



WWW.casemonstro.com.br

Robótica Educacional

Por Alexandre Ferreira - Casemonstro

Para leigos - Antes de começar a leitura desse material, recomendo que o leitor se desprenda de alguns conceitos sobre Robótica, analise como uma informação que você está tendo acesso agora.

Para os Maker e afins, já familiarizados com conteúdo – Veja esse texto como uma forma de abordar o conteúdo com alunos.

Quando falamos de Robótica, em si a grande maioria associar essas palavras a Filme, Televisão, coisas que apenas podemos ver nos Cinemas, pois bem esse material é voltado a quebrar esse tabu, a mostrar para todos o quanto a Robótica é acessível e de baixo custo.

Quando vamos falar sobre Robótica para alunos (qualquer idade) a melhor forma de abordar isso é apresentando um Robô construído por nós e a medida que o apresentamos, vamos familiarizando nossos intrépidos ouvintes com esse novo universo que se faz diante de seus olhos.



Aqui ao lado temos meu Pequeno Mini-Dalek, sou fã de filmes, animes e seriados e tiro algumas inspirações desse universo, esse pequeno robô que uso para fazer esse primeiro contato com os alunos. Quando mostro que ele é controlado com controle de vídeo-game (ps2) os alunos ficam entusiasmados, depois vou explicando cada uma de suas funções, como acender o olho, mover o braço, acender o laser (luz ultravioleta), mobilidade, Voz e iluminação da Orelha.

Em geral as pessoas gostam de vê-lo e saber o que constitui ele,uento que ele foi modelado em papel, dobradura, depois endurecido e pintado.

Porem o que surpreende as crianças são os detalhes:
Olho, foi feito com a casca de um OVO, cozинhei o mesmo(comi), limpei o interior da casca, lixei dei acabamento, apliquei material químico(plástico líquido), para fortalecer o olho e adicionei os detalhes que ele tem assim como o LED (luz pequena).

Os Braços são feitos com varetas de barraca, um dos braços tem 4 haste longas, estas foram feitas com varetas de guarda-chuva.

Na base de cada braço dele, tem uma mini-esfera de bilhar de numero 8.

Seus motores são motores de vidro elétrico, sim isso mesmo.

Suas rodas, são madeiras cortadas com serra copo, e aproveitadas de forma a se tornarem as rodas desse meu mini-Dalek.

Claro que não é simplesmente contando que usei um ovo para fazer o olho do Robô, em geral conto minha experiência, pois tentei modelar o olho 2 ou 3x até me dar conta que não estava mesmo ficando bom. Nesse momento decidi parar, esfriar a cabeça tomar um café e depois iniciar do zero:

Eu precisava fazer uma peça oval, ai comecei a buscar referencias, oval, oval, ovo, e cheguei a idéia de usar o próprio ovo.

A medida que faço essa apresentação do meu robô e seus componentes comuns para que seu corpo tenha aquela aparência, despertei mais ainda o interesse dos alunos em saber sobre Robótica, porem agora com uma informação adicional a esse querer aprender, “eu posso” pois ao ver a simplicidade dos materiais que usei elas chegam ao conceito que elas também podem.

Depois dessa apresentação da aparência do meu Robô, apresento o que torna ele possível, a Robótica usada, e nesse momento que apresento os componentes, comento o preço de alguns para mostrar que são acessíveis.

C o m p o n e n t e s

Ao apresentar a identidade do Robô, agora apresento seus componentes e suas respectivas funcionalidade, começando pelo mais comum e que talvez eles já tenham tido acesso:

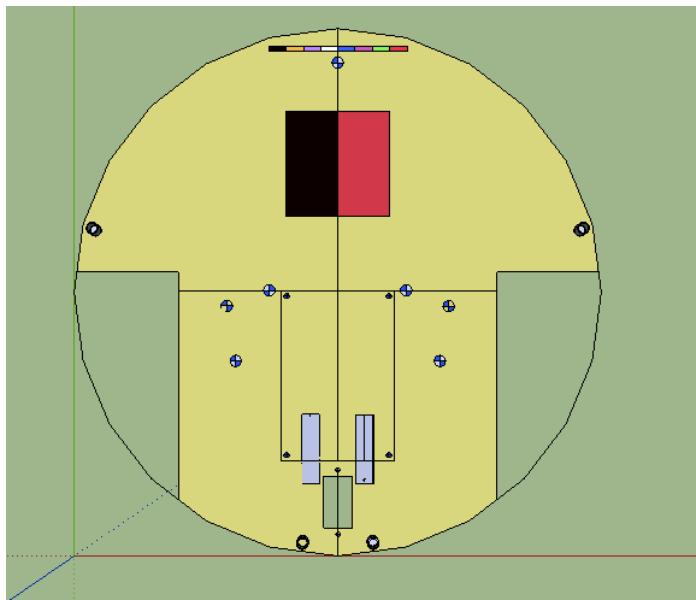
- Chassi
- Motor
- Baterias
- Leds
- Fios
- Arduino (esse deve ser a novidade)
- Ponte H (esse aqui também)

Quando falamos de Robôs de Mobilidade, usados com fins educacionais, esses são os componentes mais comuns, porem agora irei detalhar cada componente desse de forma lúdica.

C H A S S I

Quando falamos de Chassi falamos sobre a Base do Robô, sobre o que ele será construído. Essa base tem a função primordial de sustentar nossos componentes, com firmeza sem amassar/empenar e depois sobre esse chassi iremos criar uma identidade

para o robô. Para as aulas eu uso um círculo em madeira de 25cm, com alguns cortes na lateral, local destinado as rodas, esse formato se assemelha a um Cogumelo.

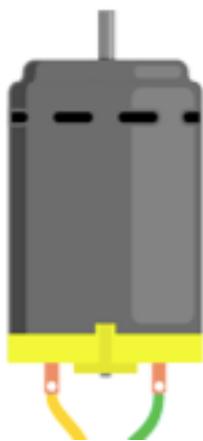


Nessa base, iremos colar os Motores, fixar o Arduino, Bateria e Ponte-H, com isso tenho tudo envolvido na mobilidade do meu Robô, porem ainda sem identidade.

Necessário compreender que até esse momento o Robô ainda não tem a Lógica para seus movimentos ele apenas está montado fisicamente.

M O T O R E S E R O D A S

Bom motores, em geral fazem parte do universo de todos, em algum momento uma criança já desmontou um brinquedo seja um carrinho ou até mesmo uma boneca que tem movimento e já viu esses famosos motores.



Aqui podemos ver o Motor (ilustrativo), seu eixo de movimento (eixo) e na sua parte de inferior 2 cabos um Verde e outro amarelo, saindo do motor.

Agora vamos entrar em alguns conceitos interessantes, mas comuns a crianças e adultos.

O motor quando ligado em uma bateria ele irá girar em um sentido, porem se invertermos os fios em relação à bateria ele irá girar sentido Contrário.

Bom vamos observar a imagem, se ligarmos:

Amarelo, "Positivo da bateria
Verde "Negativo" da bateria

Esse motor ira girar sentido Horário.

Porem se pegarmos o mesmo motor e trocarmos a posição de seus fios na bateria:
Amarelo, "Negativo da bateria
Verde "Positivo"da bateria
Esse motor ira girar sentido Anti-Horário, inverso para o outro lado.

Uma coisa interessante é que esse motor do exemplo, ele é um motor liso, ele não possui nada de diferente ao corpo do Próprio motor, porem em si nos robôs de mobilidades nossos motores possuem um formato que soma ao seu corpo um sistema de engrenagens, para dar força ao motor. Para que ele possa mover um peso maior.

B A T E R I A S

Quando falamos sobre Motores, falamos sobre baterias(powerbank), nada mais justo que explicar um pouco mais sobre elas. Bom, baterias são fontes de energia, porem não qualquer tipo de energia, no caso das baterias chamamos de Energia Continua, pois elas possuem dois pólos, um Positivos e o outro Pólo Negativos, precisamos compreender bem esses conceitos pois eles vão nos guiar daqui em diante.

As baterias, sejam elas de 1volt (volts, medida usada para baterias ou outras formas de energia elétrica) ou falamos de baterias de 12volts, sempre teremos polo Positivo e outro inverso Negativo, porem em alguns circuitos essas informações podem aparecer descritas de varias formas:

Positivo ➔	1,5v	3,7v	5v	12v
Negativo ➔	-	GND	Ground		

O positivo pode ser lido ou apresentado em qualquer valor maior que zero.
Já o Negativo, pode ser lido apenas como “- “ou GND, Ground.

Cada equipamento elétrico tem um limite no qual ele pode ser ligado, precisamos observar bem isso afim de evitar queima de equipamentos.

Esse conceito de Positivo e Negativo é muito importante para o desenvolvimento de nosso robô de Mobilidade. Precisamos deixar isso bem claro sem duvidas.

L E D S
Nossa famosas "luizinhas" pequeninas quase do tamanho de um feijão e que ascendem, bom os famosos leds, estão presentes em 100% de qualquer brinquedo que use pilhas/baterias. Eles são famosos, hoje presentes até mesmo em nossas televisões, lâmpadas de casa, etc.



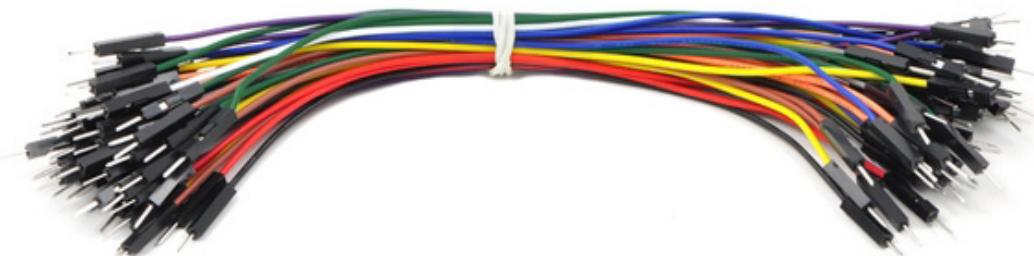
Essas pequenas lâmpadas são usadas para iluminar, novidade né, sendo uma das primeiras lições que aprendemos quando começamos a trabalhar com robótica, e todo o desenvolvimento do nosso Robô de mobilidade será baseado nesse componente, LED.

Ele é formado por 2 terminais, em geral um menor que o outro, e por lógica quando queremos acender o led ligando ele em uma bateria, o terminal menor é o NEGATIVO (-) e o terminal maior o POSITIVO. Se ligar invertido ele não irá brilhar.

Aqui vale observar que cada Led (cor e tamanho) aceita um valor Máximo de Volts ligado em seus terminais, o Branco (incolor) pode até ser ligado direto no Arduino que não queimam, porém os demais pedem um cuidado maior (em geral precisa ligar resistores, que resistem a passagem da corrente elétrica, protegendo o LED).

F I O S

Esse parece ser extremamente simples, até mesmo de explicar, porém ele merecer ser dito e bem orientado sobre eles. Os fios são responsáveis em conectar dispositivos diferentes por meio de jumpers ou conectores, sua função é bem simples levar sinal elétrico ou de comando para algum outro local. Porem uma grande dica sobre isso, crie alguns padrões, algumas regras sobre o que significa cada Cor de fio e isso irá te ajudar a compreender melhor a organização dos cabos dentro do seu Robô.



Algumas sugestões que dou:

Preto, ser o NEGATIVO/GND
Vermelho, Positivo, referencialmente 5V
Amarelo, Positivo 12v ou 9v
Demais cores para Comando.

Os Cabos podem ser de 2 tipos:

Macho
Fêmea

Para poder ligar eles ao Arduino, o cabo precisa ser Macho, e para ligar na Ponte H, são 4 fêmeas (comando) 2 Machos (elétrica).
Nesse caso ponte H vermelha de 4cm x 4cm.



Para Ponte H pequenina serão todos os 6 cabos fêmeas.

A R D U I N O

Bom chegamos ao famoso ARDUINO, mas, o que é o Arduino? De forma simples e dinâmica posso dizer que o Arduino é o cérebro do ROBO, é nele que gravamos todos os comando que queremos que o Robô Execute, ou seja ele é o “Elo” de ligação entre Eu pessoa + meu Computador ++ Arduino, cérebro do Robô e seus componentes. Ele que recebe a programação que faço no computador e envio para o Robô.

Como ocorre toda essa comunicação entre Arduino e meu Computador e depois do Arduino para com as peças ligadas nele?

Bom é simples, o Arduino tem uma conexão USB que serve para ligar ele a um computador, ou mesmo para ligar ele a uma Powerbank(bateria portátil para carregar celular). E no Arduino podemos ver essas barras laterais que possuem marcações numéricas, do lado esquerdo inferior, vai de **A0 a A5**, do lado direito inferior começa em **0 e vai até o 13**, depois do lado direito tem o GND e AREF.

Ainda do lado esquerdo:

Vazio, depois **IOREF, Reset, 3,3v, 5v, GND, GND, Vin.**

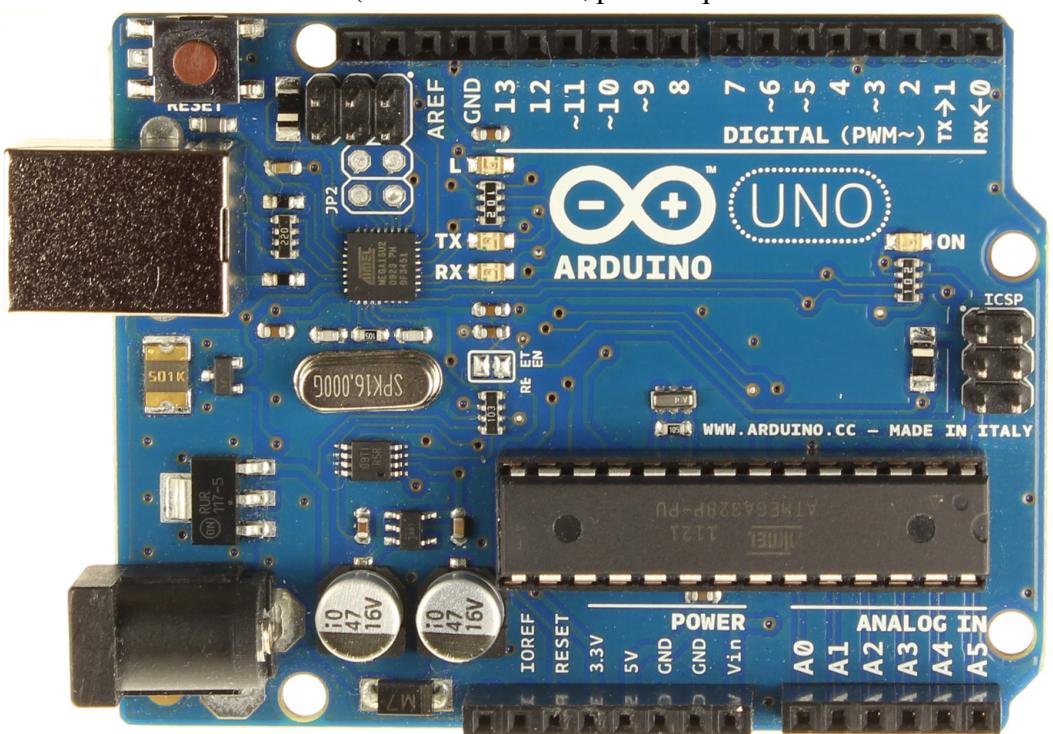
Os Pinos de A0 a A5 e 2 a 13, são usado para comandos, ou seja nesses pinos que ligamos diversos componentes que serão controlados pelo Arduino. Porem claro não apenas de comando os componentes funcionam, por isso temos os pinos relacionados a ELETRICA:

Positivo(3,3v e 5V) e GND. Pois se o componente pedir uma alimentação, essa pode ser retirada do Arduino, nesses locais. Em breve entro em mais detalhes sobre Isso.

Ainda falando sobre os Terminais do Arduino (conexões fêmeas, para as quais conectamos os cabos Machos) temos as opções de comandos e elétrica.

Em si, como funciona isso?

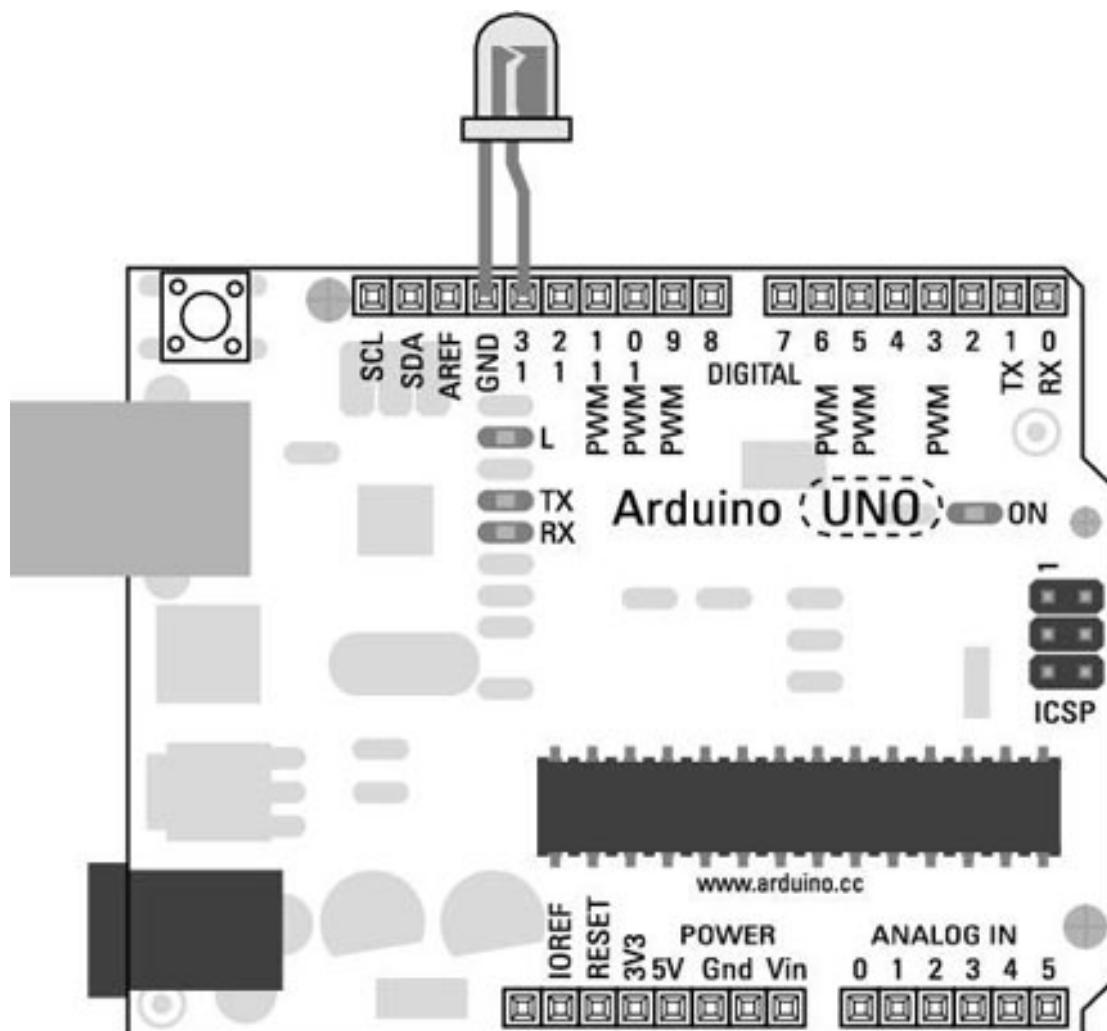
Vamos supor que eu quero acender um LED, sim exemplo do Led, o Led tem 2 terminais, maior (positivo) e o Menor (negativo) porem se você



ligar o led no Arduino em +5v e GND o led fica acesso direto, então como podemos controlar o LED?

Utilizando o GND de qualquer fêmea e ligando o Pino + do LED em alguma fêmea dos terminais de Comando.

Ex.



No código a ser escrito eu irei dizer que quero que o Led13 acenda e apague. Ele irá funcionar pois quando dou o Comando Acesso, o pino 13 sai de 0V e vai até 5v, e como ele já tem o GND ligado isso é o mínimo para alimentar eletricamente o LED e ele acenda, a vantagem é que no Código posso dizer quanto tempo quero ele acesso ou apagado.

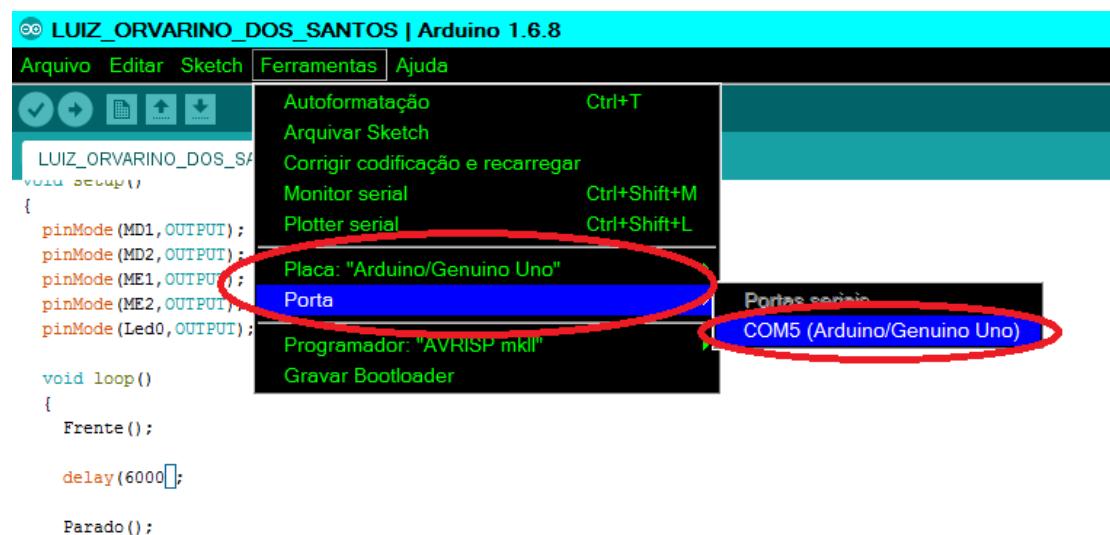
Em breve irei falar especificamente sobre esse código.

Olhando o desenho do Arduino podemos ler que alguns terminais são marcados pelas letras PWM, ou seja apenas eles podem ser usados para controle de Velocidade por isso a indicação PWM. Outra indicação importante são as portas Digitais e as portas Analógicas, que diferenciam alguns comandos.

Podemos definir que os comandos dados nos terminais do Arduino fazem uma variação de 0 a 5v de saída nos terminais, ou podem, ler uma variável de entrada neles.

Antes de encerrar o tema Arduino, vou explicar sobre o Programa Arduino e sua comunicação com a placa física Arduino.

O modelo de placa que nos utilizamos se chama “Arduino UNO” e no programa Arduino, sempre que vamos configurar a placa, precisamos ter certeza que a placa selecionada é a correta, e que a porta de Comunicação (COM) esta correta também.

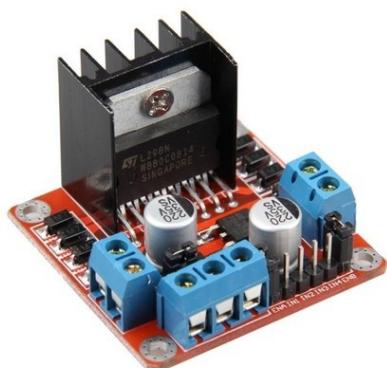


P O N T E H

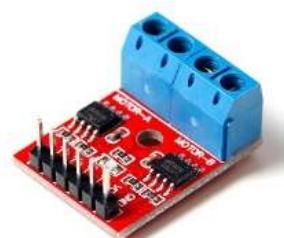
A Ponte H é uma interface inteligente que comprehende os comandos dados pelo Arduino e em consequência disso inverte ou para o giro do Motor.

Ou seja por meio de comandos você define para que lado o Motor deve girar, ou se ele deve estar parado, existe também a condição Freio, onde ele pode parar em uma rampa que a roda não vai girar.

Existem 2 modelos de Ponte H que utilizamos nesse pequeno modelo de Robô. Ambas funcionam da mesma forma para Comandos + Alimentação elétrica.



Modelo L298N



Modelo 800ma

A diferença real entre as 2 é que a maior tem a capacidade de trabalhar com motores maiores, sem muito esforço e com um consumo de no Máximo 2 Amperes. Mas para nossos pequenos robôs, elas servem sem dúvida.

O que realmente as placas fazem?

Lembrando do Motor, nos temos 2 pinos em cada motor, para que o Motor gire precisamos ter + e - ligados ao motor e ele irá girar, se invertermos a elétrica - e + ele irá girar em sentido contrário, pegando esse raciocínio podemos dizer que a Ponte H tem 2 pinos para cada Motor e conforme definimos no comando ele será Positivo ou Negativo, dando comando no pino de HIGH ou LOW para isso:

HIGH = Alto = +

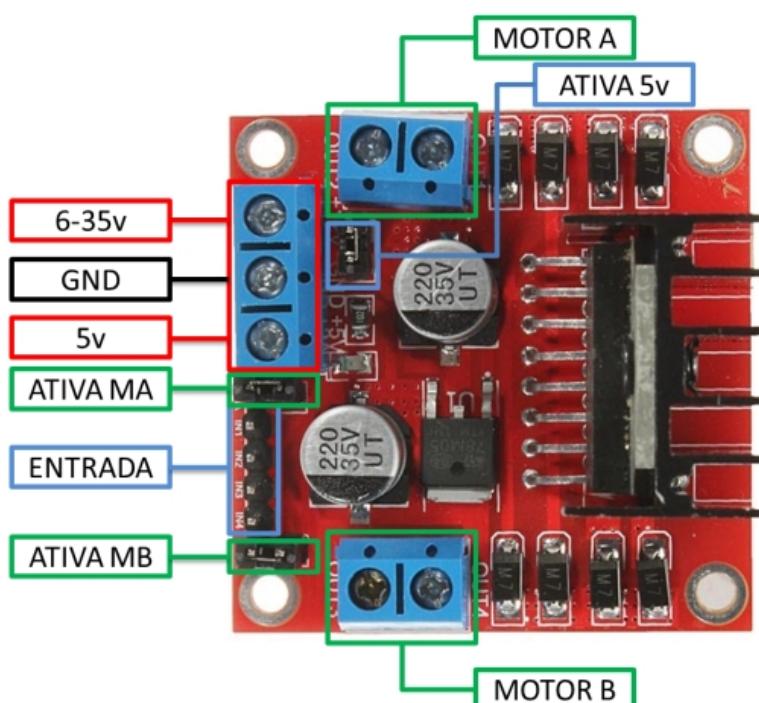
LOW = Baixo = -

Os 2 pinos de comando para cada motor. Os comandos variam de HIGH e LOW e suas combinações geram os seguintes movimentos no Motor.

Pino 1	Pino 2	Sentido
HIGH	HIGH	Freio
LOW	LOW	Parado
HIGH	LOW	Frente
LOW	HIGH	Ré

Agora que vimos um exemplo de combinação de comandos vamos compreender melhor a placa:

Esse modelo de Ponte H é alimentada com uma variável de 5 a 35v no conector marcado como 6-35v, o conector de 5v não é utilizado nesse exemplo de Robô.



Onde temos definido como “Entrada” é onde ligamos os 4 pinos de comando dos motores, 2 para cada motor.

A Indicação “Motor A” e “Motor B” é onde ligamos os fios dos Motores.

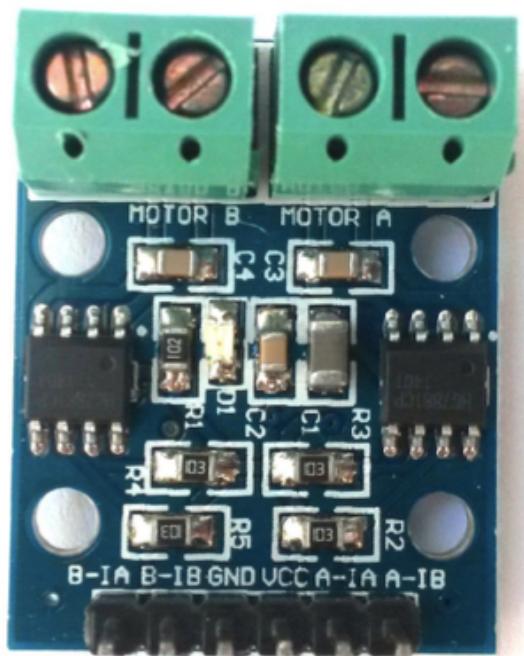
Os “Ativa MA, Ativa MB” permanecem como estão.

Para ponte H pequena ela não tem jumpers e seus Pinos são:

B-IA: Pino 1 do Comando motor B
B-IB: Pino 2 do Comando motor B

A-IA: Pino 1 do Comando motor A
A-IB: Pino 2 do Comando motor A

VCC: + Positivo
GND: - Negativo

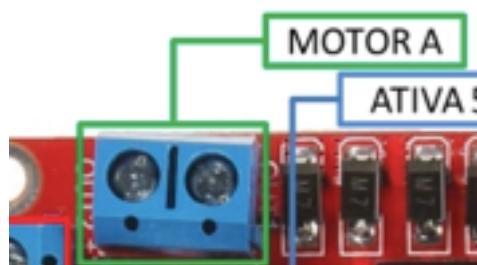


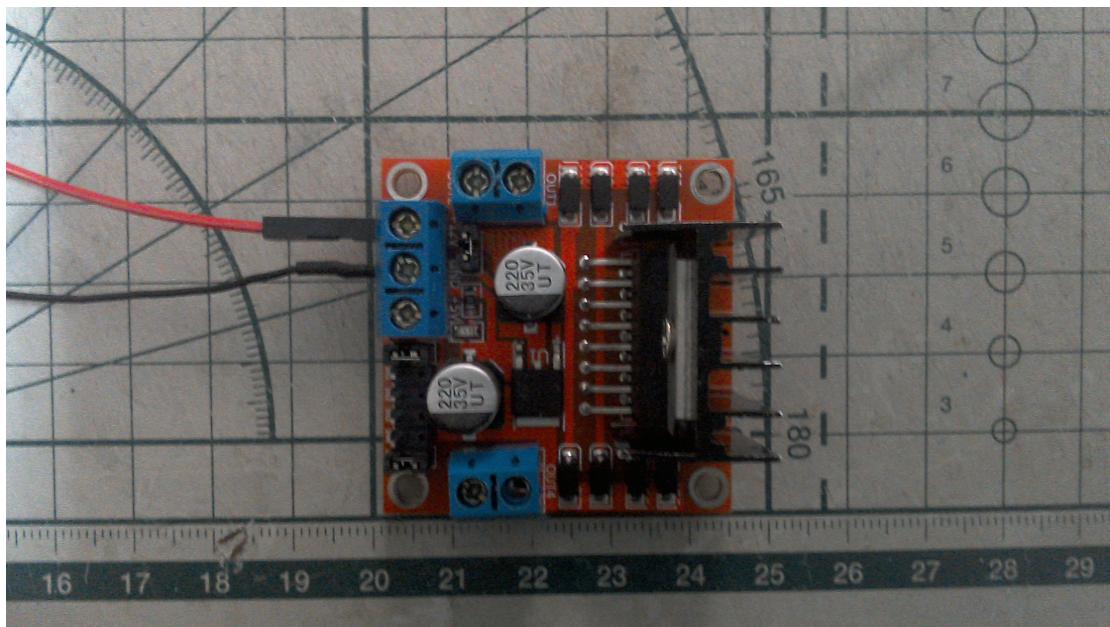
E os conectores maiores cada um respectivo para um Motor (e seus 2 pinos), se o comando Frente mandar a Rodo irar para traz é aqui que você troca o Fio de posição e concerta o movimento.

(Sim é possível corrigir isso no código também, porem vamos adotar a definição escrita aqui como correta e se necessário invertemos os cabos dos Motores aqui entre os fios do mesmo motor:



Para a placa, Ponte H maior isso é feito onde ele Define MOTOR A e o MOTOR B:





Nessa foto acima, é possível ver onde ligamos a alimentação na Ponte H, sendo:
Fio Vermelho, +5v
Fio Preto, Negativo –

Existem alguns Jumper na Ponte H, nenhum deles deve ser movido.

Após especificar todos os componentes que utilizamos, agora iremos conversar sobre a Programação, essa parte é bem dinamica, ela deve ser apresentada de forma clara, não podemos deixar duvidas no ar sobre isso.

Quando falamos em Programação, falamos sobre como funciona o Código para controlar os Robos de Mobilidade. Para melhor conhecimento do Código, costumo dividir isso em 3 partes e apresento elas de forma clara (em português) em momento algum cito uma linha de código, apenas passo a “Logica” é o mais importante nesse momento é fazer com que cada participante da Oficina entenda a ”Logica” depois fazemos analogia com o código real.

Para falar sobre programação, precisamos antes conhecer o programa Arduino, leva o mesmo nome da placa. Devemos lembrar que ao baixar esse aplicativo “Arduino” e após instalar ele, se ao ligar a placa Arduino no computador o Windows não identificar o drive para a placa, você pode executar o processo de instalação e apontar a pasta do programa Arduino, como origem do Driver para completar a instalação da placa. Site do arduino WWW.arduino.cc

Para compreender bem o Programação/Código, dividi ele em 3 partes:

Apresentação

Complemento

Comandos

APRESENTAÇÃO

No inicio da programação é o momento que apresentamos ao Robô quais componentes compõem o Robô, quais componentes serão controlados pelo Robô (no caso o Arduino). Por exemplo se é um Robô de mobilidade, ele irá possuir motores, que serão controlados, então nesse momento iremos apresentar e dar nomes a cada pino do motor.

Ex:

Nome que vou dar	Pino do Arduino	Pino da Ponte H
<i>Mdir1</i>	8	<i>IN1</i>
<i>Mdir2</i>	9	<i>IN2</i>
<i>Mesq1</i>	10	<i>IN3</i>
<i>Mesq2</i>	11	<i>IN4</i>

Apos definir os nomes, vem a apresentação escrita.

Iniciando Mdir1, pino 8

Iniciando Mdir2, pino 9

Iniciando Mesq1, pino 10

Iniciando Mesq2, pino 11

Agora que apresentamos, o que será controlado, vem a parte que chamamos de Complemento.

COMPLEMENTO

Aqui iremos informar se esse dispositivo que apresentamos, funciona como entrada ou saída de informação. E como é que isso funciona, isso é comparável com os humanos com os Sentidos, Visão, Tato, Paladar, Audição e Olfato. Ou seja esses sentidos são responsável pelo corpo humano receber informações. Quando por exemplo nossa fala, emite sons, já com os 5 sentidos, recebemos.

E no nosso Robô precisamos informar se aqueles que apresentamos são de entrada ou saída de Informação.

Apresentado	Condição
Mdir1	Saída
Mdir2	Saída
Mesq1	Saída
Mesq2	Saída

Lembrando que as condições são de Entrada/Saída apenas.

COMANDOS

Aqui iremos falar sobre como ocorre os Comandos, logo apos apresentarmos / Completarmos, agora é a vez dos Comandos, o que queremos que o Robô faça. Quando falamos de Comandos, precisamos compreender duas coisas:

Comandos de Repetição;
Comandos de Atalho;

Bom vamos por partes, quando falamos de **Comandos de Repetição**, falamos exatamente isso, comandos que ao terminar de executar todos, o Robô irá reiniciar os comandos em um Eterno Loop. Se pensarmos em uma lista com 10 comandos, que inicia em 1 e acaba no 10. Quando ele chega no 10 e termina, ele volta a iniciar a lista novamente.

Pula
Agacha
Anda
Para

Nessa lista de 4 comandos, tão logo ele execute a ação “Para” ele irá retornar a 1º Ação, “Pula” e continuara a executar a lista e ficara repetindo.

Pula
2 Segundos
Agacha
1 Segundo
Anda
3 Segundos
Para
5 Segundos

Em geral quando falamos nessas ações/comandos cada uma das ações esta associada com “TEMPO” quando tempo cada ação deve ser feita.

Ou seja o Comando de repetição esta relacionado a Ação e Tempo.
Quando falamos do Comando “Ação” ela esta relacionada .

No Exemplo acima, quando digo “Pula, 2 segundos” eu estou usando um “Comando de Atalho” pois eu não disse que era para me abaixar pegar impulso e depois pular o mais alto que pudesse. Ou seja:

Comando completo	Atalho
<i>Abaixar pegar impulso e depois pular o mais alto que pudesse</i>	<i>Pula</i>

Comandos de Atalho

Quando escrevemos o comando completo, sem atalhos, aqui escrevemos os mesmos comando que citei acima, porem de forma completa:

<i>Agachar, pegar impulso e esticar a pernas pulando</i>
<i>Apenas agachar</i>
<i>Mover uma perna a frente, apoiar no chão, inclinar o corpo a frente, depois mover a outra perna a frente mover o corpo a frente novamente</i>

Parar o movimento

Aqui vemos os comandos de forma completa, porem se para cada vez que defino um movimento, eu teria que escrever os comandos completamente, para cada vez que for repetir o comando.
Se eu desejo Pular, Andar e Pular, o comando ficaria:

Atalho	Comando
Pular	Agachar, pegar impulso e esticar a penas pulando
Andar	Mover uma perna a frente, apoiar no chão, inclinar o corpo a frente, depois mover a outra perna a frente mover o corpo a frente novamente
Pular	Agachar, pegar impulso e esticar a penas pulando

Aqui vale contar que não falei o TEMPO, pois de acordo com cada idéia de Comando iremos definir o Tempo de acordo com o projeto.

Agora que compreendemos a Lógica, iremos fazer uma analogia com a Programação do Arduino.

Vamos identificar no Arduino cada item desses que acabei de explicar:

Apresentação
Complemento
Comandos

_1Motor

```
int Mdir1 = 8;

void setup() {
    pinMode(Mdir1, OUTPUT);
}

void loop() {
    digitalWrite(Mdir1, HIGH);    //Comentario, observacoes
    delay(1000);
    digitalWrite(Mdir1, LOW);
    delay(1000);
}
```

(esse exemplo acima seria equivalente para fazer o LED piscar, bastava mudar o numero 8 para 13, que no caso é o pino que na outra imagem acima tínhamos ligado o LED)

Agora irei pegar o Código escrito acima e separar ele de acordo com o que expliquei.

Apresentação:

```
Int Mdir1 = 8;
```

Na linha acima fazemos a apresentação do que vai ser controlado, demos NOME, e onde ele esta ligado. Com isso encerramos as apresentações.

Complemento:

```
void setup() {
pinMode(Mdir1, OUTPUT);
}
```

Comando:

```
void loop() {
digitalWrite(Mdir1, HIGH); //Comentario, observacoes
delay(1000);
digitalWrite(Mdir1, LOW);
delay(1000);
}
```

No caso do exemplo de COMANDO, aqui apenas dou o comando em 1 dos pinos do Motor, hora esse Pino esta em HIGH (5v) e no momento seguinte esta em LOW (0V).

Para que possamos usar nossa Ponte H e controlar 2 motores, o que precisamos fazer é apresentar todos os Pinos, dar todos os complementos e depois escrever o Comando Atalho. Depois de criado o comando atalho, podemos chamar ele no “Comando de Repetição”.

Abaixo vou reescrever as 2 primeiras partes:

```
int Mdir1 = 8;
int Mdir2 = 9;
int Mesq1 = 10;
int Mesq2 = 11;
```

```
void setup() {
pinMode(Mdir1, OUTPUT);
pinMode(Mdir2, OUTPUT);
pinMode(Mesq1, OUTPUT);
pinMode(Mesq2, OUTPUT);
}
```

Acima acabei de listar os 2 conjuntos, Apresentação e Complemento. Apresentei os 4 pinos respectivos ao controle da Ponte H, cada 2 pinos respectivos aos comando do Motor, e depois fiz o complemento.

Com isso temos o mínimo necessário para dar os “Comandos”.

```
void loop() {
digitalWrite(Mdir1, HIGH); //Comentario, observacoes
delay(1000);
```

```

digitalWrite(Mdir1, LOW);
delay(1000);
}

```

Agora iremos compreender como criar um Atalho para realizar a mesma coisa escrita acima:

```

void Teste01() {
    digitalWrite(Mdir1, HIGH); //Comentario, observacoes
    delay(1000);
    digitalWrite(Mdir1, LOW);
    delay(1000);
}

```

Aqui criei um atalho, chamado “Teste01” ou seja se eu fosse querer escrever o “Comando de Repetição” (`void loop`) usando “Comando de Atalho” ou não, olhe a diferença que ficaria.

Comando sem atalho	Comando com Atalho
<pre> void loop(){ digitalWrite(Mdir1, HIGH); //Comentario, observacoes delay(1000); digitalWrite(Mdir1, LOW); delay(1000); } </pre>	<pre> void loop(){ Teste01(); } </pre>

Nesse exemplo podemos ver claramente a diferença dos Comandos, porem Observamos que só podemos declarar um Comando no Loop, se ele existir declarado, para declarar um “Comando Atalho” basta fazermos isso:

```

void loop(){
    Teste01();
}

void Teste01 () {
    digitalWrite(Mdir1, HIGH); //Comentario, observacoes
    delay(1000);
    digitalWrite(Mdir1, LOW);
    delay(1000);
}

```

Em si nos nossos Robôs iremos usar os atalhos, Frente, Direita, Esquerda, Reh e Parado.

Abaixo irei listar estes “Comandos Atalhos” (Funções)

```
void Reh()
{
    digitalWrite (MdirFWD, 255);
    digitalWrite (MdirREV, LOW);

    digitalWrite (MesqFWD, LOW);
    digitalWrite (MesqREV, 255);
}

void Frente()
{
    digitalWrite (MdirFWD, LOW);
    digitalWrite (MdirREV, 255);

    digitalWrite (MesqFWD, 255);
    digitalWrite (MesqREV, LOW);
}

void Esquerda()
{
    digitalWrite (MdirFWD, LOW);
    digitalWrite (MdirREV, 255);

    digitalWrite (MesqFWD, LOW);
    digitalWrite (MesqREV, 255);
}

void Direita()
{
    digitalWrite (MdirFWD, 255);
    digitalWrite (MdirREV, LOW);

    digitalWrite (MesqFWD, 255);
    digitalWrite (MesqREV, LOW);
}

void Parado()
{
    digitalWrite (MdirFWD, LOW);
    digitalWrite (MdirREV, LOW);

    digitalWrite (MesqFWD, LOW);
    digitalWrite (MesqREV, LOW);
}
```

Apos ter declarado todos os “Comandos Atalhos”, posso incluir qualquer dos atalhos no “void loop”.

Agora irei lista os comandos necessários para por o nosso Robô em movimento.

```
int MdirFWD = 8;
int MdirREV = 9;

int MesqFWD = 10;
int MesqREV = 11;

void setup(){
    Parado();

    {
        pinMode(MdirFWD, OUTPUT);
        pinMode(MdirREV, OUTPUT);

        pinMode(MesqFWD, OUTPUT);
        pinMode(MesqREV, OUTPUT);
    }

    void loop(){

        Reh();
        delay(500);

        Parado();
        delay(2000);

        Frente();
        delay(5000);

        Parado();
        delay(5000);

        Direita();
        delay(2000);

    }
}

void Reh()
{
    digitalWrite (MdirFWD, HIGH);
    digitalWrite(MdirREV, LOW);

    digitalWrite (MesqFWD, LOW);
    digitalWrite (MesqREV, HIGH);
}

void Frente()
{
    digitalWrite (MdirFWD, LOW);
```

```

digitalWrite (MdirREV, HIGH);
digitalWrite (MesqFWD, HIGH);
digitalWrite (MesqREV, LOW);
}

void Esquerda()
{
digitalWrite (MdirFWD, LOW);
digitalWrite (MdirREV, HIGH);

digitalWrite (MesqFWD, LOW);
digitalWrite (MesqREV, HIGH);
}

void Direita()
{
digitalWrite (MdirFWD, HIGH);
digitalWrite (MdirREV, LOW);

digitalWrite (MesqFWD, HIGH);
digitalWrite (MesqREV, LOW);
}

void Parado()
{
digitalWrite (MdirFWD, LOW);
digitalWrite (MdirREV, LOW);

digitalWrite (MesqFWD, LOW);
digitalWrite (MesqREV, LOW);
}

```

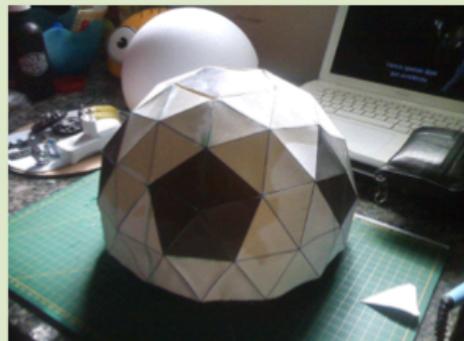
Acabamos de ver o código completo que iremos usar no Robô. Com isso poderemos ter o Robô andando. Importante lembrar que quando definimos o comando frente para o Robô e ele não andar para frente, gira para um dos lados, esse motor que girou para traz você deve trocar o fio de posição, assim como mencionei no texto acima.

Ao trocar o fio de posição no conector o comando Frente iora mandar o robô para frente.



Agora com isso já podemos pensar na personalização do nosso Robô. E uma ótima diversão.

Exemplos





outubro/novembro 2015 | VANILLA | 13

Na foto acima diversos componentes que podem ser aprimorados nos Robôs. (foto entrevista que saiu na revista Vanilla)

Facebook:

<https://www.facebook.com/casemonstro>