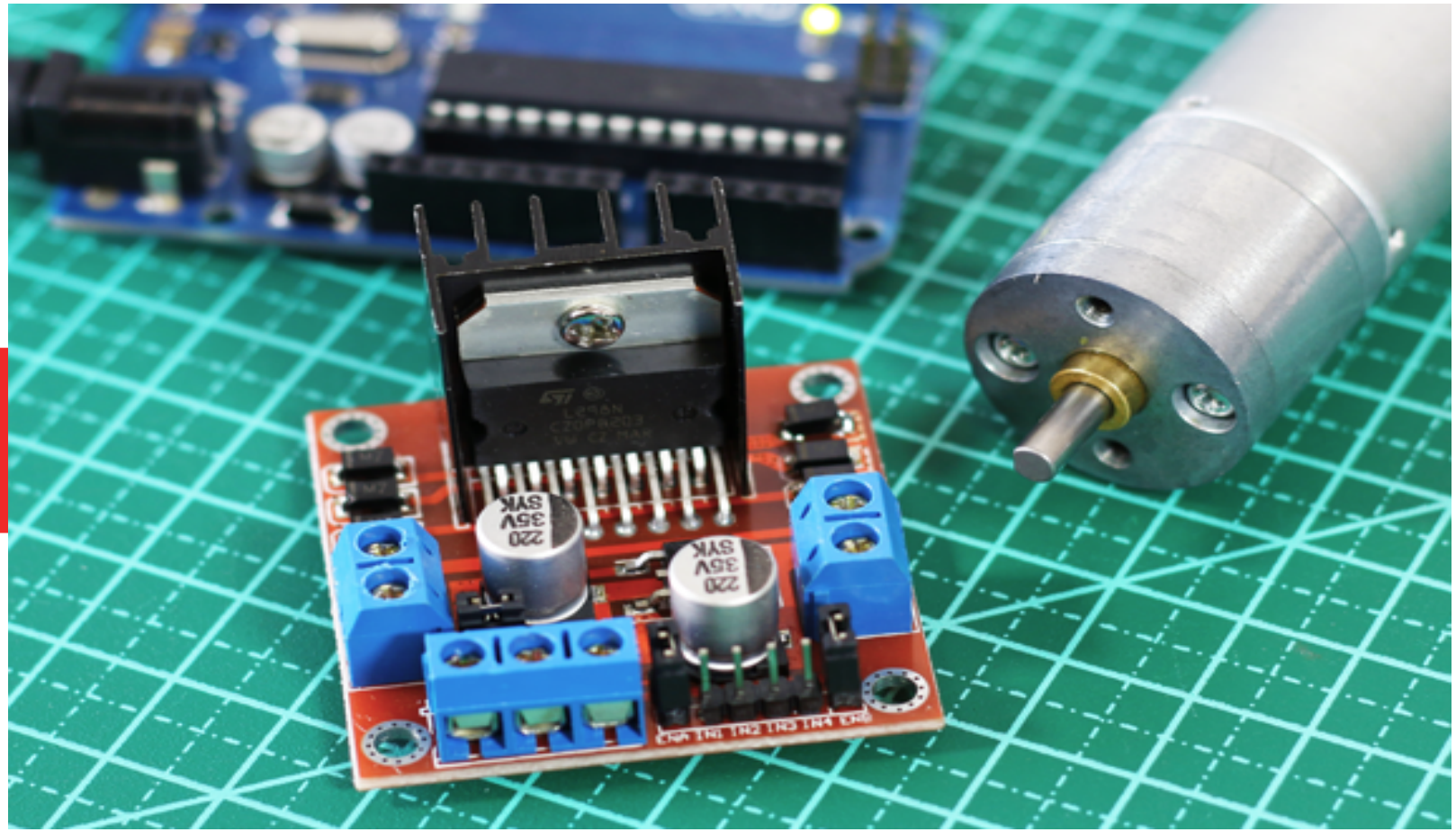
Ponte H

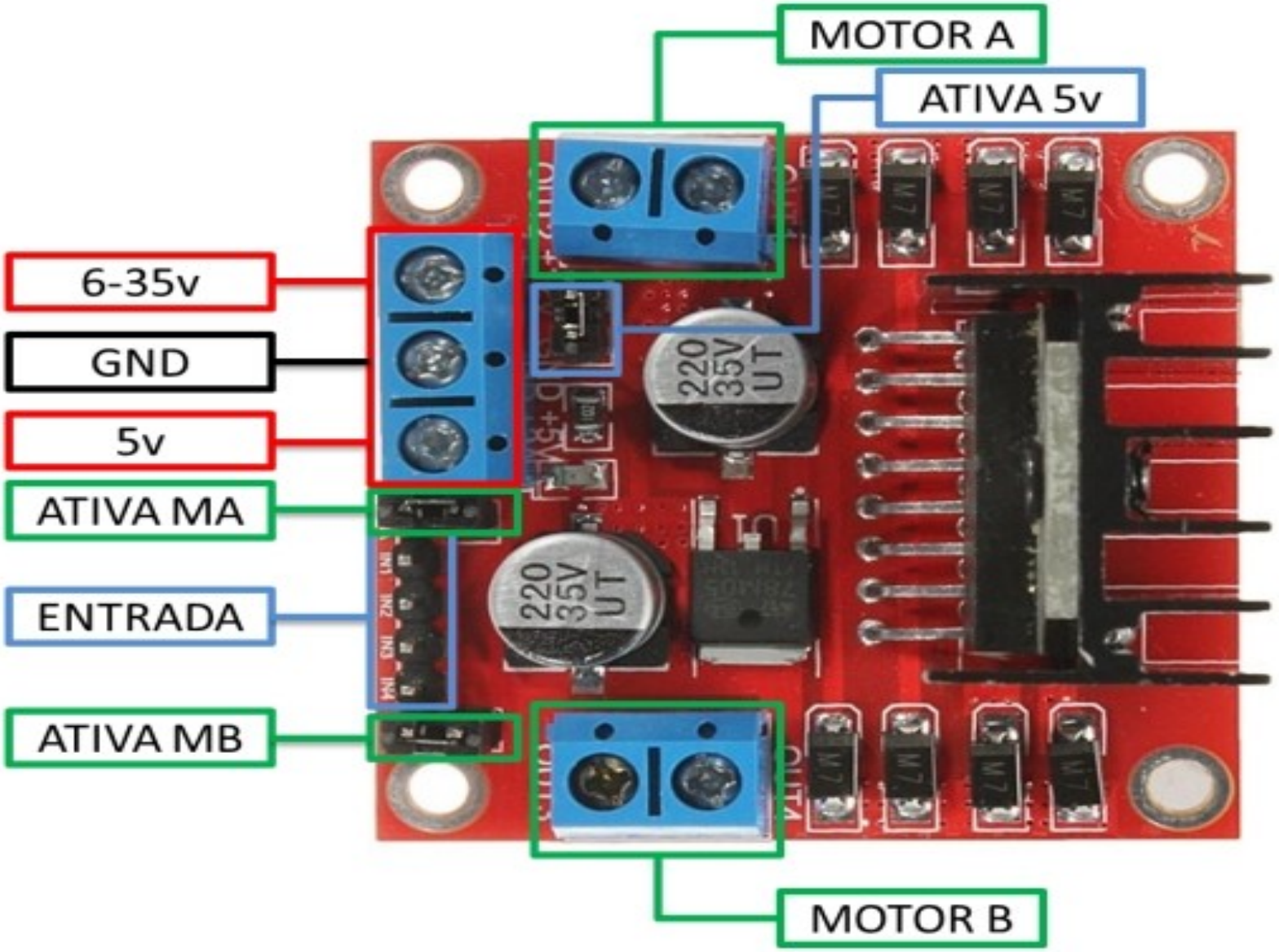
- Circuito para inverter o sentido de rotação do motor, utilizando-se de transistores e relés.



Bridge Driver H L293N

- Para este minicurso ,estaremos usando o módulo **Driver Ponte H L293N**.

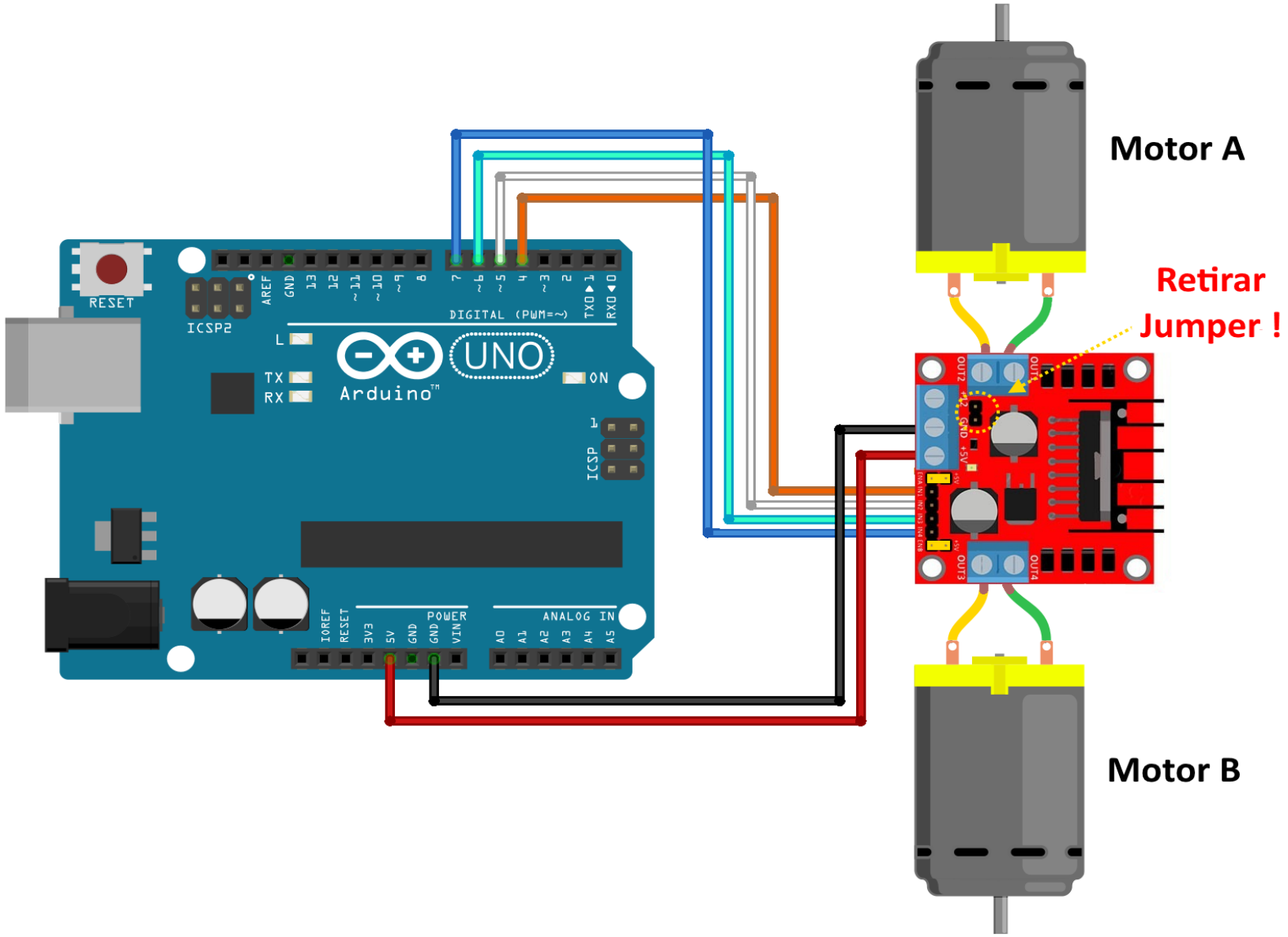



MOTOR A
ATIVA 5v
6-35v
GND
5v
ATIVA MA
ENTRADA
ATIVA MB
MOTOR B
220
35V
UT
IN1
IN2
IN3
IN4

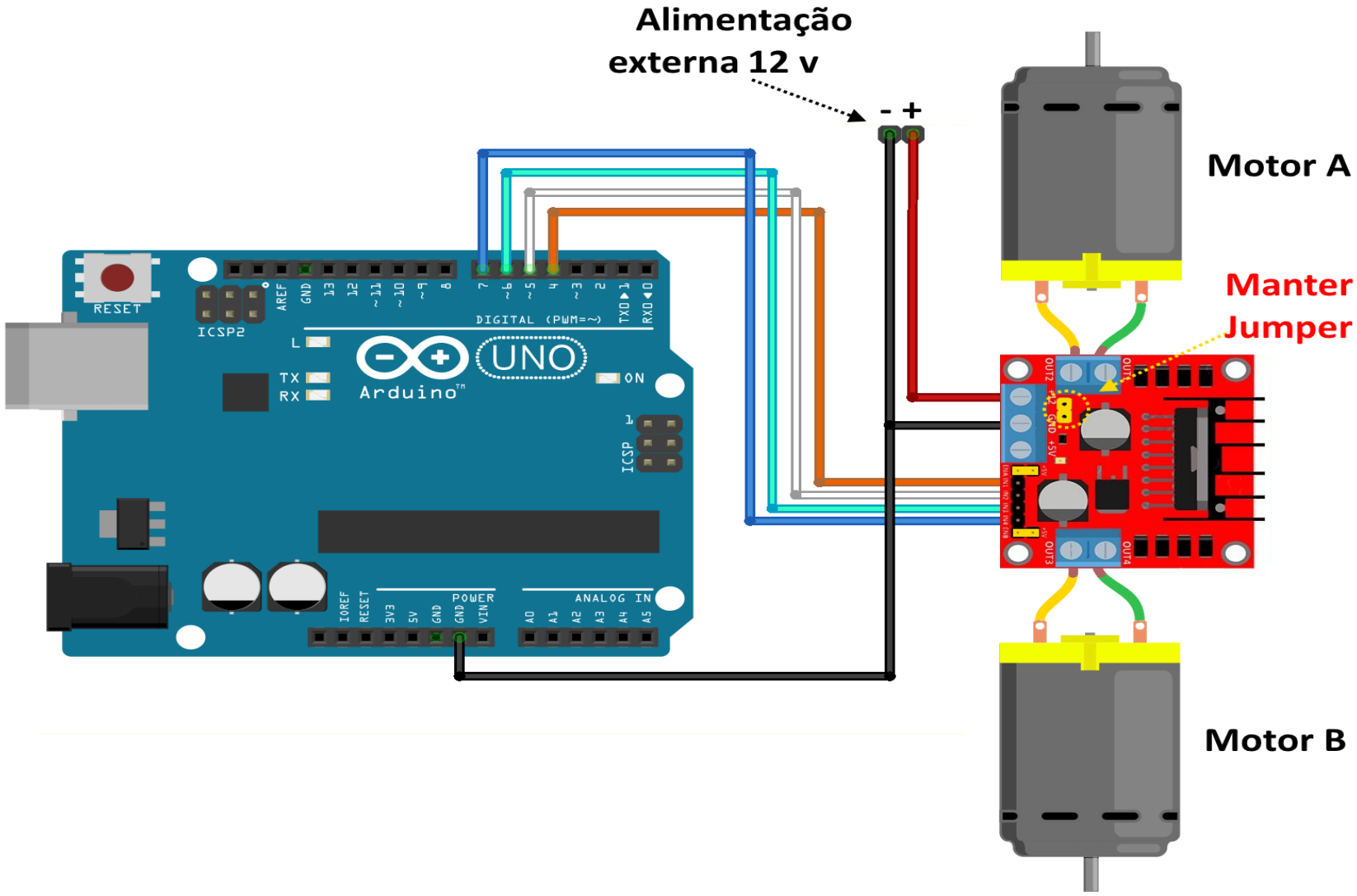
Ação dos Motores

S1	S2	S3	S4	Ação do Motor
Fechada	Aberta	Fechada	Aberta	Gira em sentido anti-horário
Aberta	Fechada	Aberta	Fechada	Gira em sentido horário
Aberta	Aberta	Aberta	Aberta	Roda Livremente
Fechada	Aberta	Aberta	Fechada	Freia ou esquerda
Aberta	Fechada	Fechada	Aberta	Freia ou direita

Alimentação Interna



Alimentação Externa



Funções procedurais → void();

- Vamos agora organizar nosso código criando **procedimentos** para cada tipo de movimentação do nosso robô(senso de direção).

- A palavra especial **void**,determina uma função que não retorna nenhum valor ou o que na programação chamamos de **“procedimento”.**

```
void nome_da_funcao(){
    //Instruções e comandos função
}
```

Void de direções();

- ```
  void frente(){
      digitalWrite(motor1A,LOW);
      digitalWrite(motor1B,HIGH);

  }
  ```

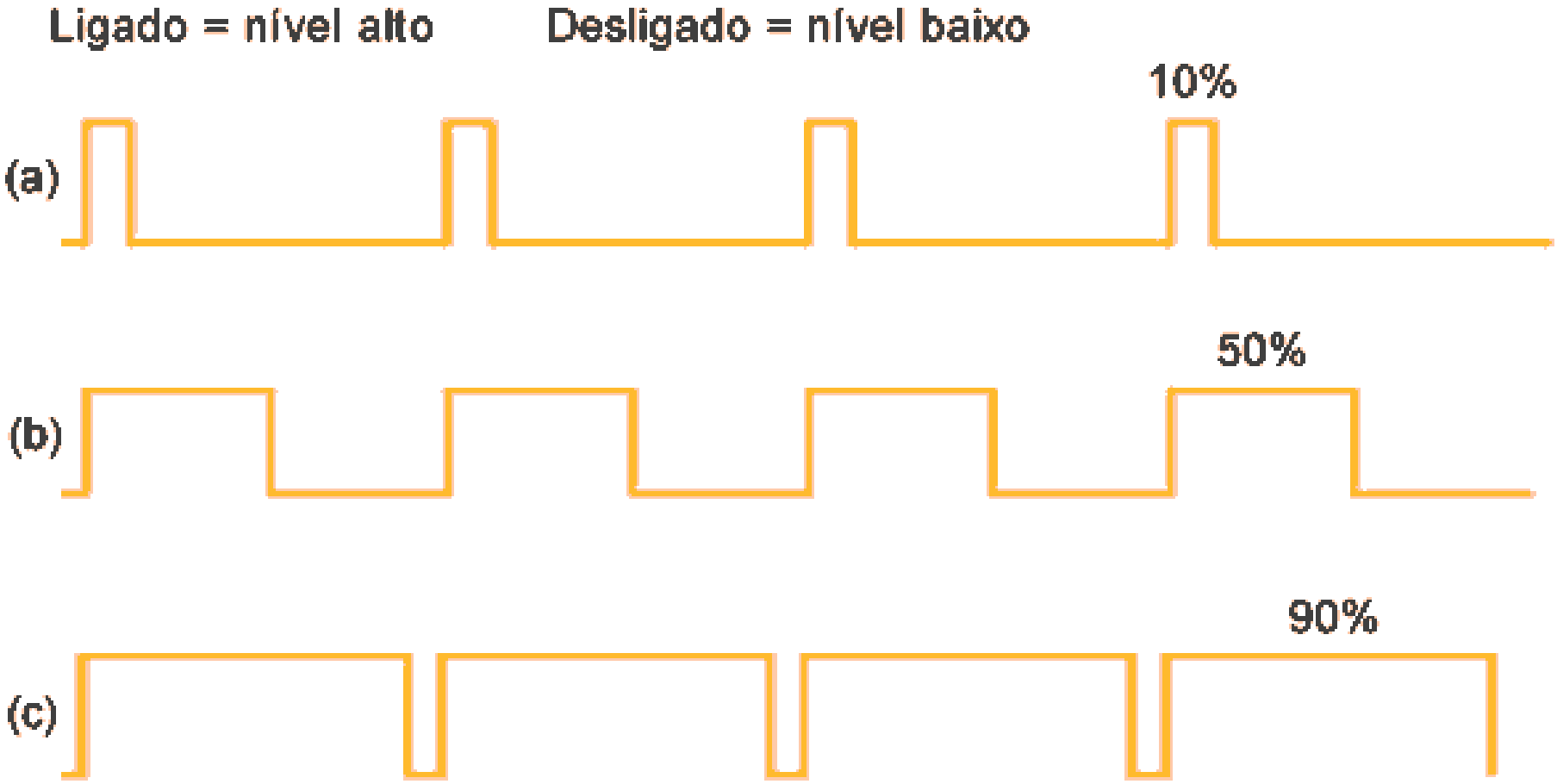
- ```
  void tras(){
      digitalWrite(motor1A,HIGH);
      digitalWrite(motor1B,LOW);
  }
  ```

- ```
  void parar(){
      digitalWrite(motor1A,LOW);
      digitalWrite(motor1B,LOW);
  }
  ```
  ```

PWM

- **Pulse width Modulation(Modulação de largura por pulso) funciona , com a aplicação da sequência de pulsos de tensão sendo aplicadas a um determinado componente.**

- **Uso a analogia do carrossel de um parque infantil:”Se você dá início a um carrossel ,ele continuará a girar sozinho até começar a diminuir a velocidade.Portanto,para que o mesmo possa continuar a ganhar velocidade ,é necessário que nós aplicamos uma força física.Imagine esta força, uma tensão aplicada.**


Ligado = nível alto
Desligado = nível baixo
10%
(a)
50%
(b)
90%
(c)

PWM

- No Arduino podemos usar o PWM de maneira bem fácil,através da instrução **: analogWrite()**

  analogWrite(pin,duty cycle);

- Na qual ***duty cyle*** é um valor entre **0 a 255**, e pin é uns dos pinos PWM do Arduino(**3,5,6,9,10 e 11)**.

- Exemplo (Aplicando PWM num LED):

  analogWrite(led,120); //Aprox 2.2 V de tensão

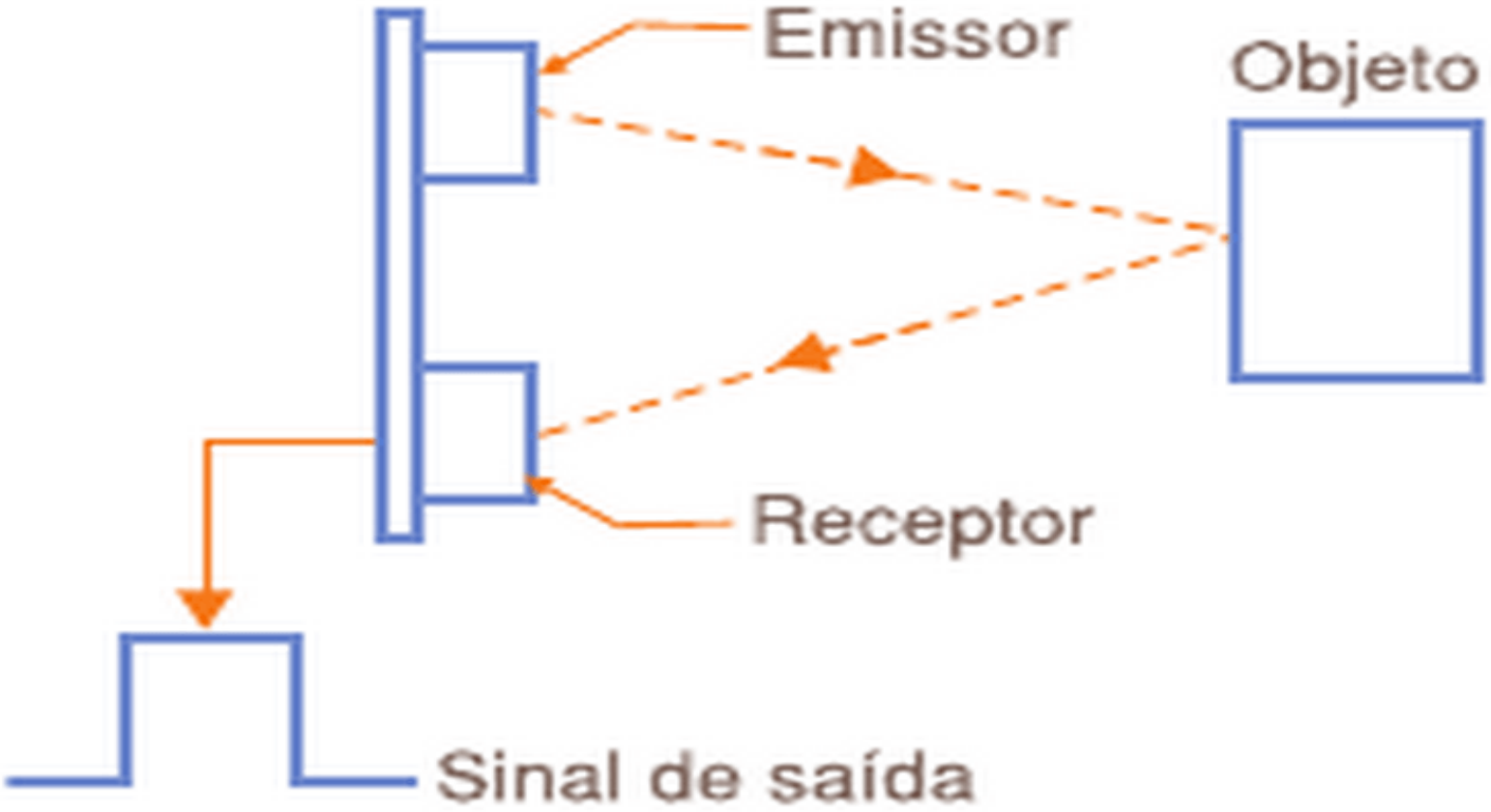
Ultrassonic Sensor(HC-SR04)

- O sensor de ultrassom é excelente para detecção de objetos nas imediações do nosso Arduino.

- O princípio de utilização funciona o seguinte: **“Você emite um som e espera até ouvir o eco, se você possuir o tempo certo,saberá se algo está lá for a e qual a sua distância”.**

- Esse é o mesmo conceito,empregado para ***ecolocalização*** e é como os morcegos encontram objetos no escuro!

Emissão e captação do ultra-som.



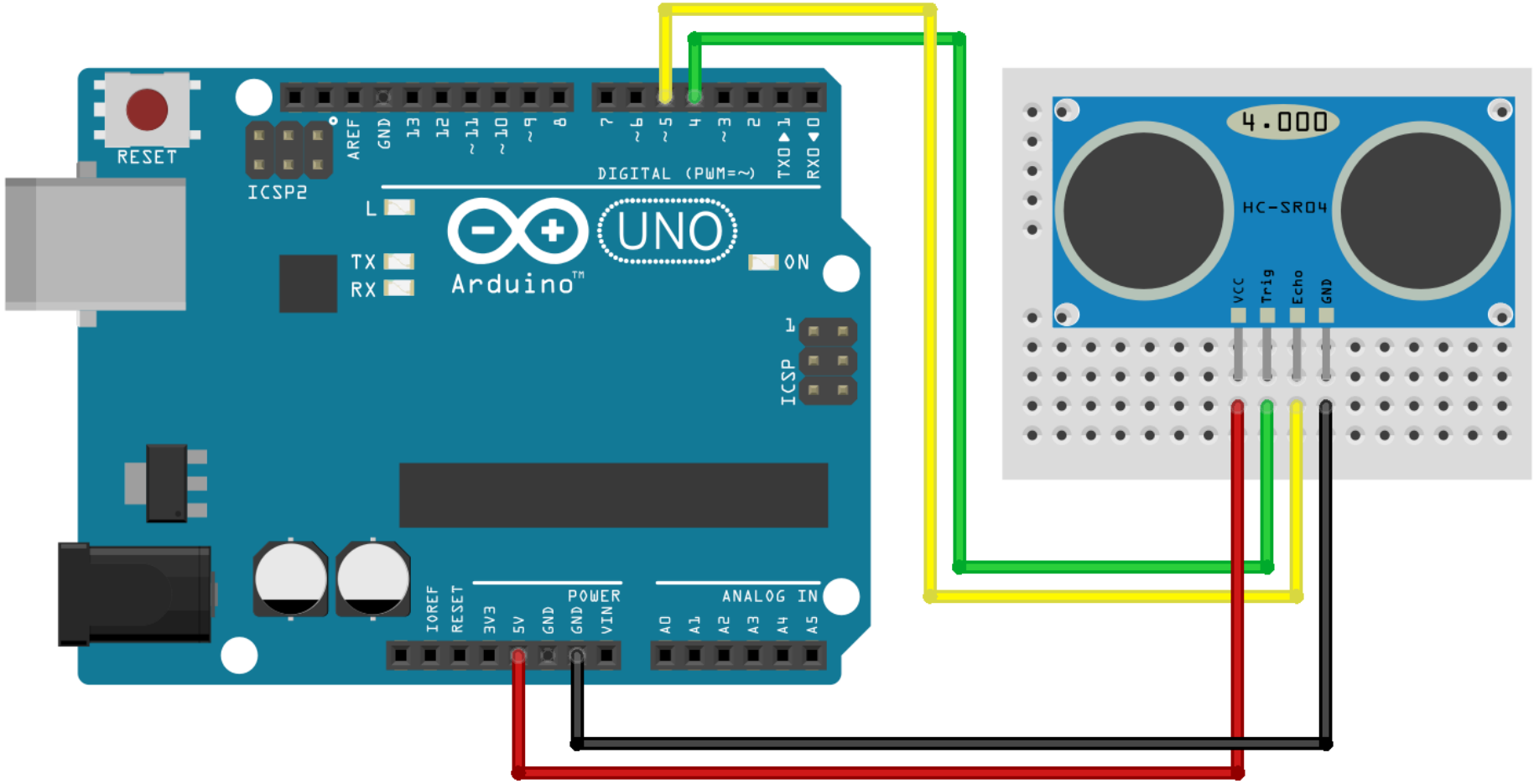
A Ultrassonic Library (“Ultrasonic.h”)

- Para utilização do sensor ultrassônico é necessário instalar a biblioteca **“Ultrasonic.h”** e incluir-la em nosso sketch.
- Link para download da biblioteca HC-SR04(UltraSensor):

  https://github.com/JRodrigoTech/Ultrasonic-HC-SR04/archive/master.zip
- Para instalar a biblioteca é só irmos em no ArduinoIDE:

  Sketch >> Library >> Add a zip File(library)

A Ultrassonic Sensor Build on Protoboard



The Ultrassonic Code (Part I)

```
#include <Ultrasonic.h>
int echo = 4;
int trigger = 5;

Ultrasonic sensor (echo,trigger);

void setup(){

    Serial.begin(9600);

}


void loop(){
    float leitura_distancia;
    long microsec = sensor.timing();
}
```

The Ultrassonic Code (Part II)

```
leitura_distancia = sensor.convert(microsec,
Ultrasonic::CM);

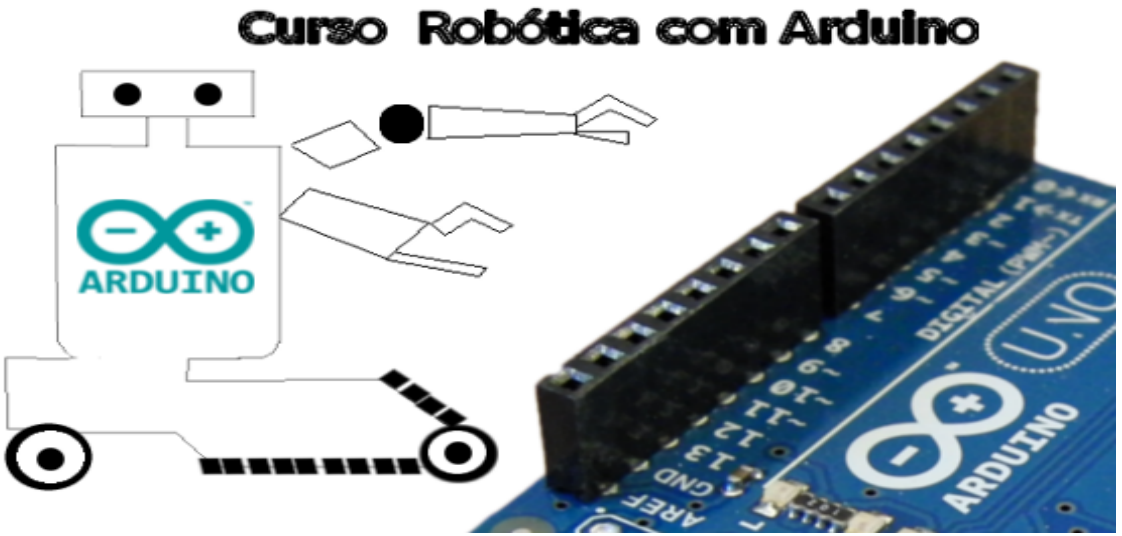

 Serial.print(leitura_distancia); // CM or INC

 Serial.println(" cm" );

 delay(1000);

}
```

Robotic Course(El curso del Robotica)



- Facebook:

https://www.facebook.com/CursodeRoboticaES/

Muchas Gracias! Thanks! :)

- My Facebook :
  https://www.facebook.com/tiago.arduinoRobotica

- My GitHub:
  https://github.com/tiglinux

- My Website:

  https://tiagoprogramador.carbonmade.com/

  Email :

  tiago.programador@hotmail.com
  gnu_tiagolinux@openmailbox.org