

Controlador de Motor de Passo baseado num 8051

1ª Parte – Implementação do circuito de accionamento do motor de passo

Dimensione o circuito de accionamento do motor de passo N82100. Seleccione o transformador a utilizar e dimensione e implemente o rectificador de onda completa a utilizar para alimentar o circuito de accionamento. Projecte o rectificador para um *ripple* de 15%.

Implemente em “*vero board*” o circuito de accionamento projectado.

Simule, recorrendo a resistências de cromo-níquel, os enrolamentos do motor de passo. Para tal deve determinar a resistência e a indutância do motor de passo.

Projecte e construa um circuito digital que gere a sequência do accionamento “*wave excitation*” nas resistências de cromo-níquel. Verifique, recorrendo ao osciloscópio, se o circuito de accionamento funciona correctamente e se a sequência de pulsos gerada pelo circuito digital está correcta.

Substitua as resistências de cromo-níquel pelo motor de passo.

- a) Como procederia para aumentar a frequência de rotação do motor de passo?
- b) Qual a frequência máxima de rotação?
- c) O que sucede ao “*holding torque*” à medida que a frequência aumenta? Justifique.
- d) Explique as diferenças entre a onda visualizada aos terminais da resistência de cromo-níquel e a onda visualizada aos terminais do enrolamento do motor de passo.

2ª Parte – Implementação das rotinas de comunicação série

Implemente, utilizando a placa EV-8051-FX, utilizando a linguagem *assembler* as seguintes rotinas:

1. Configurar comunicações – rotina que realiza as configurações necessárias à comunicação série entre o 8051 e o PC;
2. Enviar caracter – rotina que permite enviar um caracter para o PC;
3. Receber caracter – rotina que recebe um caracter proveniente do PC.

Teste as rotinas implementadas usando o simulador disponível (*8052 Simulator*).

Utilizando as rotinas desenvolvidas implemente um programa que realize o *echo* do caracter recebido. Ou seja, que devolva o caracter que recebeu via porto série. Para testar o funcionamento deste programa utilize *software* de comunicações para o PC (p. ex: ProComm).

Implemente o mesmo programa utilizando a interrupção série do 8051.

Implemente, recorrendo à programação directa dos registos da UART do PC, um programa em C que permita enviar para e receber caracteres do 8051.

Verifique se o programa funciona correctamente com as rotinas desenvolvidas para o 8051.

3ª Parte – Implementar no 8051 o “*wave excitation*”

Pretende-se gerar no porto 1 do 8051 os pulsos digitais que implementem o accionamento “*wave excitation*”. A frequência de passo deve ser controlada pelo PC. Isto pode ser conseguido enviando do PC para o 8051 um ou mais bytes que indiquem a frequência de passo desejada. Para tal baseie-se no exemplo que se segue.

Exemplo:

```
; Suponha a seguinte tabela
Org      400h
db       '00000001'
db       '00001000'
db       '00000010'
db       '00000100'
```

```
; e o programa que gera as ondas
Gera:
mov     dptr,#0400h    ;dptr aponta inicio da tabela
GeraLoop:
movx   a,@dptr        ;lê primeira linha da tabela
mov    p1,a           ;envia a onda para o porto 1
inc    dptr
mov    a,#4
subb  a,dpl
jnz   GeraLoop        ; não é final de tabela
jmp   Gera
```

Se P1.0 estiver ligado a Q1, P1.1 a Q2, P1.2 a Q3 e P1.3 a Q4, o programa apresentado no exemplo realiza o accionamento do tipo “*wave excitation*”. No exemplo, a frequência de passo é fixa. Para alterar a frequência de passo deve utilizar o temporizador 0 no modo 1 (16-bit). Desenvolva uma aplicação em C, baseada nas rotinas de comunicação desenvolvidas, que permita seleccionar a velocidade de rotação do motor de passo.

4ª Parte – Implementar no 8051 todos os accionamentos

Utilizando um esquema semelhante ao da 3ª parte, implemente agora as tabelas que permitam gerar todos os accionamentos. A selecção da tabela, ou seja do accionamento a utilizar, bem como a velocidade de rotação (em r.p.m.) é realizada pelo PC.

5ª Parte – Implementação do controlador do motor de passo

Pretende-se seleccionar, a partir do PC, a velocidade de rotação do motor, o tipo de accionamento, o sentido de rotação e o número de passos. Procure minimizar o espaço ocupado de memória de dados do 8051.

Desenvolva uma aplicação em linguagem C que permita ao utilizador seleccionar os parâmetros pretendidos. Utilize o motor de passo para verificar o funcionamento da aplicação.

6ª Parte – Implementar a solução final

Implemente a solução final para o controlador do motor de passo. Os objectivos mínimos a atingir são:

1. Programação do número de voltas;
2. Programação do número de passos;
3. Programação da velocidade de rotação;
4. Programação do sentido de rotação;
5. Comando de paragem/arranque;
6. Programabilidade: Ex: 5 voltas no sentido horário, 4 voltas no sentido anti-horário e 400 passos no sentido horário.

Desenvolva a aplicação em linguagem C, que permita ao utilizador controlar todos estes parâmetros.

Avaliação

A avaliação é baseada na informação recolhida pelos docentes em cada uma das aulas práticas e pela avaliação parcelar do trabalho prático (peso de 1/2) em cada uma das suas fases, pelo relatório (peso de 1/3) a elaborar pelos alunos e por um mini-teste (peso de 1/6) a realizar no final do trabalho prático. O peso do trabalho prático na nota final é calculado tendo como base a duração (em número de aulas) do mesmo em relação ao número de aulas total.

Bibliografia

1. Manual do Motor de Passo, “Stepper Motors Series N82100”
2. Yeralan S.e Ahluwalia A., *Programming and interfacing the 8051 microcontroller*, Addison-Wesley Publishing Company, 1995, BPG 161272
3. “MFC-51 Programmer’s Guide”, Intel
4. “MFC-51 Embedded Applications”, Vol. 1, Intel, 1995/1996
5. Afonso, J. “Motores de Passo”, Texto de Apoio à disciplina de Máquinas Eléctricas e Actuadores, 1995/96, Azurém

6. Cabral, J., Textos de Apoio à disciplina de Sistemas Digitais e Microprocessadores, 1996/97, Azurém