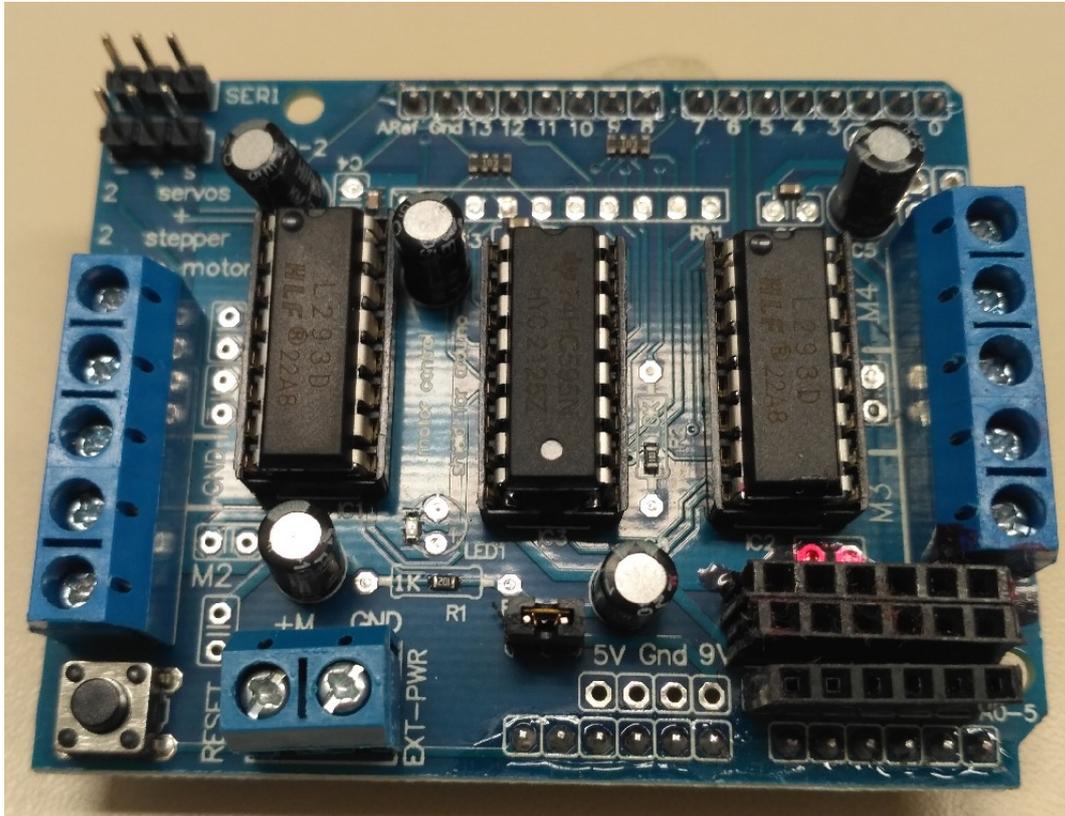


Controle de motores usando o *shield* LD293

Do Inglês, *shield* quer dizer escudo; no ambiente do Arduino chamamos de *shield* aqueles módulos que podem ser encaixados sobre a placa do Arduino. Os mais comuns são aqueles voltados ao Arduino UNO. O que utilizaremos é composto de dois controladores LD293 e um registrador de deslocamento 74HC795. Ele é capaz de controlar 4 motores DC ou 2 motores de passo. E também 2 servo motores. Os circuitos integrados são instalados em soquetes, de forma que podem ser retirados para experiências e circuitos individuais.



O *Motor Shield* utiliza o circuito integrado ponte H L293D e o circuito integrado 74HC595N para controle de comunicação. As 6 entradas analógicas (A0 a A5) estão disponíveis para uso na placa (embora nem sempre com conectores – na foto do meu *shield* soldei conectores para facilitar, incluindo as ligações GND e +5V que são ao lado). Ele utiliza os seguintes pinos do Arduino:

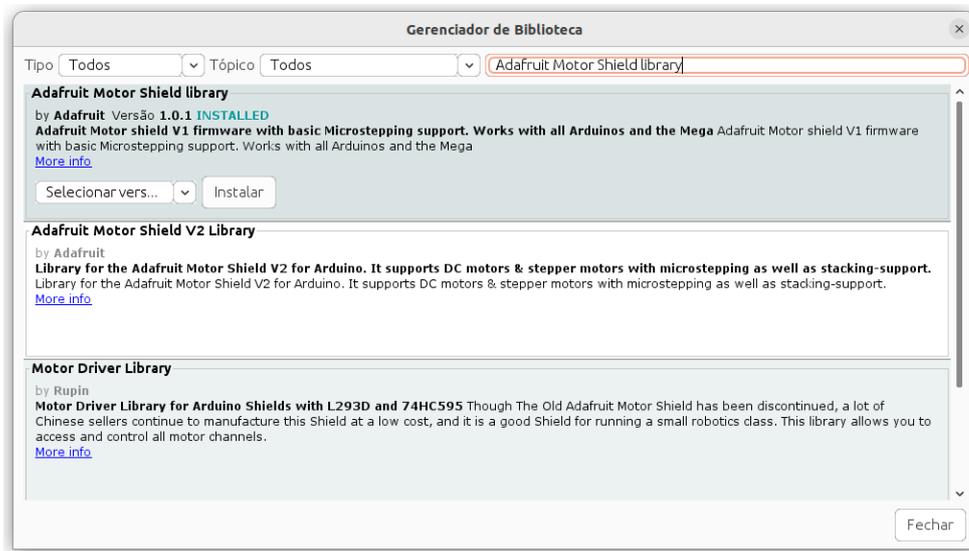
- Motor 1 (ou motor de passo 1 conexão 1): pino digital 11 do Arduino
- Motor 2 (ou motor de passo 1 conexão 2): pino digital 3 do Arduino
- Motor 3 (ou motor de passo 2 conexão 1): pino digital 5 do Arduino
- Motor 4 (ou motor de passo 2 conexão 2): pino digital 6 do Arduino
- Servo Motor 1: pino digital 9 do Arduino
- Servo Motor 2: pino digital 10 do Arduino
- Os pinos digitais 2 e 13 não são utilizados pelo *shield*, então você pode usá-los em seus protótipos se precisar.
- Os pinos digitais 4, 7, 8 e 12 são utilizados para controlar os motores por meio do 74HC795.

Veja a referência do fabricante (fornecedores chineses podem variar a placa, mas seguem as mesmas ligações físicas): <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-motor-shield.pdf>

Para facilitar a programação, iremos utilizar uma biblioteca pronta, da Adafruit. Para instalar a biblioteca, no ambiente do Arduino clique em *Sketch/ Incluir biblioteca / Gerenciar Bibliotecas* (vamos ‘buscá-la’ na internet; se você já tiver o arquivo pode clicar em ‘Adicionar biblioteca .zip’):



Na tela de gerenciamento de bibliotecas que se abrirá (pode demorar alguns instantes), vamos instalar a biblioteca ‘Adafruit Motor Shield library’:



Após instalada a biblioteca, podemos colocar o código e testar (se você digitar o código sem ter a biblioteca instalada haverá mensagens de erro, pois os comandos não serão conhecidos).

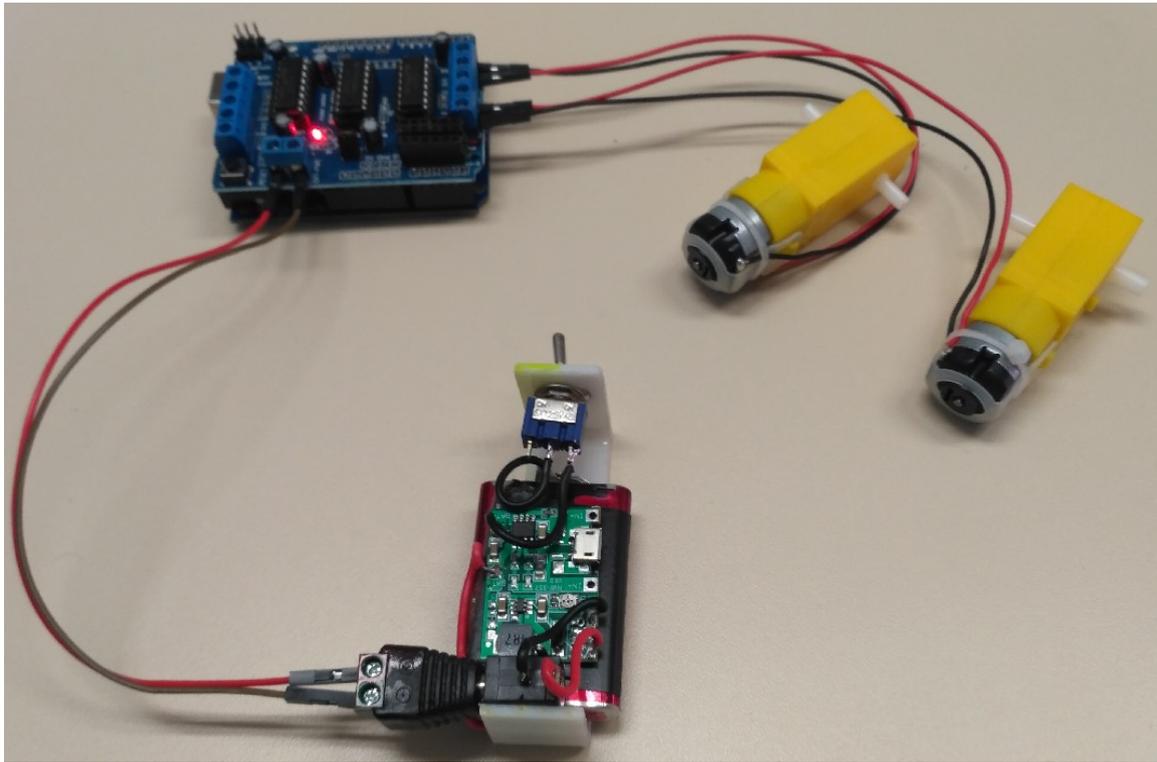
motoresComShieldLD293

```
1 #include <AFMotor.h> //esta é a biblioteca 'Adafruit Motor Shield library' que instalamos
2
3 AF_DCMotor mEsq(3); //AF_DCMotor é o objeto; mEsq o nome da variável; '3' foi a porta escolhida
4 AF_DCMotor mDir(4); //motor 'mDir' ligado na porta 4
5
6 void setup()
7 {
8   //velocidade inicial
9   mEsq.setSpeed(150);
10  mDir.setSpeed(150);
11
12  //parar os motores
13  mEsq.run(RELEASE); //'run' controla o motor e o parâmetro 'RELEASE' desliga o motor
14  mDir.run(RELEASE);
15
16  //note que não usamos o 'pinMode', pois a declaração do objeto 'AF_DCMotor' já faz isso
17 }
18
19 void loop()
20 {
21   // Liga os motores - lembre-se que já dissemos no 'setup' a velocidade inicial, 150
22   mEsq.run(FORWARD); //motor no sentido 'para frente'
23   mDir.run(FORWARD); //o sentido do movimento depende da ligação do motor e de sua posição
24
25   delay(2000); //Aguarda 2 segundos
26
27   //parar os motores
28   mEsq.run(RELEASE); //'run' controla o motor e o parâmetro 'RELEASE' desliga o motor
29   mDir.run(RELEASE);
30   delay(100); //aguarda antes de inverter o sentido de rotação
31
32   mEsq.run(BACKWARD); //motor no sentido 'para trás'
33   mDir.run(BACKWARD); //o sentido do movimento depende da ligação do motor e de sua posição
34
35   delay(2000); // aguarda 2 segundos
36
37   //parar os motores
38   mEsq.run(RELEASE); //'run' controla o motor e o parâmetro 'RELEASE' desliga o motor
39   mDir.run(RELEASE);
40   delay(100); //aguarda antes de inverter o sentido de rotação
41 }
```

Explicação:

- na linha 1 foi incluída a biblioteca de controle;
- nas linhas 3 e 4 foram declaradas variáveis (escolhi os nomes Mesq e Mdir para referência a um motor esquerdo e um direito em um carrinho), e foi informado em qual porta cada um foi conectado (usei a M3 e a M4, mas poderia ser M1 ou M2 ou qualquer 'mistura' de duas portas);
- entre as linhas 6 e 17 foi declarada a função **setup** e dentro dela há duas informações para cada um dos dois motores: a velocidade inicial (usei 150, por meio do método **setSpeed** - pode ser qualquer número entre 0 e 255); e, por meio do método **'run'**, foi informado o argumento **'RELEASE'**, a qual faz os motores desligarem;
- entre as linhas 19 e 41 temos a função **loop**; dentro dela foram utilizados os métodos **run** com os argumentos **'FORWARD'** e **'BACKWARD'**, além do **'RELEASE'**, intercalados com comandos **delay**. Isto fará com que os motores girem em um sentido durante 2 segundos (2000 ms), desliguem e permaneçam assim por 0,1s (100ms) e depois liguem girando no sentido oposto por mais 2 segundos (se você testar e os motores não girarem ambos no mesmo sentido, inverta as ligações de um dos motores).

Ligação dos motores:



(a alimentação foi fornecida por uma bateria de 3,7V ligada em um módulo carregador / *step up* integrado identificado pelo código J5019 – veja como em <https://tiaplicada.ufpr.br/wp-content/uploads/2024/08/usodomoduloj5019stepupcarregador.pdf>).

Este conteúdo pode ser copiado, editado e distribuído livremente. PINTO, José Simão de Paula. **Controle de motores usando o *shield* LD293**. Curitiba : Edição própria, 2024.