

Controlador de motor de passo com PIC 12C508

025

C. Tavernier

www.tavernier-c.com

Quando não estamos usando um motor de passo para obter um posicionamento preciso de um elemento de um robô, podemos usar este como um motor de tração, em vez dos tradicionais servos modificados. Nesse caso, não há necessidade de contar os passos que o motor dá, uma vez que o que queremos é que este gire de uma forma contínua numa direção ou em outra.

Existem várias soluções para controlar um motor, onde algumas são apresentadas nesta edição: usando um circuito integrado específico para controle de motores de passo, usando um ou mais saídas de uma porta de um microcontrolador previamente programado, ou construir um circuito de controle baseado em circuito integrados lógicos convencionais.

Contudo, estas soluções estão longe de ser perfeitas quando se pretende utilizar um motor de passo como motor de tração. Todos eles precisam gerar pulsos de uma forma contínua enquanto quisermos que o motor rode, mesmo que seja necessário implementar um oscilador programável, ou usar recursos do microcontrolador principal do robô.

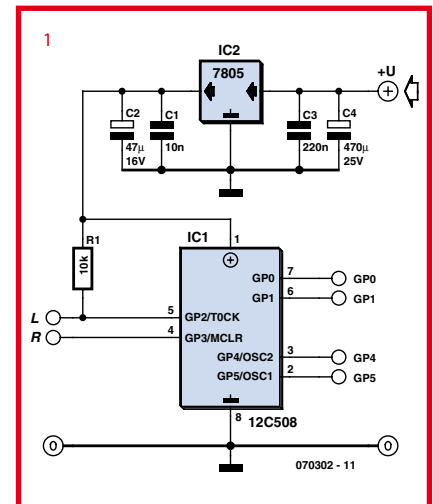
Decidimos então sugerir uma outra solução para controlar este motor de passo de modo que ele gire numa direção ou em outra, recorrendo a simples sinais lógicos. Como os motores de propulsão em robôs são usados habitualmente aos pares, vamos disponibilizar um controlador duplo, adaptando um circuito integrado bastante comum de modo a realizar uma outra tarefa diferente.

Uma vez que um motor de passo, quando usado para tração, não precisa ser preciso em termos do posicionamento, e também no que diz respeito à precisão dos passos, os modelos simples com um único pólo são adequados. Assim, o nosso circuito foi projetado para usar este tipo de motores. Isto permite controlarmos o motor através de duas entradas lógicas TTL ou CMOS. Quando estas duas entradas (L e R) estão no nível lógico alto (ou em aberto) o motor

fica parado, mas em modo travado, dado que estamos falando de um motor de passo. Quando a entrada L vai para o nível lógico zero, o motor roda numa direção, enquanto que se a entrada R for para o nível lógico zero ele roda na outra direção. Se ambas as entradas estiverem no nível lógico zero, a entrada R tem prioridade, de forma que o motor roda nessa direção. A velocidade de rotação do motor é fixa, mas, como estamos fornecendo a listagem com o código fonte do software usado nesta aplicação, é muito simples o leitor alterar isto se não estiver ao seu agrado, ou mesmo para incluir a possibilidade de ajuste de velocidade externo, caso seja necessário. O circuito da parte inteligente do nosso controlador é apresentado na **Figura 1**, e como pode ver este usa um microcontrolador PIC12C508 da Microchip. Neste caso, configurado para usar o gerador de relógio interno e o modo de reset do circuito, não precisando de qualquer componente externo para essas funções, de forma que todas as linhas das portas estão disponíveis. As portas paralelas GP2 e GP3 são usadas como entradas, e como GP2 não tem resistências pull-up internas, isto é implementado por R1. As portas paralelas GP0, GP1, GP4 e GP5 são usadas como saídas para gerar os pulsos de controle para o motor. Estes podem ser amplificados por dois tipos de estágios de potência, dependendo do tipo e do número de motores a controlar; daqui a alguns instantes vamos dar uma olhada nesses circuitos.

O 12C508 precisa ser alimentado com uma tensão de 5 V, derivada da alimentação do motor através de um regulador de três pinos (IC2). Se o controlador for apenas para um motor, ou se o motor a controlar precisar mais de 500 mA por enrolamento, pode ser usado o estágio de potência apresentado na **Figura 2**. Este utiliza transistores bipolares convencionais que, dadas as suas características, são capazes de comutar correntes até 3 A. Os diodos D1 a D8 limitam os eventuais picos de corrente gerados pela comutação da alimentação dos enrolamentos do motor,

protegendo desse modo os transistores. Contudo, se o motor usado precisar menos de 500 mA, e mais importante, se precisarmos controlar dois motores deste tipo, existe uma solução elegante e engenhosa, apresentada na **Figura 3**. Esta usa um circuito integrado excitador ULN2803, usado geralmente para excitar relés, e que inclui oito transistores Darlington de média potência mais os respectivos diodos de proteção. Assim, este circuito integrado é capaz de controlar qualquer tipo de motor passo-a-passo de pólo simples, desde que a tensão não exceda os 50 V e a corrente por enrolamento seja inferior a 500 mA. Adicionalmente, como o ULN2803 possui oito estágios idênticos, pode ser usado com dois controladores como o que se pode ver na Figura 1, e, neste caso, controlar dois motores de propulsão para robô: um na esquerda e outro na direita, designados por MA e MB na figura. Construir uma ou outra destas versões é muito simples. O PIC12C508 precisa ser programado com o arquivo objeto do programa, que você pode encontrar no site da Elektor, ou então no site do autor (www.tavernier-c.com). Se você construir o amplificador de potência com transistor, tenha em mente que T1-T4 não precisam de dissipador de calor desde que a corrente consumida pelo motor não



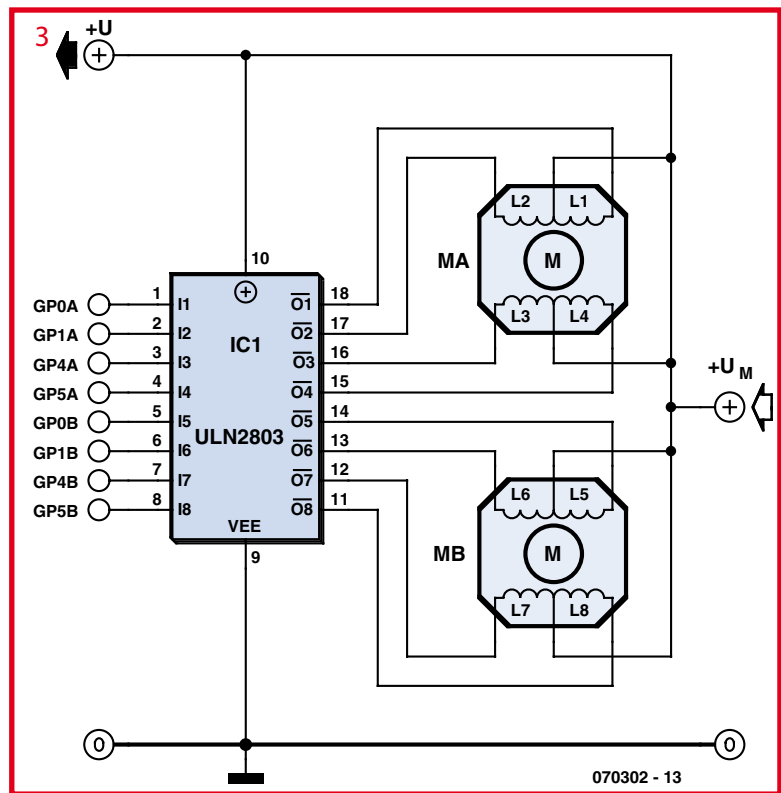
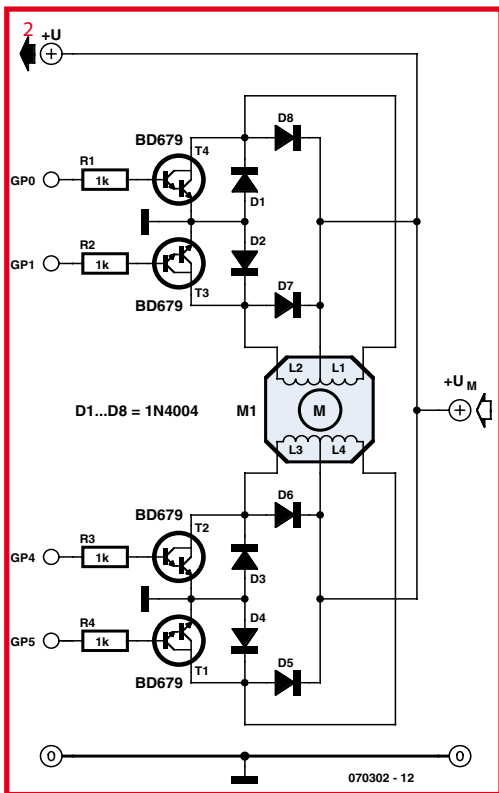


Tabela 1. Programação da duração do passo, através da alteração da constante usada no programa.

Constante binária de passo	Duração
10010010	1 ms
10010011	2 ms
10010100	4 ms
10010101	8 ms
10010110	16 ms
10010111	32 ms

exceda o 1 A. Caso contrário, parafuse-os a uma pequena chapa de alumínio com alguns cm² de área. Para simplificar

a construção mecânica, esta pode ser comum aos quatro transistores, mas neste caso vai precisar usar uma mica isoladora entre o alumínio e os transistores.

Se você construir a versão baseada no ULN2803, não existem precauções especiais, além de não exceder a corrente máxima permitida de 500 mA.

Como estamos fornecendo a listagem completa do software programado no PIC12C508, o leitor pode efetuar as modificações que achar necessárias. Se você não está familiarizado com o assembler do PIC, aqui ficam os detalhes que precisa para a alteração mais importante que possa querer fazer: alterar a velocidade dos pulsos de controle do motor, e assim, alterar a

sua velocidade de rotação. A palavra de controle pode ser encontrada na **Tabela 1**. Para fazer isto, você apenas precisa alterar a constante binária na linha:

```
MOVLW B'10010101'
```

Mesmo por cima da linha que contém OPTION na listagem do código fonte. Com o valor original, a duração de um passo é de 8 ms, mas a tabela acima indica que constante deve usar de acordo com a duração de passo que pretende.

(070302-1)

Artigo original: PIC12C508 Stepper Motor Controller - July/August

2007