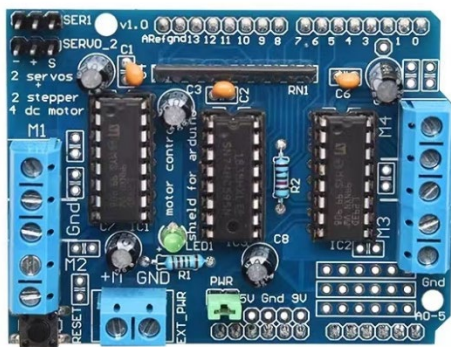


Manual do escudo de driver de motor L293D

Os motores CC são um componente essencial de muitos sistemas robóticos e de automação. São simples de controlar e podem ser alimentados por uma variedade de fontes de tensão. O shield de motor L293D pode ser usado para todos os tipos de projectos de robótica. Neste tutorial de shield de motor arduino, você aprenderá como usar o L293D Motor Shield para Arduino para controlar a velocidade e a direção de um motor DC. O tutorial ensina como instalar a biblioteca AFmotor e como usá-la para acionar o motor DC. O código do programa e a demonstração em vídeo também são fornecidos.

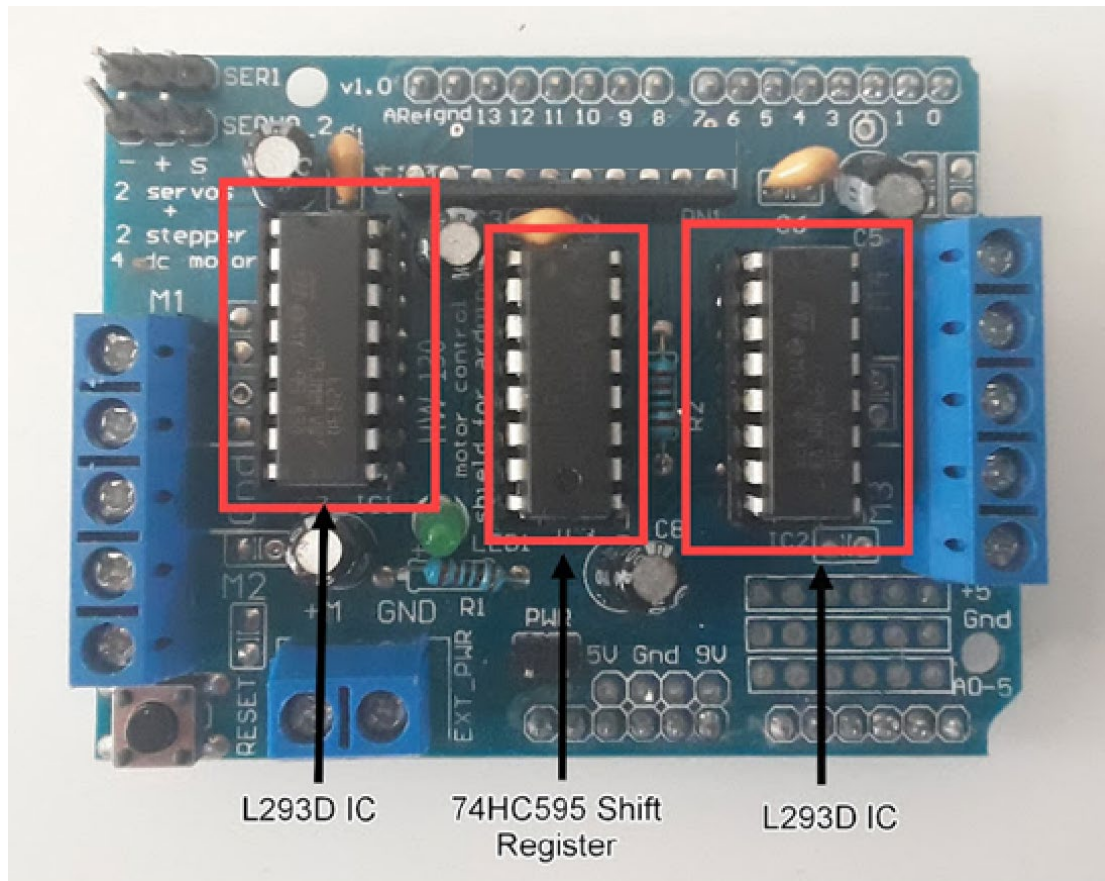
A proteção do motor L293D é mostrada abaixo.



A proteção do motor L 293D é um dispositivo popular utilizado para controlar motores DC. Trata-se de uma placa compacta e fácil de utilizar que pode controlar até dois motores DC. A proteção está equipada com um circuito integrado L 293D, capaz de acionar motores com uma tensão que varia entre 4,5 V e 36 V e com uma corrente até 600 mA. A proteção do motor L 293D proporciona uma forma simples e conveniente de controlar motores DC, uma vez que não requer quaisquer componentes ou cabos adicionais.

Circuitos integrados de driver de motor

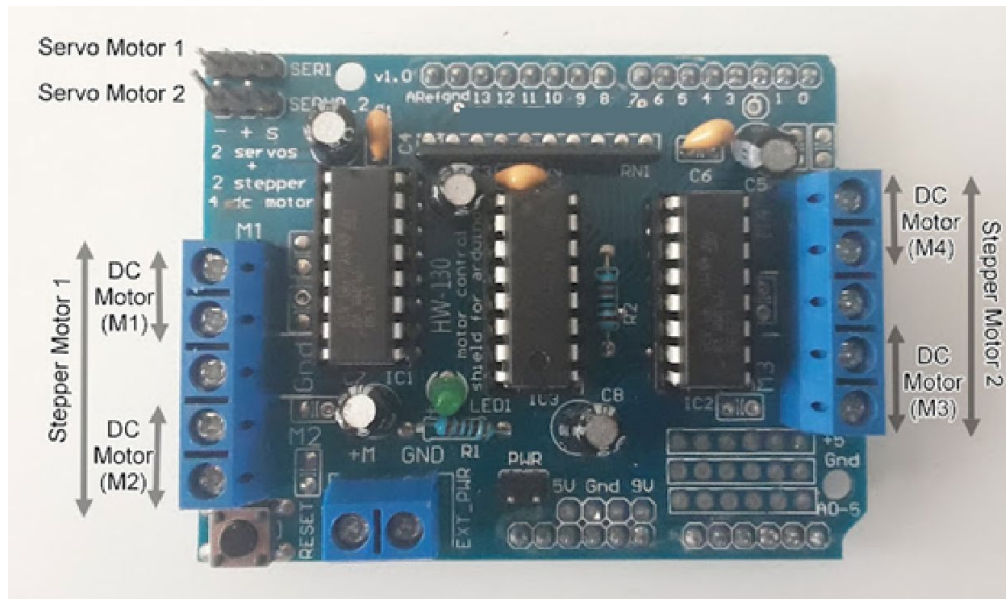
O módulo de acionamento do motor é constituído principalmente por dois circuitos integrados de acionamento do motor L293D e um circuito integrado de registo de deslocação 74HC595, juntamente com resistências, condensadores e portas de ligação de apoio.



Porta de ligação do motor

O controlador de motor L293D pode acionar:

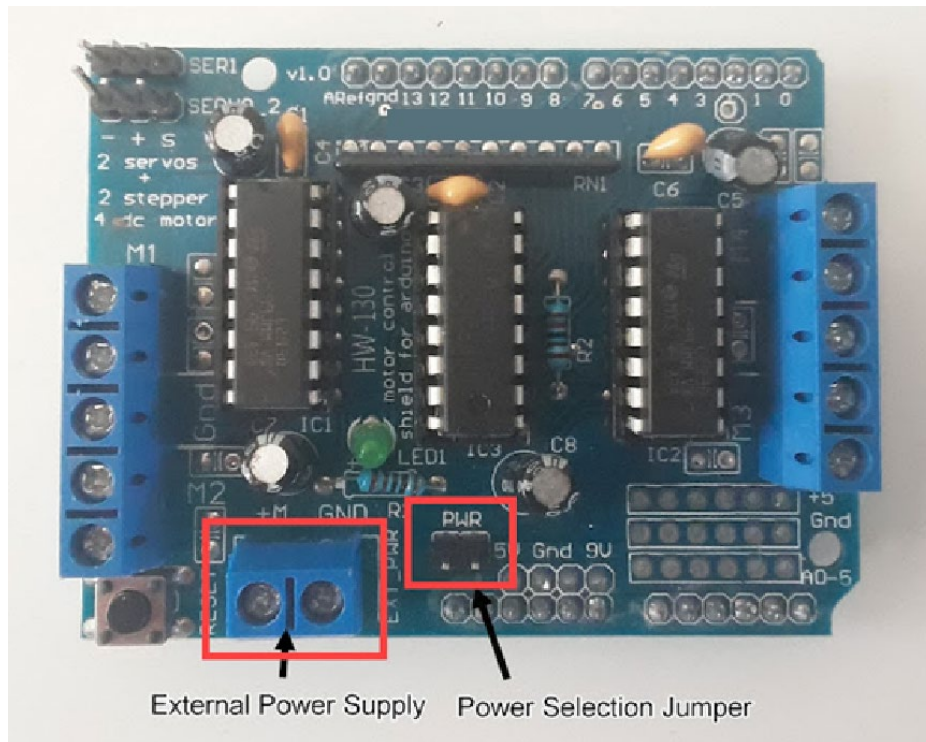
- Quatro motores DC que podem ser controlados bi-bidireccionalmente e com velocidade variável utilizando PWM
- Dois motores de passo
- Dois servo-motores



Potência

O shield de driver de motor pode ser utilizado para acionar pequenos motores que podem funcionar com um requisito de corrente na ordem dos 600mA por motor e corrente de pico de 1,2A e requisito de tensão na ordem dos 4,5V a 25V.

No shield existe um jumper de seleção da fonte de alimentação. Se pretender alimentar o Arduino e o(s) motor(es) com uma fonte de alimentação externa, coloque o jumper. Se pretender uma alimentação separada para o Arduino e o(s) motor(es), seja porque pretende alimentar o Arduino através de USB ou de uma tomada DC externa, retire o jumper de seleção de alimentação. É preferível utilizar uma fonte de alimentação diferente para alimentar o Arduino e os motores e, nesse caso, remover o jumper de seleção de alimentação.

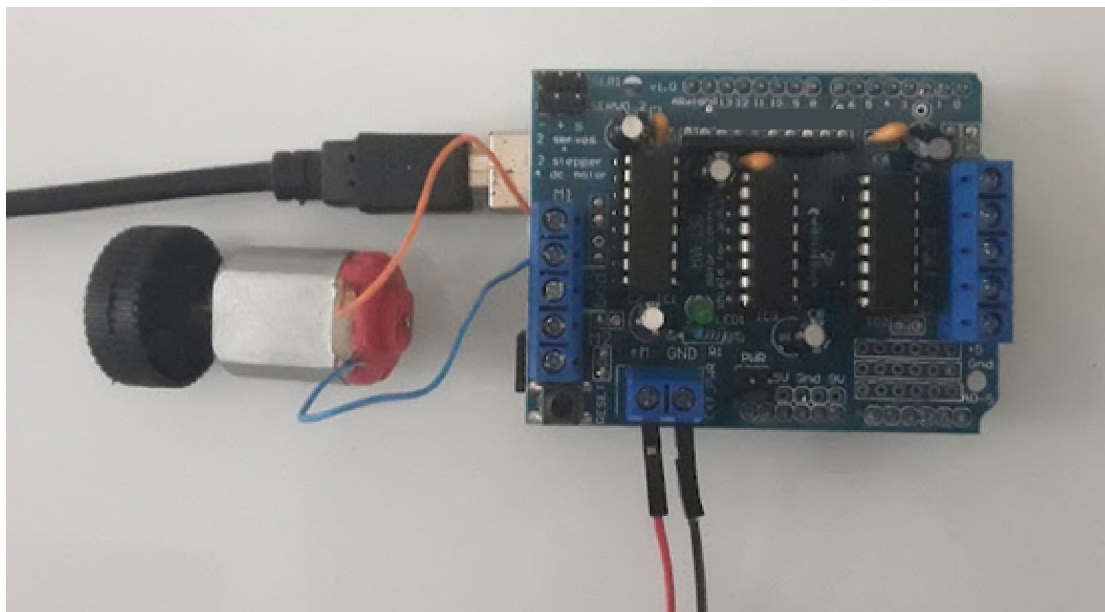


Interface do motor DC com o Motor Shield

Para controlar o motor DC utilizando um shield de motor L 293D e um Arduino, precisamos de ligar o shield à placa Arduino. A shield tem cabeçalhos que são compatíveis com os cabeçalhos de uma placa Arduino, pelo que a ligação entre os dois é simples. Depois de ligar o shield ao Arduino, podemos ligar o motor DC ao shield. A shield tem duas ligações para o motor, cada uma com o seu próprio conjunto de linhas de alimentação e controlo.

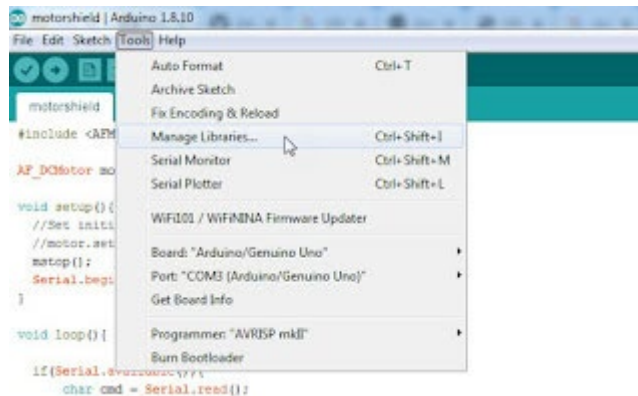
Assim que a shield estiver ligada ao Arduino e ao motor DC, podemos escrever um sketch (programa) para controlar o motor. O sketch irá utilizar os pinos de E/S digitais do Arduino para controlar o motor. Para controlar a velocidade do motor, podemos utilizar a modulação por largura de impulso (PWM) num dos pinos de E/S digitais. A PWM permite-nos controlar a tensão média aplicada ao motor, o que, por sua vez, controla a sua velocidade.

Neste tutorial, vamos ligar um motor DC à porta de ligação do motor M1. Vamos utilizar uma fonte de alimentação diferente para acionar o Arduino UNO e o motor, pelo que vamos remover o jumper de seleção de alimentação. Uma fonte de alimentação de 9V é ligada ao terminal de alimentação externa.

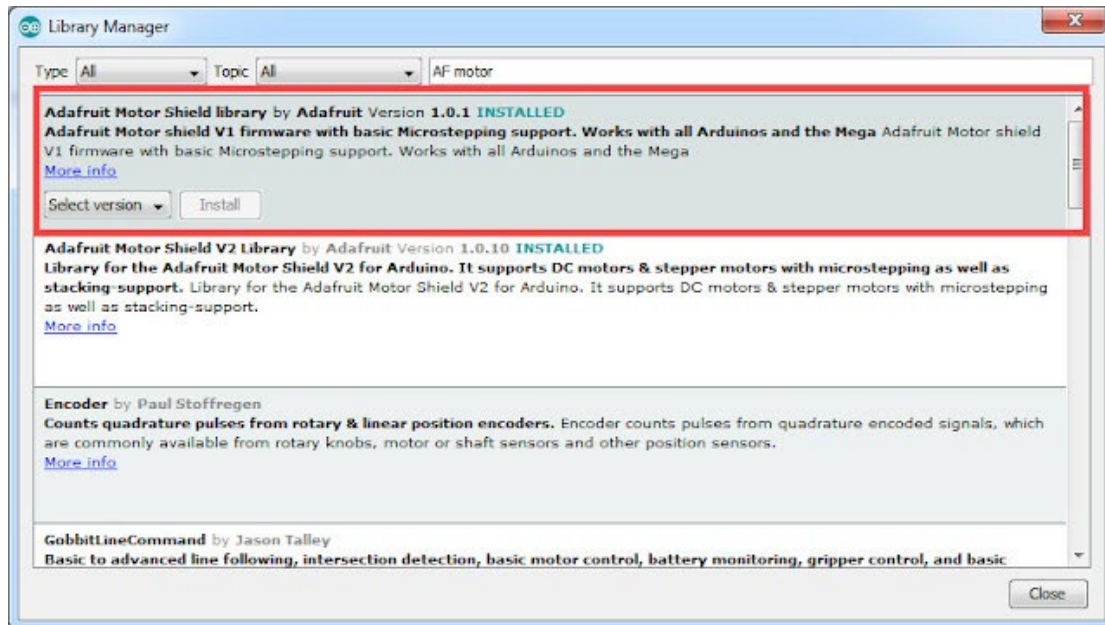


Programar o Arduino com o Motor Shield

Para programar o Arduino para fazer funcionar um motor DC utilizando o Motor Shield, vamos utilizar a biblioteca Adafruit Motor Shield. Para instalar a biblioteca, no IDE do Arduino, vá a Tools > Manager Libraries...



Depois, na janela Library Manager, procure por AF motor, encontre a Adafruit Motor Shield Library e instale-a. Como mostrado abaixo, porque a biblioteca no nosso caso já estava instalada, o botão de instalação está desativado. Mas para si, se ainda não tiver instalado a biblioteca, o botão de instalação estará disponível. Clique nele para instalar a biblioteca e reiniciar o IDE.



A utilização da biblioteca é bastante simples. Abra um sketch e escreva o seguinte no topo para utilizar as funções da biblioteca.

```
#include <AFMotor.h>
```

Depois de incluir o ficheiro de cabeçalho, temos de criar um objeto com um nome alternativo. Para os motores de corrente contínua, temos de utilizar a classe `AF_DCMotor` para criar um objeto de motor de corrente contínua. Por exemplo, a linha seguinte cria um objeto de motor de corrente contínua chamado `motor`.

```
AF_DCMotor motor(1);
```

O objeto `motor` aceita dois parâmetros que são o número (1, 2, 3 ou 4) ou o canal do motor ao qual o motor será ligado. E o segundo parâmetro é a frequência. As definições de frequência para os canais 1 e 2 são:

```
MOTOR12_64KHZ
```

```
ou, MOTOR12_8KHZ
```

```
ou, MOTOR12_2KHZ
```

```
ou, MOTOR12_1KHZ
```

E as frequências para os canais 3 & 4 são:

MOTOR34_64KHZ

ou, MOTOR34_8KHZ

ou, MOTOR34_1KHZ

É opcional especificar o parâmetro de frequência ao criar o objeto do motor DC. Se não especificarmos a frequência, a frequência predefinida é 1KHz.

Uma vez criado o objeto DC, podemos utilizar os seus vários métodos. Estes métodos são explicados brevemente de seguida.

1. motor.setSpeed(parameter);

Este método setSpeed() define a velocidade do motor de corrente contínua, que recebe um parâmetro que varia entre 0 (velocidade mais baixa) e um máximo de 255 (velocidade mais alta).

eg. motor.setSpeed(127);

2. motor.run(parameter)

Este método run() define o sentido de rotação do motor. Pode receber os parâmetros - FORWARD, BACKWARD, RELEASEg. motor.run(FORWARD)

Código do programa

O programa seguinte irá mover o motor DC ligado ao canal 1 do escudo do motor para a frente, para trás, acelerar, desacelerar ou parar de acordo com o comando que enviarmos para o Arduino através da porta série. Se enviarmos o carácter "f", o motor funcionará para a frente, se enviarmos o carácter "b", o motor funcionará para trás, se enviarmos "a", o motor acelerará, se enviarmos "d", o motor desacelerará e se enviarmos "s", o motor deixará de girar.

```
#include <AFMotor.h>
```

```
AF_DCMotor motor(1);
```

```
void setup(){  
  
  mstop();  
  
  Serial.begin(9600);  
  
}
```

```
void loop(){  
  
  if(Serial.available()){  
  
    char cmd = Serial.read();  
  
    switch(cmd){  
  
      case 'f':  
  
        forward();  
  
        break;  
  
      case 'b':  
  
        backward();  
  
        break;  
  
      case 'a':  
  
        accelerate();  
  
        break;
```



```
    case 'd':  
        deaccelerate();  
        break;  
  
    case 's':  
        mstop();  
        break;  
  
    default:  
        break;  
    }  
}  
}
```

```
void forward(){  
    motor.setSpeed(200);  
    motor.run(FORWARD);  
}
```

```
void backward(){  
    motor.setSpeed(200);
```

```
motor.run(BACKWARD);
```

```
}
```

```
void accelerate(){
```

```
motor.run(FORWARD);
```

```
uint8_t i;
```

```
for (i=0; i<255; i++){
```

```
motor.setSpeed(i);
```

```
delay(10);
```

```
}
```

```
}
```

```
void deaccelerate(){
```

```
motor.run(FORWARD);
```

```
uint8_t i;
```

```
for(i=255; i!= 0; i--){
```

```
motor.setSpeed(i);
```

```
delay(10);
```

```
    }  
}  
  
void mstop(){  
    motor.run(RELEASE);  
}
```

Em conclusão, o controlo de um motor DC utilizando uma placa de motor L 293D e um Arduino é uma forma simples e conveniente de controlar motores numa variedade de aplicações. Com apenas alguns componentes simples e um pouco de programação, podemos criar sistemas de controlo sofisticados que utilizam motores DC. Quer esteja a construir um sistema robótico ou a automatizar um processo, a placa de motor L 293D e o Arduino são ferramentas poderosas que o podem ajudar a atingir os seus objectivos.