



Exercícios VI – Programação pela porta paralela

Será utilizado o kit didático para programação pela porta paralela. O kit contém leds para indicar a utilização dos pinos de dados (2 a 9) e de controle (1, 14, 16 e 17), e *push buttons* para a utilização dos pinos de status (10, 11, 12, 13 e 15). Os pinos de dados são identificados pelos leds de cor vermelha e verde e os pinos de controle pelos leds amarelos.

Faça o *download* da “caixa preta” no endereço www.harpi.eng.br, seção disciplinas, e realizar a ativação da *dll* necessária à chamada das funções de linguagem C para usar a porta paralela. A “caixa preta” consiste em linhas de código para a ativação desta *dll*.

1. Desenvolva um programa para a utilização dos pinos de dados (2 a 9) da porta paralela que simule a utilização de um motor de passo. Utilize os *nibbles* (1 *nibble* = 4 *bits*) mais significativo e o menos significativo de forma separada.

	Passo 1				Passo 2				Meio passo			
1° Loop	1	0	0	0	1	1	0	0	1	0	0	0
2° Loop	0	1	0	0	0	1	1	0	1	1	0	0
3° Loop	0	0	1	0	0	0	1	1	0	1	0	0
4° Loop	0	0	0	1	1	0	0	1	0	1	1	0
									0	0	1	0
									0	0	1	1
									0	0	0	1
									1	0	0	1

Dica: para melhor visualização da ativação de cada bit deve-se utilizar um tempo entre cada ativação. Faça uma pesquisa e identifique a função da linguagem C que pode efetuar essa tarefa em seu programa. Utilize a função **outportB(valor, endereço)**; sendo endereço 378 em hexadecimal.

2. Desenvolva um programa para a utilização dos pinos de controle (1, 14, 16 e 17) da porta paralela. Utilize-os da forma que achar apropriado. Uma opção é fazer a mesma simulação solicitada na questão anterior.

Dica: deve-se prestar atenção na ativação destes bits, pois apenas um bit é ativado com nível lógico alto (1). Os outros três bits são acionados com o nível lógico baixo (0). Veja o anexo para maiores dúvidas. Utilize a função **outportB(valor, endereço)**; sendo endereço 37A em hexadecimal.

3. Desenvolva um programa para a utilização dos pinos de status (10, 11, 12, 13 e 15) da porta paralela. Faça um programa em linguagem C para realizar um loop infinito e efetuar a leitura na entrada de cada pino. A entrada de cada pino é simulada com a ativação dos botões que realizam curto – circuito em seus respectivos pinos.

Dica: deve-se prestar atenção na ativação destes bits, pois um bit é ativado com nível lógico baixo (0). Veja o anexo para maiores dúvidas. Utilize a função **inportB(endereço)**; sendo endereço 379 em hexadecimal. Lembre-se que esta função retorna valor.

Departamento de Engenharia e Arquitetura

Disciplina: Linguagem de Programação I

Prof. Rafael G. B. de Araújo



Anexo

Sobre os pinos de dados: pinos de 2 à 9, normais, i.e., nível lógico alto (um) ativa. A letra D significa registrador de DADOS, e o número significa a posição do bit no byte.

Pinos de Dados								
Identificação	D7	D6	D5	D4	D3	D2	D1	D0
Pino	9	8	7	6	5	4	3	2
Posição	7	6	5	4	3	2	1	0
Byte	1	1	1	1	1	1	1	1

Sobre os pinos de controle: C0 (*Strob* – pino 1), C1 (*Auto Feed* – pino 14) e C3 (*Slct in* – pino 17) que são invertidos, i.e., nível lógico baixo (zero) ativa. Apenas o bit C2 (*Init* – pino 16) é normal, i.e., nível lógico alto (um) ativa. A letra C significa registrador de CONTROLE, e o número significa a posição do bit no byte.

Pinos de Controle								
Identificação	Nenhum				C3	C2	C1	C0
Pino					17	16	14	1
Posição	7	6	5	4	3	2	1	0
Byte	0	0	0	0	1	1	1	1

Sobre os pinos de status: S7 (*Busy* – pino 11) é invertido, i.e., nível lógico baixo (zero) ativa. Os outros bits, S6 (*Ack* – pino 10), S5 (*Paper end* – pino 12), S4 (*Slct out* – pino 13) e S3 (*Error* – pino 15) são normais, i.e., nível lógico alto (um) ativa. A letra S significa registrador de STATUS, e o número significa a posição do bit no byte.

Pinos de Status								
Identificação	S7	S6	S5	S4	S3	Nenhum		
Pino	11	10	12	13	15			
Posição	7	6	5	4	3	2	1	0
Byte	0	1	1	1	1	1	1	1

Os pinos 18 ao 25 são *ground* (Terra).