**1. INTRODUÇÃO**

A robótica móvel é uma área que desperta a curiosidade de muitos e que tem evoluído bastante nos últimos anos para o desenvolvimento de sistemas de controle de robôs.

O projeto propõe um controle de trajetória de um carro robô usando PID, onde o robô irá sempre seguir reto na direção inicial.

As etapas da construção do veículo estão esquematizadas em três sistemas e consequentemente seus subsistemas, tais como sistema mecânico (estrutura), sistema elétrico e o sistema eletrônico de controle (controle de velocidade, aceleração e posição, motores e bateria).

No sistema mecânico, a estrutura pode ser comprada onde é composta de chassi e pneus.

A etapa que faz referência a parte elétrica e eletrônica do sistema, serão abordados os motores DC, o sensor de velocidade/aceleração (MPU5060), a ponte H, bateria e o controlador Arduino UNO.

**2. EMBASAMENTO TEÓRICO/PRÁTICO**

**2.1 MODELAGEM DO SISTEMA**

O motor de corrente contínua é representado por 3 equações:

* Equação da armadura

https://lh6.googleusercontent.com/6P53jjYM45mVbxK0P-7tgUdZEFn2zpO_SHTtQ2aDt-IaygCfD1A-eiJzoouQ94JW4l2yCdJ3avEQGBdR7pydVJgXvt-VUG3xWO6mQiOd3pdAq8eH9R_VTTzgJ4_JE9qr96DiHTo

onde https://lh4.googleusercontent.com/U62UbIa7rFXEGh9MImkingBoCX6Z5L39hsqe-jKYsDRj9sNYrbZfZBcjK7ameDApiiHZ5IvEFT7gk4qBo7Iiz0FqwnWMKi5C9rBe7WqpgHVecrvdvl-XLzXcciQlhpDYWOTNebgé a força eletromotriz, https://lh6.googleusercontent.com/K6Znew0xpwkEKwC0bF6X_gMFQbgpQpj6Y_Ybdn5umF3vDIhVjGfZX56XuU3ZRbNAeLIsBXlg-GwefGYVHITJUIzR1dIB0F4BFe2yPsBY3KOv48UxV7QavatkwGHQs3GABeIKPecé a tensão de armadura e https://lh5.googleusercontent.com/sHZEbBNLo8kW4edpwvXuwr9WXiAmgImFrjcHFEbAEUp2vPJlVN-LD878PV7z9DKoE3Wf0gpAuzWP89kXYRnAt3273TWjHS_6Dy_F0Er5pETZ--Z4wprr6d6XB78AkQNAQySrqLQe https://lh5.googleusercontent.com/nAJ3-hjn8cQm-UKn80h1iH2teybanu2i2RUcOfBNmDtJ2BcYc9DBIMDtX3DQ4n_CFWqixkawHlmYJ4oZklvPGvhCCFaSUcp73M3Fkw5JYNL9SSkZG2vPDB5eNEp0tGKlrNGGTT8são a resistência e a indutância da armadura, respectivamente.

* Equação da força eletromotriz

https://lh4.googleusercontent.com/5Gj5FCvVKOLNKPG4a0OaAYO__cNxDNRTnQ9wqjnDHgpFfXil8g86De5v_1vAfQ40Ai37wwhhnozeYt4Zo9ygjn9SY-2BnstmD_hCjIwIbXedg2HYvJ19m0a-4iSZqdC06xVZuJU

onde https://lh6.googleusercontent.com/yO6sGFdIP2je2RNdEt6eEwnq-jda1BdmEfgOZhmNwyElvVFCmdmhhWj_PgXAmd-ikZrN5L6LbGXMZ6tecLPGS4o0FoJbYUeah5MJUlF2NW3riS_vKBYLeiZB2ketQAF0rFg_jlIé o ângulo mecânico e https://lh5.googleusercontent.com/saIge8HL2OfmzyM-DwQ5VlwhaD7R5NW7KwQEpWm37Mw3ktx3qNdi56TemtCcP4P08IeZfTpRGIVBG6lGq-XQOZgpaSiFAKrHZO3AiGPH4_2vaciUVvQ7gf3qrB7BfZlOk7oFV7sé uma constante.

* Equação do torque gerado

https://lh4.googleusercontent.com/KHNHwY8IAWc_KdCHf5L1il-gvfMolnVq-dqMcwypDwq2f36J5qnRHjBojuZa00Lq-Bmxf-1VXuPndJLs1OgV0-kO3bf-yqRVrRHjBIzzrkoDV0X7K3AE7IwI6-ZtBTe1oCU_aDo

onde https://lh5.googleusercontent.com/2H_dM784K_o0l6CSof_0pjPd8UFLaJYKoFasDAUNQMkOLzorgcSOLCmmazasBy94jgT1BPzv2r3Kg09Dz1FHLrzwlvttA7sAyrWvnWk2AmGSIjgqCrr8bOMupNDs9TxNvO0HK0sé o torque mecânico gerado pelo motor e https://lh5.googleusercontent.com/mDKwvpB88JyWoNVZzZD10LuyS2OO4NfRBF_X5mgxNxeTgKI-wxHm-bUYr1yc3HKV9fE0IueLC4wei1WrbqX68L9LPe8CneBMXe-A1Zq8K3QZv5nBKjeHC778jCTONtfLybQO8icuma constante.

Segunda lei de Newton para o movimento de rotação

https://lh6.googleusercontent.com/6mpuKLAP9o_PJXsiaa3dw4LwA9zgzzwOKrhmUK7zvdzBlkonMX-sYcHc_7Gv1NmBLGpRMNiRVXfUCwhaSaEe7DZBZgQG749-nZu3l8IBDskt_SSOGjh9ouP2UlNm9crh0nrGxb0

onde https://lh3.googleusercontent.com/5k54pFCij4yBj8LnIl3oaZAzUjuxXShj0Pff5a_jU3EnVVaLNoYma8_MmWJ8FYxX5WCh1m39hIh7IfsHR6ej3L-uMW3_slHmsIhvMfhBGMjktSucgL_rZCUQ6SdIQRUbGKVgQ8Eé a inércia do conjunto motor-carga, *b* representa um coeficiente de atrito e *d* representa a variação de direção que constitui a perturbação.

O ângulo mecânico é escolhido como variável de saída, ou seja, https://lh5.googleusercontent.com/4rTvNb4fIWdhLjnjP0Z2L7r5S3PYSnFrSGngGz-qS6rNVuLAxcAODljzrMINOnu-Nx9DdCcs_HZoSgaFWCoMNzSXNw006WrFxVDHfypVqMMWkmEuhz13OE2oIr3HCDggRg3jl2Q. Então as equações da força eletromotriz e do torque gerado são reescritas como:

https://lh3.googleusercontent.com/WAW4TmSc3OcXGOmZnR4wI9UdEWEtWR7daJqBBaVXMzoPH1zJilx-mYg_9f5G_xxAD9zaBUObri2XTbk40zx4uou2LMyRTHoHtvrqRBaIgkRazYG4GwOVGI0yPTdT4lKlktJcfjo

https://lh4.googleusercontent.com/sjF0p5hr-JXrz5M3GiB3hgA-99zA577wqrkH5GLjZkCvkYt1gjv2-kfRmEFcgpJRACgDjw7-cGfeQYSJa4kCBKIInanbfgWHBV7ToIlnlZDTdwXlB4glWqgzF9fbrO2lLMHzwlM

Fazendo a Transformada de Laplace das quatro equações, tem-se:

https://lh6.googleusercontent.com/YdD2Z8j0fCqbkUNAiSM4gr-1iyoZ3Hu2SByI0lkIvTorlNSeEJhliqvnPego6HF3JlTqZoa2KxQOXF20VffEObMFk4Kw4lpJOGHIfZ3X230eexujN3t6KTlfSEv3l_8shQDyjMY (1)

https://lh4.googleusercontent.com/uXnXTBpplivKabIrAB_-_bsYmk7hV3IFjQFqEz3_tBYZHxS6lEr1XjD54b65Ewo6SnX-aptr-9Gn3iKfiDKpKana66ORD2hFz9r6rRxKPLwVDppQfT5ClLzQi9hwIcHtBC09WEg

https://lh6.googleusercontent.com/Fo_6Z1ktfvwCwvprficO5dd0URwyH0nbKm7A4RK6nJC9xC2T1MIpggCGmrPy63EKdZIINugNgrc2NL36kAsalXV7xy2kMmEdV-iWTjE0jmMKI9I0cYEZ1taSagqK02d3-9J57hM (2)

https://lh6.googleusercontent.com/n4JucfJngIaRnQKQkW0ajE-w5SDX0vgx_6nuLJZ2_9mKBJZyoG8oYm34ZPbzHonQ8eb5pt9f9BqJZBTeZKlosK5ZMO8w3ojK8JuROqq7cZ7ooh-DJTyS42dNnXn5BPgg8J6XudA (3)

Isolando o valor de https://lh6.googleusercontent.com/ruAIV7ZMYQjCaO-4lTA-I99UV64EhRNdxbpl7RUYcQjoBI-PhQ_vq2wC1oPlWODBn-lJ-SpXX_HDbB0wDjde2I1zos-iCyMjAZDdN-2C-X0KPlXDYJFKsRLKNQUz8F9PJvbc_cona Equação 1, usando a Equação 2 e substituindo na Equação 3, obtém-se:

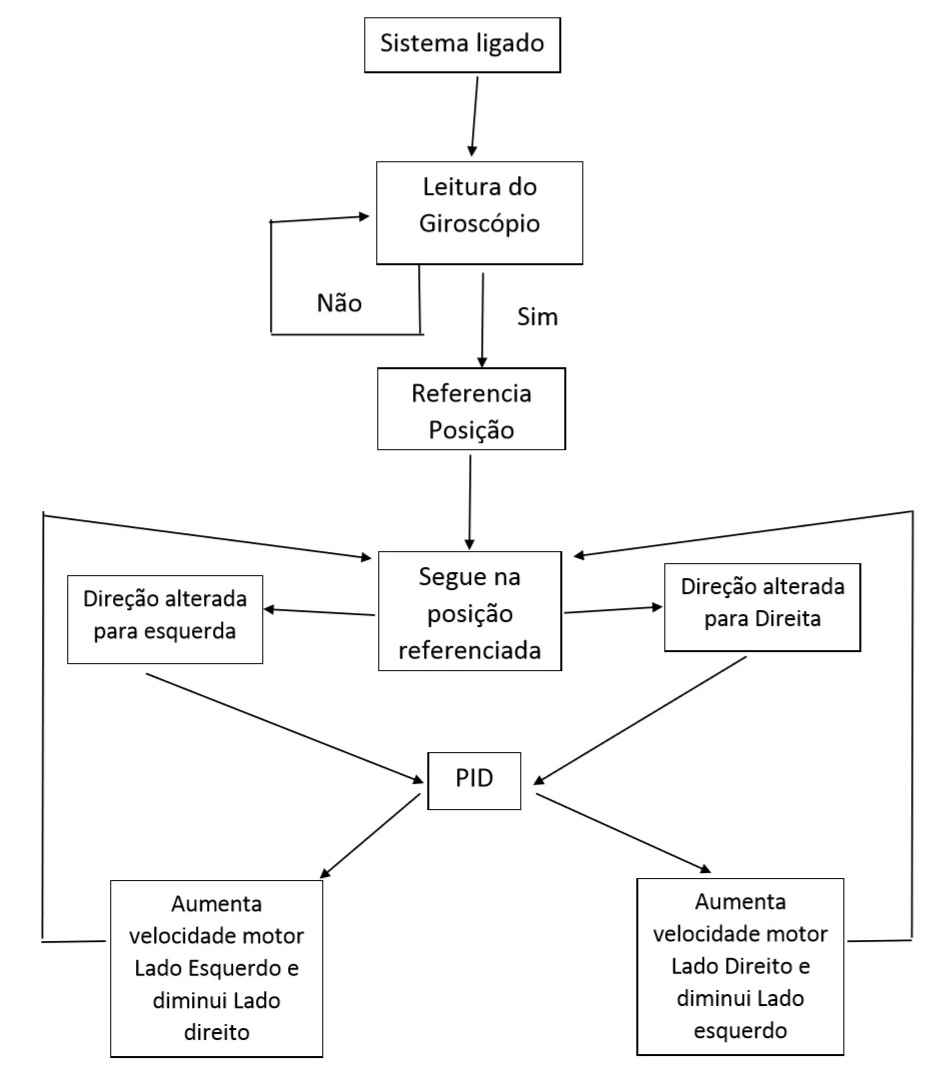
https://lh3.googleusercontent.com/1as7K7ydIDYsARzPeSSp5fm6IGG5UUnUdneaOP66oXF46_9KrcFxPMzuIe7DLzO5WVagT0AxurwjsD5IL7YC9n6KVnbZu0o8PXhyoUC3PVpvvbVXXqg-kOg8V0dCBH-7fYMn45Q

**2.2 DIAGRAMA DE BLOCOS**

O projeto do sistema de controle é fundamental para minimizar os efeitos dos distúrbios sobre a variável de processo que, no caso, é o ângulo.



**2.3 FLUXOGRAMA**

****

**3. IMPLEMENTAÇÃO**

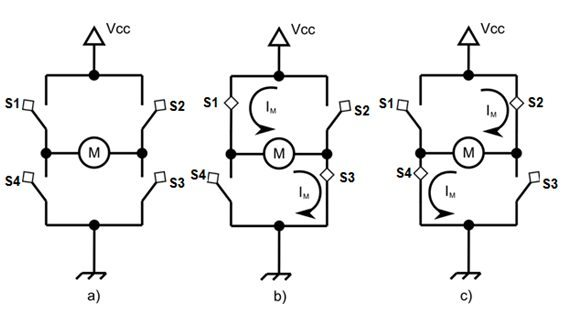
**3.1 PONTE H**

**3.1.1 O que é ponte H?**

É um circuito composto por um arranjo de 4 transistores, é capaz de acionar simultaneamente dois motores DC controlando não apenas seus sentidos, como também suas velocidades.

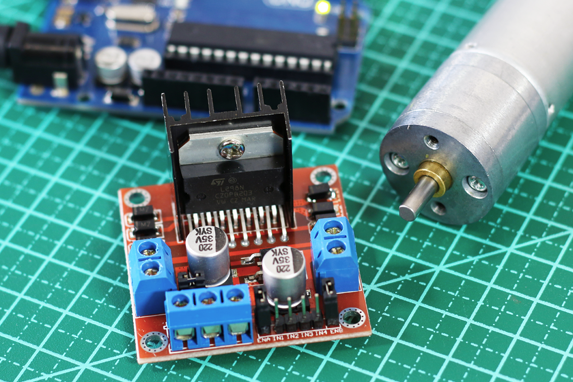
Seu funcionamento dá-se pelo chaveamento de componentes eletrônicos usualmente utilizando do método de PWM para determinar além da polaridade, o módulo da tensão em um dado ponto de um circuito.

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



Quando nenhum par de chaves está acionado, o motor está desligado (a). Quando o par S1-S3 é acionado a corrente percorre S1-S3 fazendo com que o motor gire em um sentido (b). Já quando o par S2-S4 é acionado a corrente percorre por outro caminho fazendo com que o motor gire no sentido contrário (c).

**3.1.2 Drive Motor Ponte H - L298N**

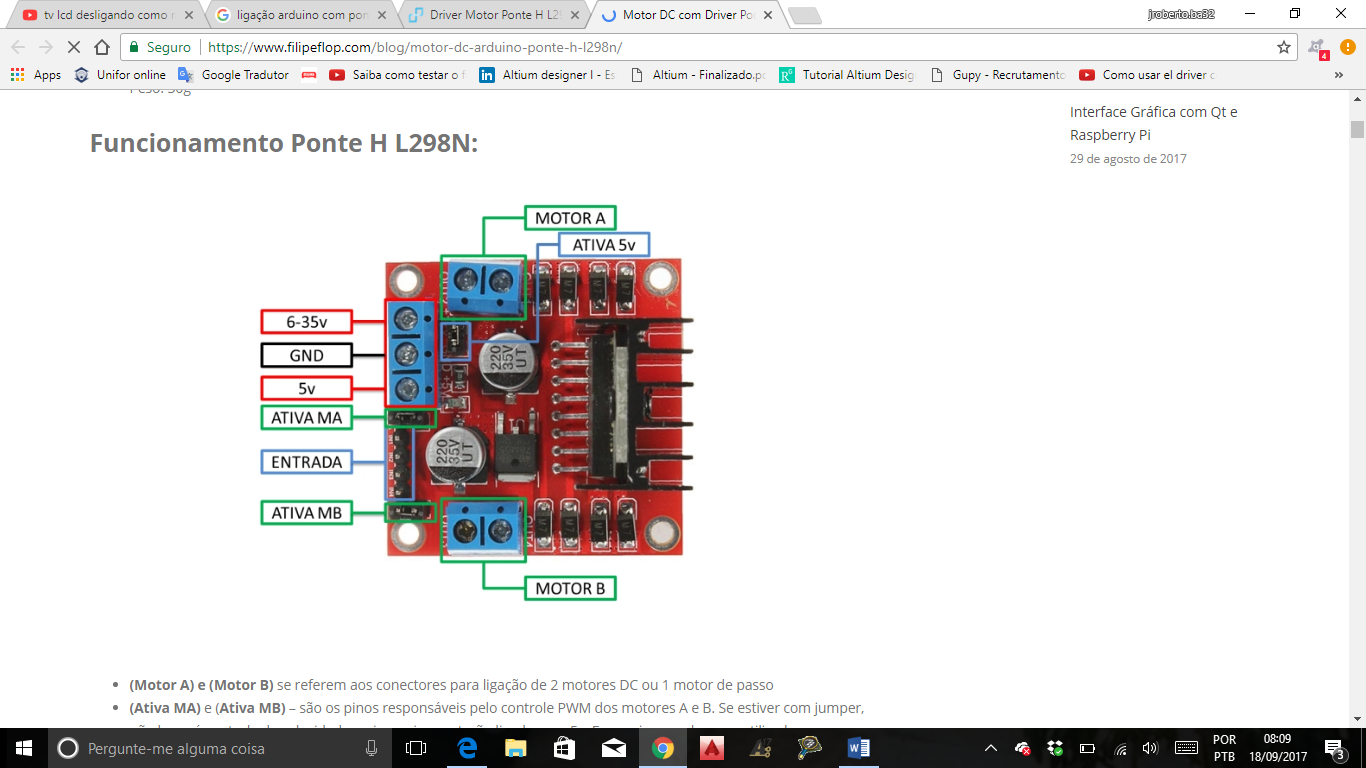


Este Driver Ponte H é baseado no chip L298N, construído para controlar cargas indutivas como relés, solenoides, motores DC e motores de passo, permitindo o controle não só do sentido de rotação do motor, como também da sua velocidade, utilizando os pinos PWM do Arduino.

**3.1.2.1 Especificações:**

– Tensão de Operação: 4~35v  
– Chip: ST L298N  
– 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  
– Dimensões: 43 x 43 x 27mm  
– Peso: 30g

**3.1.2.2 Funcionamento:**



**(Motor A) e (Motor B)** se referem aos conectores para ligação de 2 motores DC ou 1 motor de passo.

* **OUT1** - Terminal positivo do Motor 1 DC ou A+ do motor de passo
* **OUT2** - Terminal negativo do Motor 1 DC ou A- do motor de passo
* **OUT3** - Terminal positivo do Motor 2 DC ou B+ do motor de passo
* **OUT4** - Terminal negativo do Motor 2 DC ou B- do motor de passo

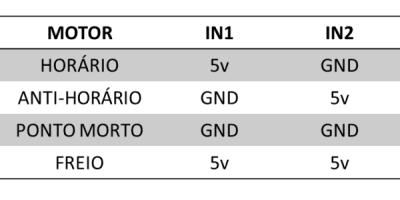
**(Ativa MA)** e (**Ativa MB)** – são os pinos responsáveis pelo controle PWM dos motores A e B. Se estiver com jumper, não haverá controle de velocidade, pois os pinos estarão ligados aos 5v. Se estiver sem o jumper, esses pinos podem ser utilizados em conjunto com os pinos PWM do Arduino

**(Ativa 5v) e (5v)** – Este Driver Ponte H L298N possui um regulador de tensão integrado. Quando o driver está operando entre 6-35V, este regulador disponibiliza uma saída regulada de +5v no pino (5v) para um uso externo (com jumper), podendo alimentar por exemplo outro componente eletrônico. Portanto não alimente este pino (5v) com +5v do Arduino se estiver controlando um motor de 6-35v e jumper conectado, isto danificará a placa. O pino (5v) somente se tornará uma entrada caso esteja controlando um motor de 4-5v (sem jumper), assim poderá usar a saída +5v do Arduino.

**(6-35v) e (GND) –** Aqui será conectado a fonte de alimentação externa quando o driver estiver controlando um motor que opere entre 6-35v. Por exemplo se estiver usando um motor DC 12v, basta conectar a fonte externa de 12v neste pino e (GND).

**(Entrada) –** Este barramento é composto por IN1, IN2, IN3 e IN4. Sendo estes pinos responsáveis pela rotação do Motor A (IN1 e IN2) e Motor B (IN3 e IN4).

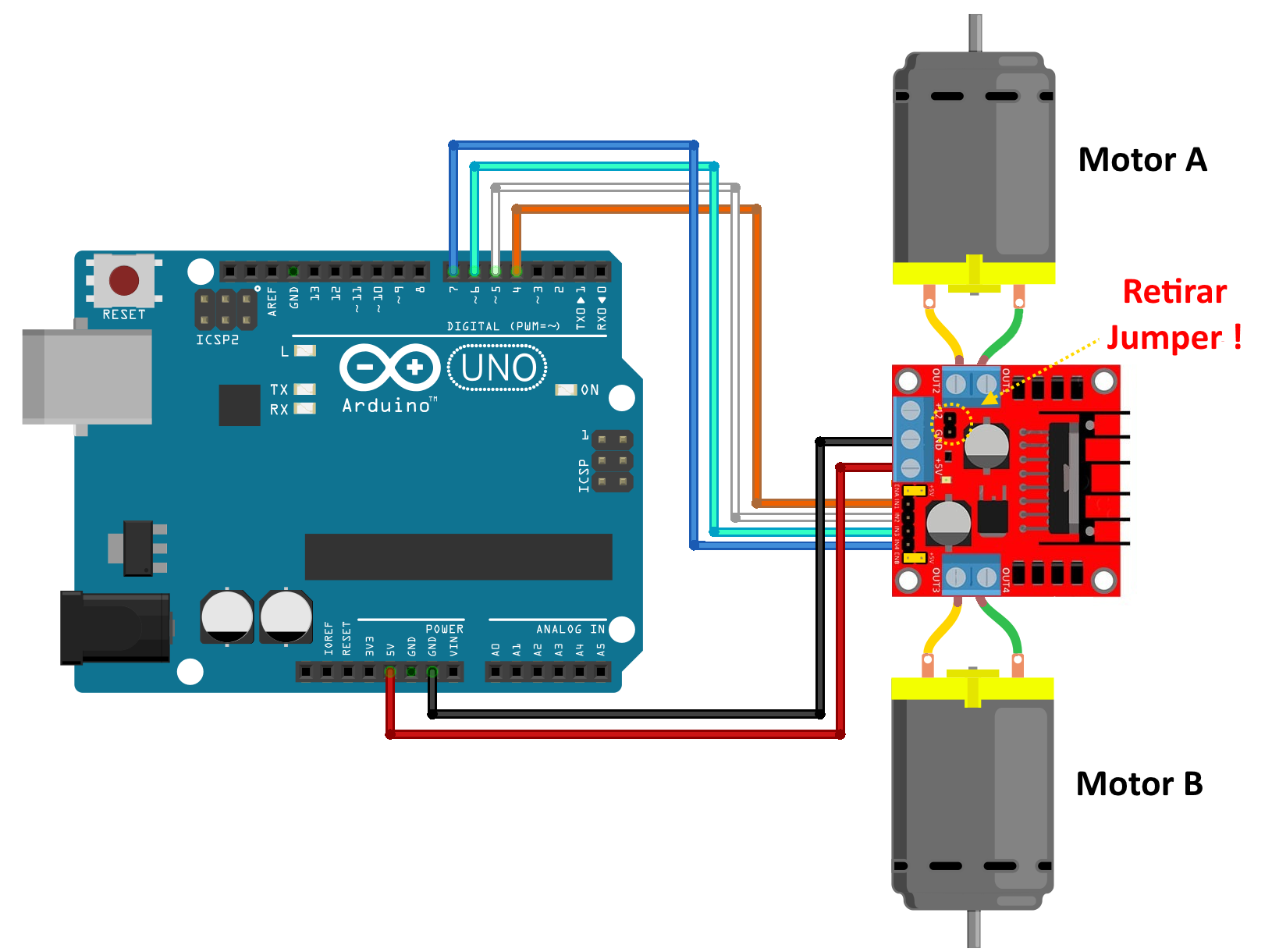
A tabela abaixo mostra a ordem de ativação do **Motor A** através dos pinos **IN1 e IN2**. O mesmo esquema pode ser aplicado aos pinos **IN3 e IN4**, que controlam o **Motor B.**



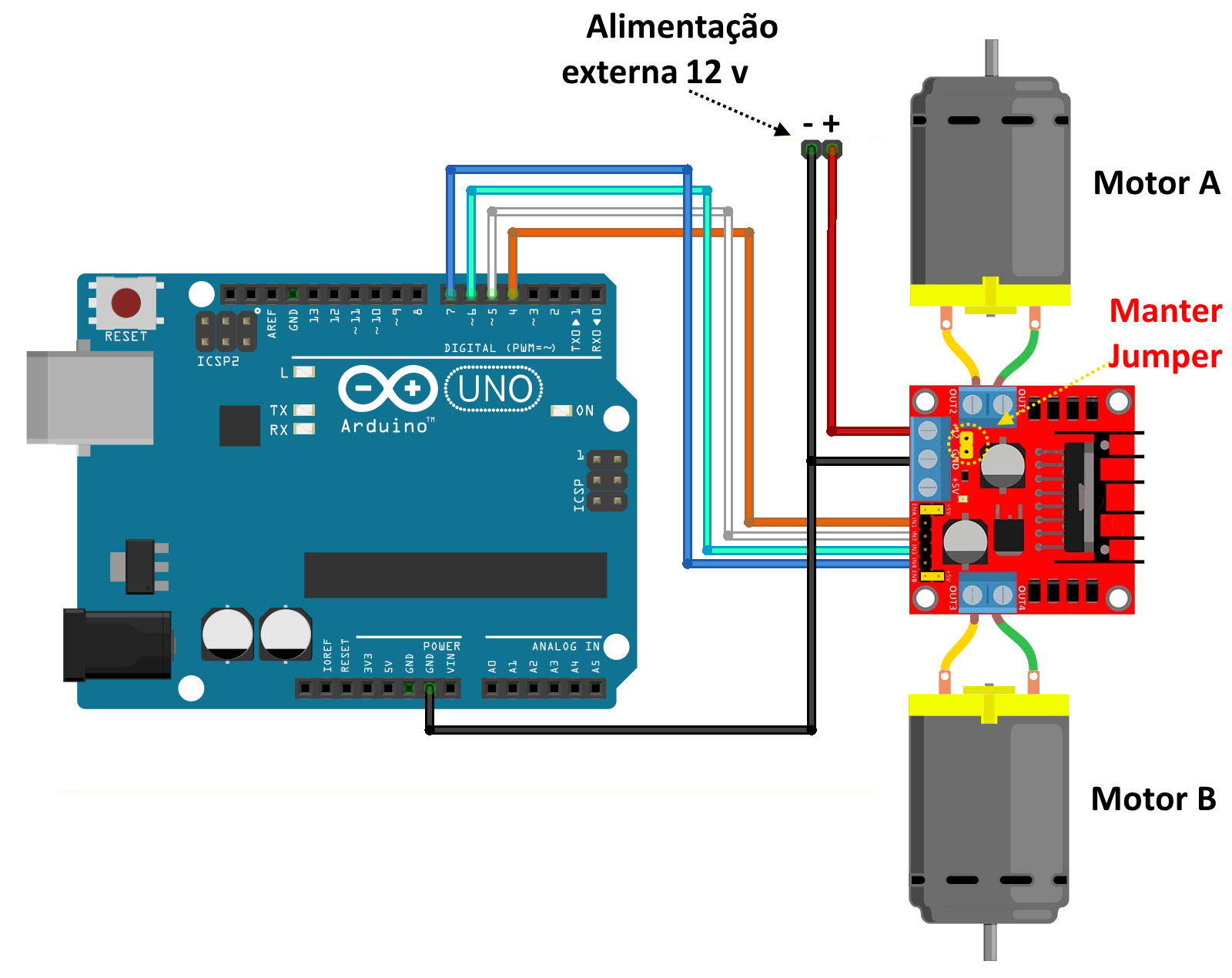
**3.1.2.3 Conexão com o Arduino UNO:**

Vamos mostrar dois esquemas de ligação deste módulo ao [Arduino Uno R3](https://www.filipeflop.com/produto/placa-uno-r3-cabo-usb-para-arduino/).

O primeiro circuito utiliza a alimentação do próprio Arduino, e deve ser feito **sem o Jumper** em **(Ativa 5V)**. Utilizamos **2 motores DC 5V.**



O segundo circuito (*esquema do nosso projeto)* utiliza alimentação externa e **2 motores DC de 12V**. Nesse caso precisamos colocar o jumper em **Ativa 5v:**



**3.1.2.4 Controlando os motores DC:**

Para controlar os motores DC, ligamos cada um dos motores às conexões A e B no módulo L298N. No caso do nosso projeto, estamos usando quatro motores, porém os dois motores de cada lado estão ligados em paralelo entre si, sendo assim o módulo L298N enxerga como se existisse apenas um motor de cada lado. Assegure-se de que a polaridade dos motores é a mesma, caso contrário, é preciso trocá-las garantido que ambos os motores rodam para frente ou para trás e não um para a frente e outro para trás.

Em seguida, ligamos a fonte de alimentação (no nosso caso, uma pilha de 9v): os terminais positivo e o negativo da fonte nos pinos **6-35v** e **GND** do módulo L298N, respectivamente. Utilizamos outra pilha de 9v para alimentar o Arduino UNO. Não esqueça de ligar o GND do Arduino ao GND do módulo L298N para completar a alimentação do circuito.

Agora ligaremos seis pinos das saídas digitais do Arduino, dois destes têm de ser PWM (modulação por largura de pulso) que serão ligados aos pinos **Ativa MA** e **Ativa MB** do módulo L298N para controlar a velocidade dos motores, os outros quatro pinos devem ser ligados aos pinos de entrada do módulo L298N (**IN1, IN2, IN3 e IN4**) para controlar o sentido de rotação dos motores. Os pinos PWM são indicadas pelo til (“~”) ao lado do número do pino, como mostra a figura abaixo:



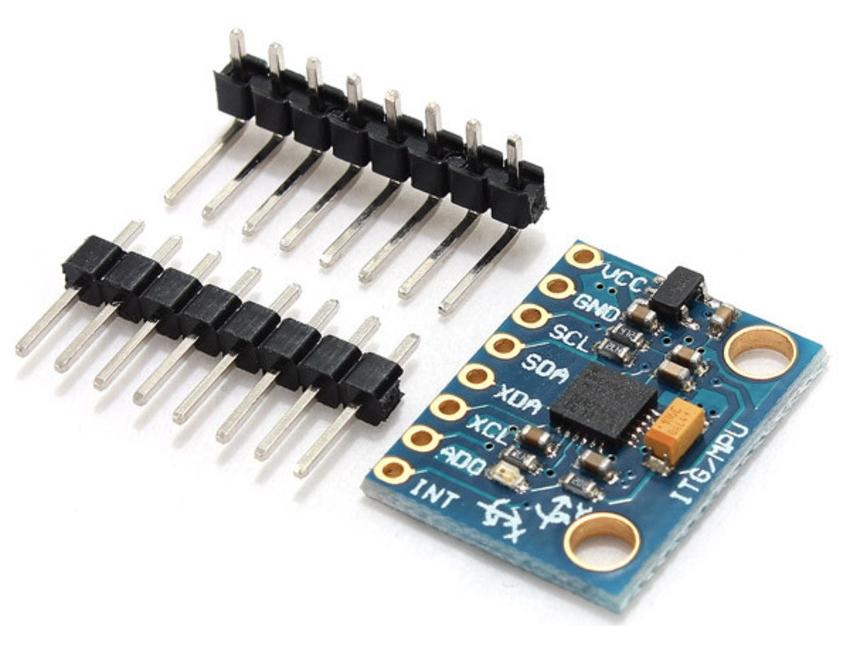
No nosso projeto ligamos os pinos D7, D8, D9 e D10 da saída digital do Arduino aos pinos IN1, IN2, IN3 e IN4 do módulo L298N, respectivamente. Em seguida, ligamos os pinos D6 e D11 da saída digital do Arduino aos pinos Ativa MA e Ativa MB do módulo L298N, respectivamente. Não esquecer de remover os jumpers de Ativa MA e Ativa MB.

A direção do motor é controlada através do envio de um sinal de HIGH ou LOW para cada motor (ou canal). Por exemplo para o motor A, um sinal HIGH para IN1 e LOW para IN2 fará com que ele gire num sentido, já um sinal LOW para IN1 e HIGHT para IN2 fará com que ele gire noutro sentido.

No entanto, os motores não irão girar até que um sinal HIGH seja aplicado no pinos Ativa MA e Ativa MB. Eles podem ser desligados com um sinal LOW aos mesmos pinos. No entanto, se for preciso controlar a velocidade dos motores (*que é o caso do nosso projeto*), o sinal PWM do pino digital ligado ao pino respectivo permite garantir a mesma.

Em seguida, temos que fazer o upload do programa para o Arduino.

**3.2 MÓDULO ACELERÔMETRO E GIROSCÓPIO MPU6050**



Nesse módulo temos em uma mesma placa um acelerômetro e um giroscópio de alta precisão, tudo isso controlado por um único CI, o [MPU6050](https://www.filipeflop.com/produto/acelerometro-e-giroscopio-3-eixos-6-dof-mpu-6050/). Além dos dois sensores, tem embutido um recurso chamado **DMP (Digital Motion Processor)**, responsável por fazer cálculos complexos com os sensores e cujos dados podem ser usados para sistemas de reconhecimento de gestos, navegação (GPS), jogos e diversas outras aplicações. Outro recurso adicional é o sensor de temperatura embutido no CI, que permite medições entre -40 e +85 ºC.

## **3.2.1 Pinagem e endereçamento do MPU 6050**

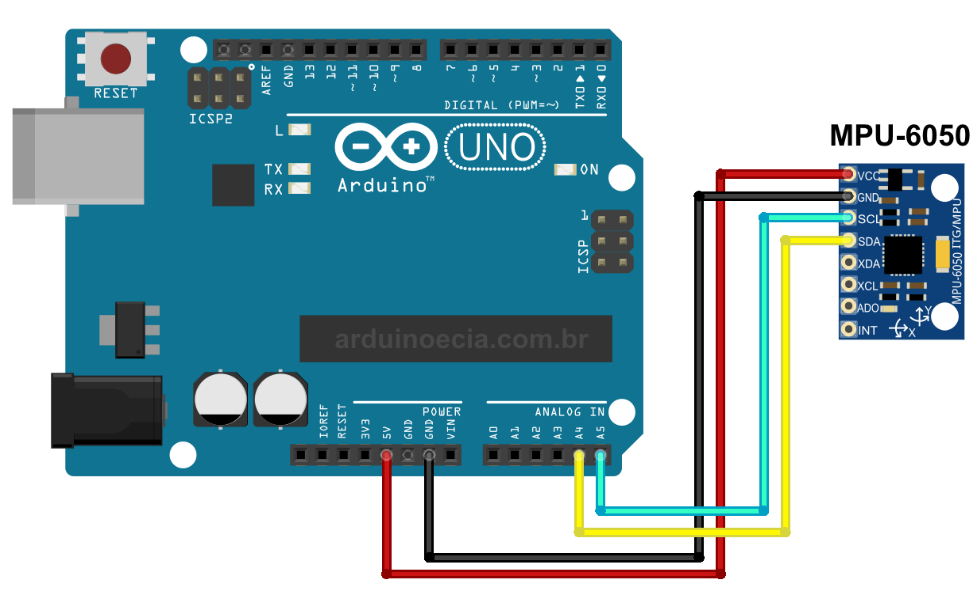
## 

A comunicação com o Arduino usa a interface I2C, por meio dos pinos **SCL** e **SDA** do sensor. Nos pinos **XDA** e **XCL** você pode ligar outros dispositivos I2C, como um magnetômetro por exemplo, e criar um sistema de orientação completo. A alimentação do módulo pode variar entre 3 e 5v, mas para melhores resultados e precisão recomenda-se utilizar 5v. O pino **AD0** desconectado define que o endereço I2C do sensor é **0x68**. Conecte o pino AD0 ao pino 3.3V do Arduino para que o endereço seja alterado para **0x69**. Essa mudança permite que você tenha dois módulos MPU-6050 em um mesmo circuito.

**3.2.2 Conexão com o Arduino UNO:**

Para obter as informações de medição do ângulo de inclinação do veículo, foi utilizado o acelerômetro. Os giroscópios são utilizados para manter ou para medir orientação e medem a velocidade angular do módulo (provavelmente graus por segundo).

A figura a seguir mostra a conexão do módulo MPU6050 ao Arduino.



**3.3 SOFTWARE**

**3.3.1 Programação (Código)**

// Declarando as variáveis do programa

float set\_point = 0;

float angulo\_atual = 0;

int erro = 0;

int Kp=4,Ki=0.01, Kd=0.5;

int Ta, Tatual, Dt;

int Ang\_ant, Ang;

int Erro, SetPoint;

int P, I, D;

int X, Y;

// Ligar os pinos da controladora aos pinos digitais do Arduino

// Motor A (esquerda)

int enA = 6;

int in1 = 7;

int in2 = 8;

// Motor B (direita)

int enB = 11;

int in3 = 9;

int in4 = 10;

// Função criada para o carrinho se movimentar para frente

void frente(int X, int Y)

{

// liga o motor A

digitalWrite(in1, HIGH); // esta função irá fazer rodar os motores para frente

digitalWrite(in2, LOW); // esta função irá fazer rodar os motores para frente

analogWrite(enA, X); // a velocidade é definida para X (intervalo de 0 a 255)

// liga o motor B

digitalWrite(in3, LOW); // esta função irá fazer rodar os motores para frente

digitalWrite(in4, HIGH); // esta função irá fazer rodar os motores para frente

analogWrite(enB, Y); // a velocidade é definida para Y (intervalo de 0 a 255)

}

void setup(){

pinMode(enA, OUTPUT); // definir todos os pinos de saída da controladora dos motores

pinMode(enB, OUTPUT); // definir todos os pinos de saída da controladora dos motores

pinMode(in1, OUTPUT); // definir todos os pinos de saída da controladora dos motores

pinMode(in2, OUTPUT); // definir todos os pinos de saída da controladora dos motores

pinMode(in3, OUTPUT); // definir todos os pinos de saída da controladora dos motores

pinMode(in4, OUTPUT); // definir todos os pinos de saída da controladora dos motores

Serial.begin(115200); // configura a comunicação serial com 115200 bps.

sensor\_setup();

while(first <= 400){

sensor\_loop();

}

}

void loop(){

sensor\_loop();

// Cálculo do PID

Ta = Tatual;

Tatual = millis(); // Esta função retorna o tempo em milissegundos desde quando o arduino foi ligado ou reiniciado

Dt = Tatual - Ta;

Ang\_ant = Ang;

Ang = Euler[0]; // variável que recebe o valor angulo;

Erro = SetPoint - Ang; // setpoint = primeira leitura. angulo inicial

P = Erro\*Kp;

I = I+(Ki\*Erro\*Dt);

D = (Kd\*(Ang - Ang\_ant))/Dt;

X = 125 + P + I + D; // Motor esquerdo

Y = 125 -(P + I + D); // Motor direito

// Filtro para que os valores de X e Y não sejam negativos e nem maiores que 255

if(X>255){

X=255;

}

if(Y>255){

Y=255;

}

if(X<0){

X=0;

}

if(Y<0){

Y=0;

}

frente(X,Y);

**3.3.2 Sistema de Controle**

O sistema de controle tem como principal objetivo, manter o processo dentro dos parâmetros desejados, com o uso de sensores, atuadores e sistemas de computadores projetados para realizar as atividades de maneira segura. Sendo o processo, no ramo industrial, um conjunto de operações realizadas por um determinado equipamento, ou determinados equipamentos, com pelo menos, uma variável física ou química do material. Alguns conceitos importantes a mencionar:

* Variáveis de Processo: São as condições internas ou externas, que influenciam diretamente o processo industrial, sendo necessário realizar o controle das mesmas. Exemplo: Pressão, volume, temperatura, vazão, pH, velocidade, etc.;
* Variável Controlada: Indica, diretamente, a forma desejada do produto;
* Variável Manipulada: Onde o controlador automático atua, a fim de manter, dentro dos parâmetros desejados;
* Valor desejado (setpoint): Valor de referência para a variável, que deseja manter o controle;
* Controle de Malha Aberta: O sinal de saída não exerce nenhuma ação de controle no sistema. Utilizado somente quando não há distúrbios internos ou externos;
* Controle de Malha Fechada: São sistemas com realimentação, tendo o objetivo de minimizar os erros e acertar a saída do sistema, de acordo com um valor desejado.

O algoritmo PID é aplicado, justamente, em controle de malha fechada. Como foi mencionado acima, tem o objetivo de manter a saída dentro de um valor desejado (setpoint).

### 3.3.3 PID (Proporcional Integral Derivativo)

O Controle PID é um algoritmo utilizado em controle de processos, de forma a deixar o mesmo muito mais preciso. Unindo as ações proporcional (minimiza o erro), integral (zera o erro), derivativo (antecipa o erro), controlando uma variável de processo.

Um controlador proporcional (Kp) terá o efeito de reduzir o tempo de subida e reduzirá, mas jamais eliminará, o erro em regime permanente. No modo Proporcional, o controlador simplesmente multiplica o Erro pelo Ganho Proporcional (Kp) para obter a saída do controlador.

Um controlador integral (Ki) terá o efeito de eliminar o erro em regime permanente, mas ele deixará pior a resposta transiente.

Adicionando a ação derivativa pode permitir que você tenha maiores ganhos de P e I e ainda manter o loop estável, dando-lhe uma resposta mais rápida e melhor desempenho de loop. Se você pensar nisso, a ação Derivativa pode melhorar a ação do controlador porque ela prediz o que ainda está por acontecer ao projetar a taxa atual de mudança para o futuro. Isto significa que não está sendo contabilizado o valor medido atual, mas sim um valor de medição futuro.Um controlador derivativo (Kd) terá o efeito de incrementar a estabilidade do sistema, reduzindo o overshoot, e melhorando a resposta transiente.

**3.3.4 Testes e Ajuste das Variáveis**

Primeiramente consideramos o Kp igual a 10 e zera-se Ki e Kd. Observou-se que o carrinho tem uma resposta com uma boa velocidade. Com os vários testes vimos que a velocidade do carrinho estava diminuindo e então percebeu-se que era por conta das baterias de 9 volts estarem descarregando.

Após a troca das baterias, vimos que o carrinho ficou mais potente. Isto fez com que o Kp diminuísse para 4. Aumentamos o Kp apenas para ver o seu efeito no sistema e vimos que quando o seu valor é muito grande, a variável de processo começará a oscilar. Se Kc é aumentado ainda mais, as oscilações ficarão maior e o sistema ficará instável e poderá oscilar até mesmo fora de controle.

Já com o valor de Kp novamente definido, tivemos que ajustar o valor de Ki. O primeiro valor de Ki foi 0.01, observou-se que a resposta ficou mais lenta, porém com menos oscilações.

Uma vez que o P e I foram definidos para que o sistema de controle seja rápido com o steady state (é a diferença final entre as variáveis do processo e do set point.) mínimo e constante, o termo derivativo é aumentado até que o loop seja aceitavelmente rápido em relação ao seu ponto de referência. Aumentar o termo da derivada diminui o overshoot, aumentando o ganho, mantendo a estabilidade e ainda fazendo com que o sistema seja altamente sensível ao ruído. Portanto definimos o valor de Kd sendo 0.5.

**4. CONSIDERAÇÕES**

**4.1 APLICAÇÃO**

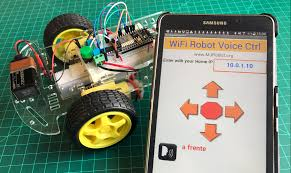
VEÍCULO AUTO EQUILIBRANTE

****

Utilizando-se do conceito de pêndulo invertido, o veículo auto equilibrante é capaz de transportar indivíduos, incluindo os com deficiência locomotora, com total segurança e conforto, buscando a máxima acessibilidade e ampla utilização do mesmo, a fim de atenuar ou até mesmo eliminar uma deficiência utilizando-se de tecnologia. O projeto está estruturado com uma metálica, com um sistema elétrico e eletrônico, que faz referência ao armazenamento de carga, aos sensores e ao sistema de atuação e o sistema de controle, que limita o controle de velocidade, aceleração e posição do protótipo.

**4.2 MELHORIAS**

**4.2.1 Aplicativo de controle pelo celular**

****

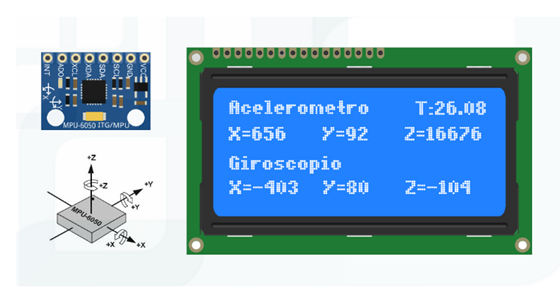
**4.2.2 Módulo Bluetooth RS232 HC-05**

uma solução simples e barata de comunicar um microcontrolador seja ele Arduino, Raspberry, Pic ou outro com a interface bluetooth do seu dispositivo com alcance de até 10 metros.

****

**4.2.3 Display LCD 20×4**

Vamos utilizar um [Display LCD 20×4](https://www.filipeflop.com/produto/display-lcd-20x4-backlight-azul/) para mostrar os valores lidos do [MPU6050](https://www.filipeflop.com/produto/acelerometro-e-giroscopio-3-eixos-6-dof-mpu-6050/) em conjunto com um [Arduino Uno](https://www.filipeflop.com/produto/placa-uno-r3-cabo-usb-para-arduino/). Nas duas primeiras linhas do display, além da temperatura no canto superior direito, temos os valores de X, Y e Z para o Acelerômetro, e nas duas últimas linhas os valores de X, Y e Z para o giroscópio.



**4.3 CUSTOS**

**4.3.1 Kit Arduino**

O Kit Arduino nível zero é composto por todos os componentes mais básicos para um sólido aprendizado dos fundamentos de Arduino, como: LEDs, botões, LDR, NYC, TDRT5000 entre outros, além do próprio Arduino UNO, cabos e um organizador para seus componentes. Esse kit fornece o material necessário para elaboração de diversos projetos básicos.

*Valor: R$ 119,90*



**4.3.2 Kit Chassi 4WD**

O Kit Chassi 4WD é composto por:

02 - Chassi em acrílico  
04 - Motores DC   
04 - Rodas  
04 - Discos de Encoder  
06 - Espaçadores  
01 - Jogo de Parafusos e Porcas

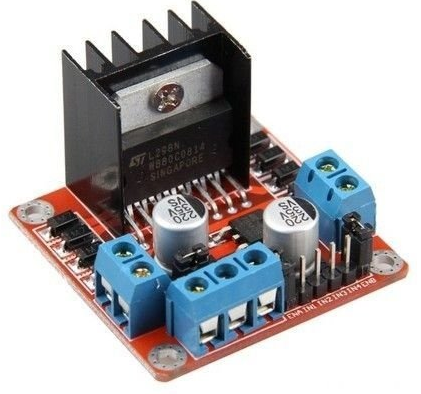
*Valor: R$ 129,90*



**4.3.3 Drive Ponte H L298N**

Este Driver Motor Ponte H L298N tem como principal função o controle de cargas indutivas como: motores DC, solenóides, relés etc. Com este driver é possível controlar independentemente a velocidade e rotação de dois motores DC.

*Valor: R$ 26,90*



**4.3.4 Módulo Acelerômetro e Giroscópio MPU6050**

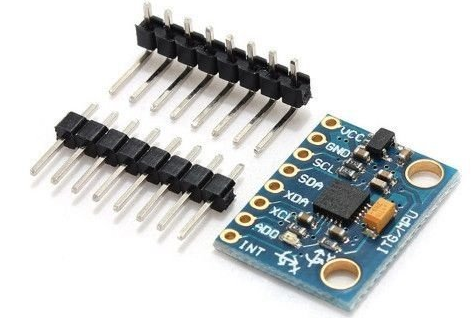
O Módulo Acelerômetro e Giroscópio 3 Eixos MPU6050 acompanha:

01 - Acelerômetro e Giroscópio 3 Eixos MPU6050

01 - Barramento de Pinos 90 Graus

01 - Barramento de Pinos 180 Graus

Valor: R$ 22,90



**4.3.5 Pilhas 9V**

Precisamos de duas pilhas 9 volts alcalina. Uma está alimentando a ponte H e a outra o Arduino.

*Valor: R$ 40,00*



Custo total de material: R$ 379,50

Estimamos este mesmo valor relacionado ao tempo dedicado e à mão de obra para a montagem, programação e testes do projeto.

Um exemplo de produto que o nosso projeto poderia se tornar, com as devidas melhorias e adaptações, é o citado na aplicação, **veículo auto equilibrante,** com o valor médio de mercado em torno de R$ 2.000,00. Considerando que investiríamos mais algo em torno de R$ 600,00 nas melhorias e adaptações, totalizando um valor final para o produto de R$ 1.359,00. Teríamos, então, lucro de 47%.

**5. CONCLUSÃO**

Neste projeto conseguimos alcançar o objetivo de fazer o controle de trajetória de um carro robô usando a ferramenta base controlador PID.

O projeto utilizou somente componentes de baixo custo, com um ótimo desempenho fazendo com que o controle de caminho se mostrasse de bastante valia e preciso.

Acreditamos ser necessário em um próximo momento estudar possíveis técnicas de controle que possam substituir ou atuar em paralelo ao PID de forma a otimizar o equipamento aqui proposto.

Por tal motivo acreditamos que esse projeto deve ter continuidade tendo em vista as possíveis aplicações que ele pode proporcionar.