

**INSTITUTO POLITÉCNICO DE TOMAR**

Escola Superior de Tecnologia de Tomar

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

Engenharia Electrotécnica e de Computadores / Engenharia Informática

**MICROPROCESSADORES E APLICAÇÕES**

2005/2006

Ficha N.º2

# CONTROLO DE UM MOTOR DE PASSO

## 1 Objectivos

O objectivo deste trabalho é desenvolver uma aplicação embebida, utilizando um microcontrolador da família PIC18, que implemente um controlador, tipo meio-passo, para um motor de passo.

## 2 Introdução

A popularidade dos motores de passo deve-se à adaptação destes dispositivos à lógica digital. Estes são utilizados em inúmeras aplicações associadas a periféricos de computadores, tais como: impressoras, unidades de disco, plotters, etc. Para além da informática, estes motores são também utilizados na robótica em sistemas de movimentação de braços mecânicos, etc.

Em termos gerais um motor de passo consiste num motor DC de magnetes permanentes ou de relutância variável que apresenta as seguintes características de desempenho:

- Rotação em ambas as direcções;
- Variações incrementais da posição angular;
- Repetição de movimentos de forma exacta;
- Binário de sustentação à velocidade zero; e
- Possibilidade de implementar controlo digital.

Um motor de passo pode mover-se em incrementos angulares bastante exactos, conhecidos como passos, em resposta a pulsos digitais aplicados a um driver de potência, que neste caso será o CI L293, a partir de um controlador digital, que neste caso será o  $\mu$ controlador da MICROCHIP PIC18F258.

*O número de pulsos e a cadência com que estes pulsos são aplicados controlam a posição e a velocidade do motor, respectivamente.*

Geralmente os motores de passo podem ser fabricados com 12, 24, 72, 144 e 200 passos por revolução, que resultam em incrementos de 30, 15, 5, 2.5, 2 e 1.8 graus respectivamente. O motor de passo utilizado neste trabalho é do tipo unipolar, ou seja requer apenas uma fonte de alimentação, de tensão contínua.

Os motores de passo unipolares são facilmente reconhecidos pela derivação ao centro em cada um dos enrolamentos. O número de fases é duas vezes o número de bobinas, uma vez que cada bobina se encontra dividida em duas. Na Figura 1 encontra-se a representação de um motor de passo unipolar de 4 fases. A derivação central dos enrolamentos está ligada à terra e os extremos de cada enrolamento são ligados alternadamente ao terminal positivo da fonte de alimentação para assim inverter a direcção do campo gerado por cada um dos enrolamentos.

A Figura 1 mostra o esquema básico dos enrolamentos de um motor de passo unipolar.

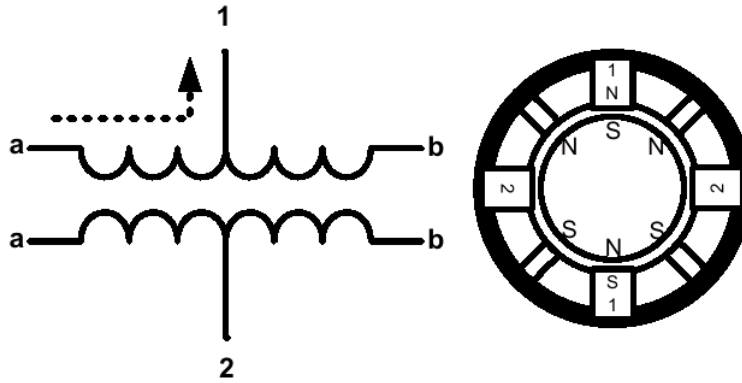


Figura 1: Esquema de enrolamentos de um motor de passo unipolar

Na Figura 1 podemos ver o corte transversal de um motor com um passo de 30 graus. O enrolamento 1 encontra-se distribuído entre o pólo superior e pólo inferior do estator do motor, enquanto que o enrolamento 2 encontra-se distribuído entre o pólo esquerdo e o pólo direito do estator. O rotor é um magnete permanente com seis pólos (3 pólos sul e 3 pólos norte), dispostos ao longo da circunferência do rotor.

Para uma resolução angular maior, o rotor deverá conter proporcionalmente mais pólos. Tal como apresentado na Figura 1, a corrente a fluir do terminal a para a derivação central do enrolamento 1 faz com que o pólo superior do estator seja um pólo norte enquanto que o pólo inferior seja um pólo sul. Esta situação provoca um deslocamento do rotor para a posição indicada na Figura 1. Se for removida a alimentação do enrolamento 1 e for alimentado o enrolamento 2, o rotor irá deslocar-se 30°, ou seja, um passo.

Para obter uma rotação contínua do motor, deverão ser alimentados alternadamente os enrolamentos do motor. Assumindo uma lógica positiva, em que o valor lógico "1" significa passar a corrente num dos enrolamentos, a sequência apresentada na Tabela 1, fará rodar o motor meio-passo de cada vez.

Index	1a	1b	2a	2b
1	1	0	0	0
2	1	0	1	0
3	0	0	1	0
4	0	1	1	0
5	0	1	0	0
6	0	1	0	1
7	0	0	0	1
8	1	0	0	1

Tabela 1: Sequência de meio-passo para motores unipolares

Esta sequência deverá ser fornecida aos enrolamentos do motor através do nibble mais significativo do porto B do  $\mu$ controlador .

### 3 Operação com o PIC18F258

#### 3.1 Algumas considerações sobre o programa para o PIC18F258

Neste trabalho irá ser utilizado o  $\mu$ controlador da MICROCHIP *PIC18F258*.

Para colocar os valores correctos da sequência no nibble mais significativo do porto C, sugere-se que utilize

um *array* que armazene a sequência e uma máscara que conserve os valores do nibble menos significativo. Percorrendo os valores do *array* conseguir-se-á colocar de forma correcta os valores desejados à saída do porto.

A título de exemplo mostra-se se poderia usar os 7 bits menos significativos do porto B para comandar um display de 7 segmentos. As cadeias de bits usadas para gerar os números de 0 a 9 num display de sete segmentos de ânodo comum encontram-se armazenadas num array colocado na ROM do  $\mu$ controlador. Note-se que se as 7 saídas menos significativas do porto B do  $\mu$ controlador estão ligadas às entradas a, b, c, d, e, f, g (A entrada g está ligada a PB.6). A saída mais significativa do porto B, PB.7 está a ser usada por outro dispositivo e não deve ser alterada por esta rotina.

De seguida apresenta-se o código de uma rotina que recebe como parametro o número a aparecer no display de 7 segmentos e que activa as saídas necessárias do porto B para o conseguir.

---

### Código: 3.1

```
// Variaveis Globais

rom far const char SeteSegmts [] = {0b10011111, 0b00100101, 0b00001101,
                                     0b10011001, 0b01001001, 0b11000001,
                                     0b00011111, 0b00000001, 0b00011001};

void mostraValor(char valor){
    char var_aux;                               //variavel auxiliar
    var_aux = LATB;                             //Guarda o Valor actual da Saida do Porto B
    var_aux = 0b10000000 & var_aux;            //Limpa os 7 bits menos significativos (BmS)
    var_aux = SeteSegmts[valor] | var_aux;     //Coloca a sequencia nos 7 BmS.
    LATB = var_aux;                             //Coloca valor na saída do porto B
}
```

---

## 3.2 Ligações necessárias

Para que o motor de passo possa funcionar de acordo com a sequência descrita na Tabela 1 é necessário aplicar ao sistema um interface de potência. Este interface é responsável por transformar as grandezas fornecidas pelo  $\mu$ controlador em grandezas compatíveis com o motor de passo. O interface de potência utilizado, usualmente designado por driver de potência será um L293 (quatro meias pontes em H).

A Figura seguinte mostra as ligações necessárias entre o  $\mu$ controlador e o driver de potência (L293) e entre o L293 e o motor de passo.

## 4 Trabalho a realizar

Deverão ser cumpridas as seguintes tarefas:

1. Elabore uma aplicação (programa e circuito) que faça rodar o motor de passo continuamente no sentido dos ponteiros do relógio. O controlo deve ser do tipo meio-passo, devendo haver um intervalo tempo

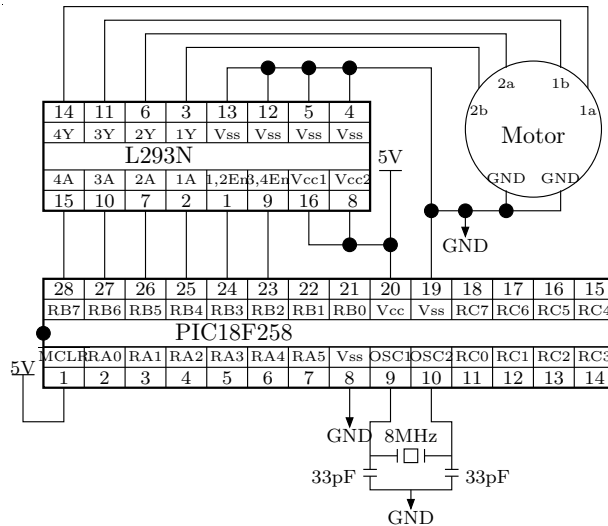


Figura 2: Esquema de ligações entre o  $\mu$ controlador e o motor de passo

suficiente entre cada meio-passo, de modo a que cada deslocamento do motor seja perceptível pelo ser-humano.

2. Altere a aplicação de forma a que os deslocamentos ocorram a uma frequência de 1Hz. Utilize para tal uma Interrupção do Timer 0.
3. Altere a aplicação para fazer o mesmo mas com controlo de direcção. Para rodar no sentido contrário a sequência fornecida na Tabela 1 deverá ser corrida ao contrário. O controlo de direcção deve ser feito utilizando a interrupção externa 1. Note que sempre que o comando de alteração de direcção for activado através da interrupção externa, a sequência deverá começar a ser corrida no sentido contrário, partindo do ponto onde se encontra.

**Tarefa final:** Implementar um controlador de motor de passo que deverá funcionar da seguinte forma:

- O valor do intervalo de tempo que deve decorrer entre cada deslocamento do motor de passo deve ser introduzido utilizando as rodas de numeração existentes no kit digital. Cada dígito deverá ser ligado a um *nibble* do porto C. Note que 01 corresponde a 100ms e 99 corresponde a 9,9s.
- Quando o valor estiver correcto, o utilizador carrega no interruptor monoestável A, que se encontra ligado à entrada de interrupção externa 0. O  $\mu$ controlador deve ler os dois dígitos introduzidos (código BCD) e a rotina de temporização deve ser adaptada ao valor introduzido.
- Neste momento devem ser integradas as tarefas realizadas anteriormente, devendo ser possível controlar a direcção (através da interrupção externa 1) e a velocidade do motor de passo.

Deverá ser entregue um fluxograma ou pseudo-código, o código em assembly o mais comentado possível, e o desenho do circuito.

## 5 Avaliação

A avaliação será feita pelo docente ao longo da aula. A cada tarefa cumprida o grupo deverá chamar o docente que avaliará a aplicação, e a resposta aos pedidos adicionais. A resposta a todos os requisitos de cada tarefa perfaz 15% do valor do trabalho prático. A resposta à tarefa final corresponde a 25% do valor do trabalho prático.