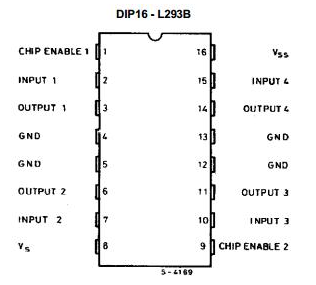
# Controlando Motores DC com o Arduino

A utilização de motores DC como atuadores em projetos mecatrônicos é muito comum. Estes motores podem ter o sentido de rotação controlado de acordo com o sentido da corrente aplicada em seus terminais. Além disso, utilizando modulação por largura de pulso (PWM) é possível controlar a velocidade de rotação do motor sem que se perca o torque do mesmo.

Um circuito muito útil para controlar o sentido da corrente que alimenta um motor DC é o Circuito Ponte H. Não é o objetivo deste artigo explicar o funcionamento deste circuito, de modo que basta saber que o mesmo é composto por transistores e diodos que, pela sua configuração, permite alterar o sentido da corrente aplicado em um motor DC.

O circuito integrado [L293B](http://pdf.datasheetcatalog.com/datasheet/SGSThomsonMicroelectronics/mXurruu.pdf) é um CI que possui em seu interior duas Pontes H, sendo o mesmo capaz de trabalhar com correntes de até 1A. Como neste artigo nosso objetivo é apresentar uma forma de se controlar motores DC utilizando o Arduino, o L293B nos atenderá perfeitamente, pois, além de sua simplicidade, ele permite fornecer ao motor uma corrente bastante acima daquela suportada pelo Arduino que é por volta de 40mA.

Para iniciar nosso estudo vamos começar analisando o CI L293B. A figura abaixo apresenta a pinagem do mesmo:



Este circuito contém internamente duas Pontes H que podem controlar o sentido de rotação (esquerda, direita) de dois motores DC. Os pinos 2, 3, 7 e 6 são respectivamente os pinos de entrada, saída, entrada e saída da primeira Ponte H. Os pinos 15, 14, 10 e 11 são, respectivamente, entrada, saída, entrada e saída da segunda Ponte H. Note que cada ponte possui duas entradas e duas saídas. De acordo com a entrada acionada, sua saída correspondente conectada ao motor ficará ativa, determinando o sentido de rotação do motor ligado à mesma.

Os pinos 1 e 9 são utilizados para ativar tanto a primeira quanto a segunda Ponte H. Quando o pino 1 estiver acionado, a primeira ponte ficará operante, por outro lado, se o pino 9 for acionado, a segunda ponte passa a funcionar. Se tanto o pino 1, quanto o 9 estiverem acionados, as duas pontes passam a funcionar ao mesmo tempo.

Os pinos 4, 5, 12 e 13 devem ser conectados ao terra. Os dois primeiros são correspondentes à primeira ponte e os dois últimos à segunda ponte. O pino 16 deve ser utilizado para alimentar o circuito e a tensão aplicada ao mesmo deve estar entre 4,5V e 36V. Além disso, o pino 8 é utilizado para conectar a alimentação dos motores e também dever ter entre 4,5V e 36V.

É importante ressaltar que deve-se evitar acionar ao mesmo tempo as duas entradas da mesma Ponte H, sob risco de danificar o CI. Deste modo, se o pino 1 estiver acionado, é preciso que o 7 esteja desativado. O mesmo ocorre com os pinos 15 e 10.

Uma dica interessante é que podemos controlar a velocidade de rotação dos motores alimentando os pinos 1 e 9 com PWM, de modo que variando a largura de pulso aplicado aos pinos varia-se a velocidade de rotação dos motores 1 e 2 respectivamente.

Já que conhecemos o CI L293B, vamos iniciar a montagem do nosso projeto com o Arduino.

OBJETIVO: Controlar o sentido de rotação de um motor DC conectado ao Arduino utilizando-se dois botões. O acionamento de cada botão determinará o sentido de rotação do motor. Desse modo, quando o botão 1 for acionado, o motor deverá girar para a esquerda e quando o botão 2 estiver acionado o sentido de rotação inverterá. Em nenhum momento deverá ser permitido o acionamento dos dois botões ao mesmo tempo. Código fonte para o Arduino.

int input1 = 4;

int enable1 = 2;

int input2 = 5;

int button1 = 6;

int button2 = 7;

void setup(){

pinMode(input1, OUTPUT);

pinMode(enable1, OUTPUT);

pinMode(input2, OUTPUT);

pinMode(button1, INPUT);

pinMode(button2, INPUT);

}

void rotateLeft(){

digitalWrite(enable1, HIGH);

digitalWrite(input2, LOW);

digitalWrite(input1, HIGH);

}

void rotateRight(){

digitalWrite(enable1, HIGH);

digitalWrite(input1, LOW);

digitalWrite(input2, HIGH);

}

void parar(){

digitalWrite(enable1, LOW);

digitalWrite(input1, LOW);

digitalWrite(input2, LOW);

}

void loop(){

while(digitalRead(button1) == HIGH){

rotateLeft();

}

while(digitalRead(button2) == HIGH){

rotateRight();

}

parar();

}

**Referência:**

1. http://labdeeletronica.com.br/motordc-e-arduino/