

Exercícios para a Prova 2

Exercício 1) Escreva uma rotina que gere, pelos pinos P1.0 e P1.1, as formas-de-onda mostradas abaixo. Use os timers e as interrupções. Seja cuidadoso com os tempos.

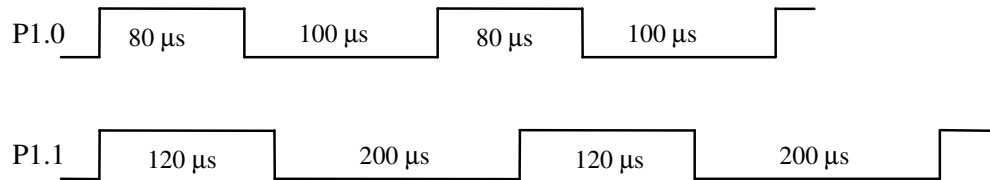


Figura 1. Formas-de-onda a serem geradas no Exercício 1.

Exercício 2) Escreva uma rotina que gere, pelos pinos P1.0 e P1.1, as formas-de-onda mostradas abaixo. Use os timers e as interrupções. Seja cuidadoso com os tempos.

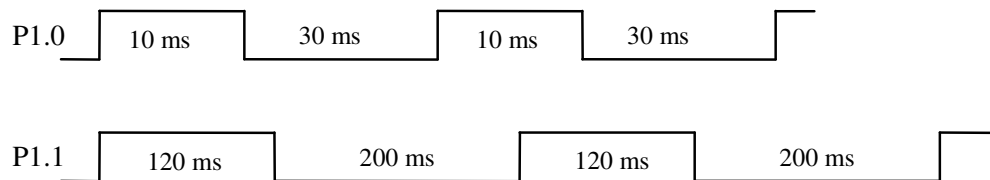


Figura 2. Formas-de-onda a serem geradas no Exercício 2.

Exercício 3) O circuito da Figura 3 apresenta uma solução para acionar um *display* de 7 segmentos com o emprego da porta P1. Escreva uma rotina para fazer o *display* contar na frequência de 1 Hz. Use uma tabela para simplificar os códigos enviados para o *display*. O CI 74LS244 funciona como um *buffer* não inversor. Note que o *display* está conectado a VCC, ou seja, cada segmento acende quando se apresenta nível baixo no pino correspondente.

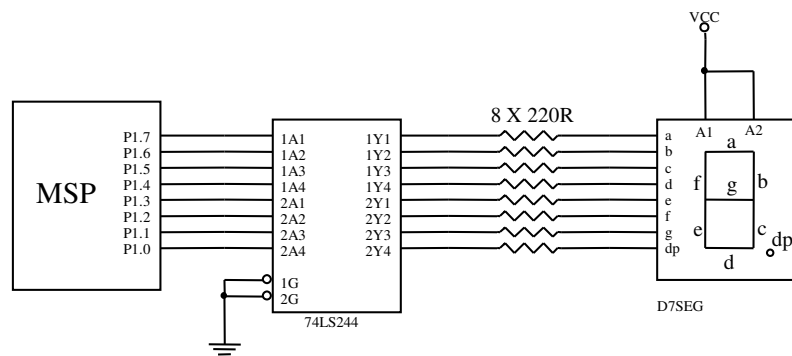


Figura 3. Display de 7 segmentos sendo acionado pela porta P1.

Exercício 4) O circuito da figura 4 apresenta um *display* de 7 segmentos sendo acionado com o emprego da porta P1, mas com a alimentação (A1 e A2) controlada pelo pino P3.4, através de um transistor. Assim, através de P3.4, o *display* pode ser ligado ou desligado. Esses *displays* consomem muita energia. Por exemplo, se for considerado que cada segmento consome 15 mA, então, com o *display* todo aceso, o consumo chega a 120 mA. Uma técnica para economizar energia é não deixar o *display* aceso permanentemente, mas sim fazê-lo piscar numa frequência superior a 30 Hz, pois, acima dessa frequência, o olho humano não percebe variação luminosa ou cintilação. Escreva uma rotina que faça o *display* contar na frequência de 1 Hz e, ao mesmo tempo, que o ligue e o desligue na frequência de 100 Hz.

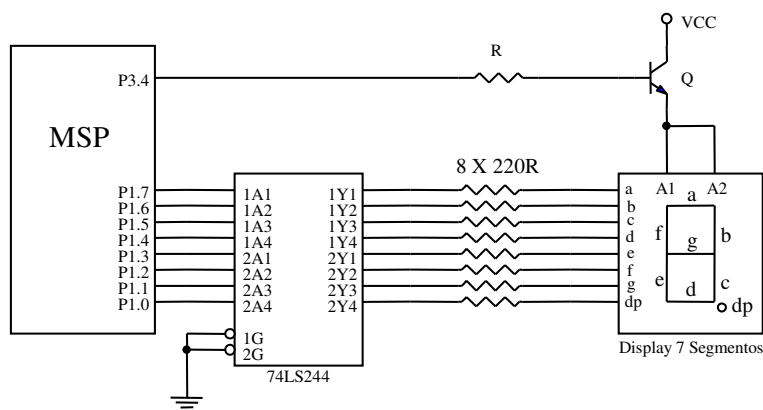


Figura 4. Display de 7 segmentos acionado pela porta P1 e com alimentação controlada pelo pino P3.4.

Exercício 5) Um teclado nada mais é que a reunião de uma grande quantidade de chaves. Para economizar a quantidade de portas, é comum organizar o teclado sob a forma matricial. A figura a seguir apresenta um teclado de 12 teclas conectado à porta P1. Em cada cruzamento de fios, existe uma pequena chave que, quando acionada, fecha contato entre eles. Assim, a tecla “1” está no cruzamento dos fios ligados a P1.4 e P1.3. Já a tecla “5” é conseguida com o cruzamento de P1.5 com P1.2.

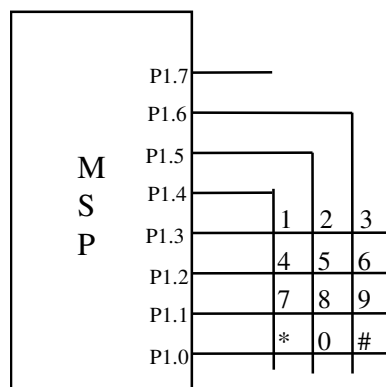


Figura 5. Sugestão de um teclado tipo telefônico conectado à porta P1.

O princípio para leitura do teclado é muito simples. Inicialmente, toda a porta P1 é colocada em nível alto. Com P1.4 = 0, é possível verificar o estados das teclas 1, 4, 7, *. Por exemplo, se a tecla 7 estiver pressionada, então apenas o pino P1.1 estará em nível baixo. Em seguida é feito P1.4 = 1 e P1.5 = 0, agora verifica-se o estado das teclas 2, 5, 8, 0. Assim por diante. Pelos pinos P1.4, P1.5 e P1.6 é feita a varredura (*scan*) e o resultado é lido por P1.0, P1.1, P1.2 e P1.3. O MSP, de tempos em tempos, faz uma varredura do teclado para verificar se alguma tecla foi acionada.

Tabela 1. Códigos de varredura para o teclado proposto.

P1.6	P1.5	P1.4	Teclas varridas
0	1	1	3, 6, 9, #
1	0	1	2, 5, 8, 0
1	1	0	1, 4, 7, *

A Figura 6 apresenta exemplos de varredura do teclado. Para saber qual tecla foi acionada, basta verificar onde surgiu o zero. Se não for lido qualquer valor zero, então nenhuma tecla foi acionada. Por outro lado, se forem lidos dois ou mais zeros, é porque duas ou mais teclas estão acionadas. Nesse caso, é costume rejeitar a leitura. É preciso tomar cuidado para não se detectarem múltiplas vezes o acionamento de uma mesma tecla. Por isso, para que se aceite uma nova tecla, o teclado precisa obrigatoriamente passar pelo estado de nenhuma tecla acionada.

Considerando o teclado sem rebotes, escreva uma rotina de interrupção que faça a varredura do teclado a cada 10 ms e armazene na variável global “tecla” o código da última tecla acionada. Assim, o programa principal precisa apenas consultar essa variável “tecla”. Para evitar múltiplas respostas à mesma tecla, o programa principal, após ler uma tecla válida, escreve ZERO nesta variável.

- nenhuma tecla acionada → 0;
- uma tecla acionada → código ASCII correspondente.

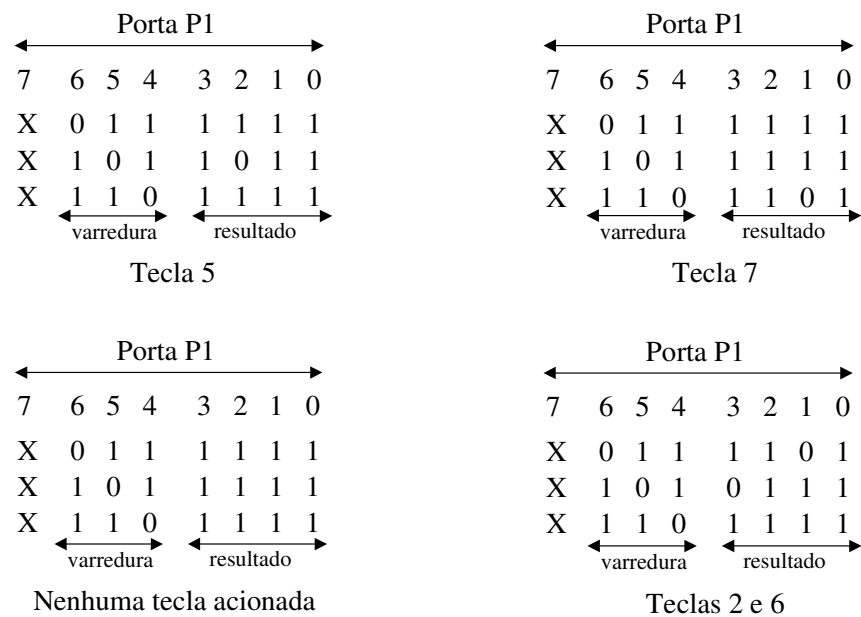


Figura 6. Quatro exemplos de leitura do teclado.

Exercício 6) Refaça o Exercício 5, mas agora considerando que o teclado tenha rebotes.

Exercício 7) Um motor de corrente contínua (motor CC) oferece muito mais torque e velocidade que um motor de passo. Por outro lado, pede um controle um pouco mais sofisticado. A Figura 7, desenhada a partir do CI L298, apresenta uma ponte “H” para controle de um motor CC. Se os transistores Q1 e Q4 estiverem ligados, diz-se o motor está girando no sentido normal. Já com os transistores Q3 e Q2 ligados, o giro será no sentido reverso. Os diodos D1, D2, D3 e D4 devem ser adicionados externamente e têm a função de proteger os transistores das altas tensões geradas quando se corta a energia das bobinas do motor.

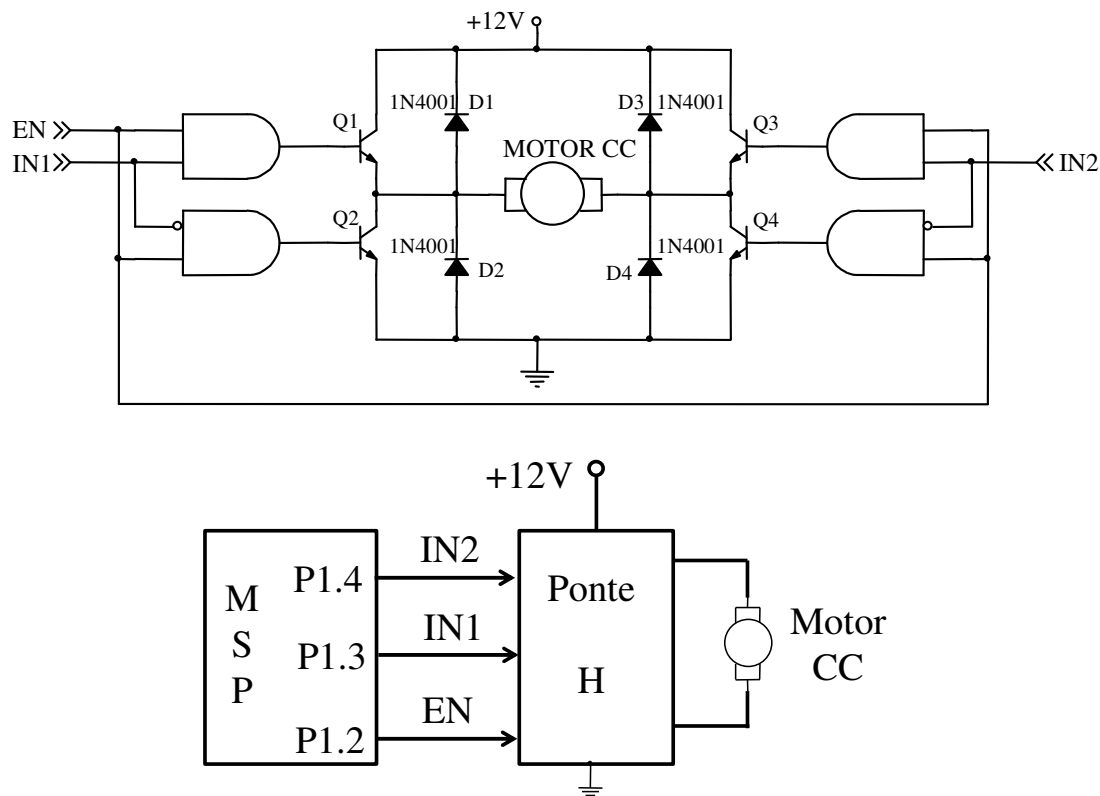


Figura 7. Ponte “H” (CI L298) para controlar um motor CC.

O controle dos transistores é feito com uma habilitação, identificada por EN, e duas entradas, denominadas IN1 e IN2. Se EN = 0, todos os transistores estão desligados e, com isso, o motor está livre. Se EN = 1, através de IN1 e IN2 controla-se o sentido da rotação do motor. A tabela a seguir resume as possibilidades de operação. O CI L298 oferece duas pontes “H”, trabalha com alimentação de até 46 V e corrente de até 4 A. O fabricante é a STMicroelectronics e sua página pode ser acessada em www.st.com.

Tabela 2. Funções disponibilizadas pela ponte “H”.

EN	IN1	IN2	Função
0	X	X	motor livre
1	1	0	rotação normal
1	0	1	rotação reversa
1	1	1	parada rápida
1	0	0	parada rápida

A maneira mais fácil de se controlar a velocidade de rotação de um motor CC é através da modulação por largura de pulso, designada pela sigla PWM (*Pulse Width Modulation*). Deve-se usar períodos de 100 ms e resolução de 100 divisões. Em outras palavras, a resolução será de 1 ms. A Figura 8 ilustra quatro situações da modulação PWM. Com modulação de 0%, o motor está parado, já com modulação de 100%, o motor está na velocidade máxima. Entre 0% e 100%, convencionam-se 100 passos, disponibilizando assim diversos valores intermediários de rotação.

Considere as seguintes conexões:

- P1.2 = EN
- P1.3 = IN1
- P1.4 = IN2

Escreva uma função `crtl_motor()`, prototipada abaixo, que controle o sentido e a velocidade de rotação do motor.

```
void ctrl_motor (char PWM, char sentido);
```

- PWM = modulação PWM, valor de 0 até 100
- Sentido = 0 → sentido normal
- sentido = 1 → sentido reverso

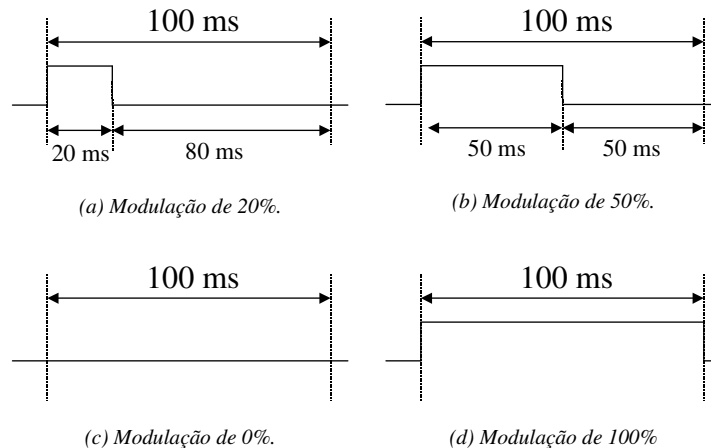


Figura 8. Quatro situações de modulação PWM.

Exercício 8) Uma forma de controlar-se a potência entregue a dispositivos ligados à rede elétrica, tais como lâmpadas incandescentes, aquecedores elétricos ou motores CA, é com o emprego de tiristores. Neste exercício será empregado um tipo de tiristor denominado TRIAC. A figura a seguir apresenta uma sugestão para controle da potência entregue à carga RL com o emprego de um TRIAC (Q1). Como o TRIAC vai operar diretamente com a rede elétrica, é bom prever uma proteção para o resto do circuito, que no caso foi feita com o foto-TRIAC ISO1 (MOC3020), que oferece isolamento óptico.

Na Figura 9, sempre que o pino P1.0 for colocado em nível baixo, o foto-TRIAC conduz, o que leva à condição de disparo para o terminal de “Gate” do TRIAC, que passa a conduzir. Se o TRIAC conduz, a energia da rede chega até a carga RL. Para evitar disparos indevidos, adiciona-se o capacitor C1. Para eliminar um pouco os ruídos gerados pelo chaveamento do TRIAC, usa-se o filtro formado por C2 e R4.

Para que se tenha noção dos instantes em que se deve disparar o TRIAC, é necessária a sincronização com a rede elétrica. Uma maneira simples de se conseguir tal sincronização é através de um circuito que detecte as passagens pelo zero, ou seja, um circuito que avise quando a tensão alternada da rede estiver passando pelo valor zero. A Figura 10 apresenta este circuito.

O transformador com tomada central (*center tap*) e os diodos D1 e D2 fazem um retificador de onda completa. Enquanto houver tensão, o acoplador óptico IS02 conduz e, com isso, tem-se um nível baixo em P1.2. Mas, quando a rede estiver passando pelo zero, não haverá tensão para acender o *led* do acoplador e, com isso, o foto-transistor vai para o corte. Nesse caso, graças ao resistor de *pull-up*, o pino P1.2 irá para nível alto, como mostrado na Figura 10. Os pulsos de sincronização com a rede surgirão em conformidade com a Figura 11. A intenção é usar este pino para provocar uma interrupção.

O controle de potência é feito a cada semiciclo. O instante de disparo do TRIAC, comandado através de P1.0, deve tomar como referência a passagem pelo zero (pino P1.1), como mostrado na Figura 12. Se o disparo for logo após a passagem pelo zero, como é o caso do pulso P1, a carga recebe quase que todo o semiciclo. Um disparo um pouco mais tarde, como acontece com P2, entrega um pouco menos de potência. Por outro lado, se o disparo for muito tarde, como é o caso de P4, quase nenhuma potência chega até a carga. Desta forma, através do instante de disparo do TRIAC, controla-se a potência que é entregue à carga.

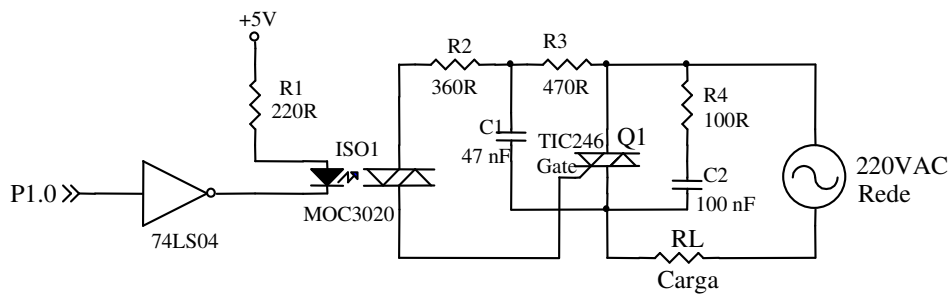


Figura 9. Circuito para controle da potência entregue à carga RL.

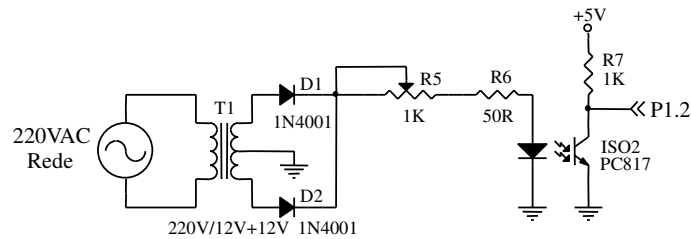


Figura 10. Circuito detector de passagem pelo zero.

Suponha que, além dos circuitos apresentados, existam duas teclas ligadas aos pinos P2.1 e P1.1 que operam da seguinte forma:

- P2.1 (SW1) = aumentar a potência em 1/100
- P1.1 (SW2) = diminuir a potência em 1/100

Escreva uma rotina que inicie com a carga recebendo potência zero. Controlado pelo acionamento das teclas, essa potência será aumentada ou diminuída, dentro de uma faixa de 0 a 100. O semiciclo da rede tem a duração de 8,33 ms. Para simplificar, deve-se dividir o semiciclo em 100 passos, ou seja, cada passo tem a duração de 83,3 μ s. Lembre-se de que:

- Potência = 0% → sem pulso de disparo
- Potência = 25% → pulso de disparo em 6,25 ms
- Potência = 50% → pulso de disparo em 4,17 ms
- Potência = 75% → pulso de disparo em 2,083 ms
- Potência = 100% → pulso de disparo logo após a passagem pelo zero

Obs.: o pulso de disparo gerado por P1.0 não pode ser muito estreito; um valor perto de 50 μ s já deve ser suficiente.

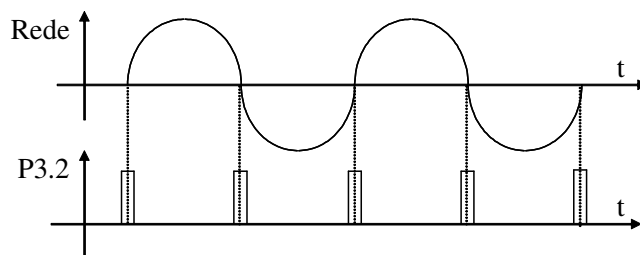


Figura 11. Pulsos gerados em P1.1 pelo circuito detector de passagem pelo zero.

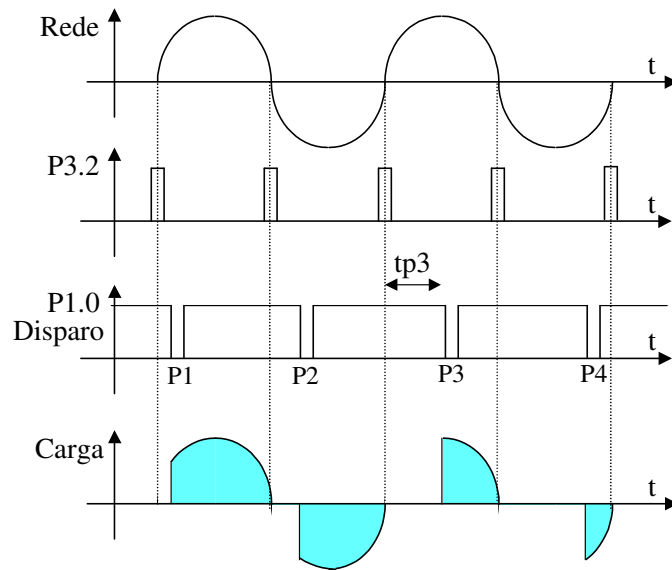


Figura 12. Quatro situações de disparo do TRIAC e a quantidade de potência entregue à carga.

Exercício 9) No exercício anterior, dividiu-se o semiperíodo em 100 intervalos de 83,3 μ s. Entretanto, cada um desses intervalos não corresponde a 1/100 da potência total. Refaça o exercício anterior recalculando os instantes de disparo de tal forma que realmente se trabalhe com incrementos de 1/100 da potência plena. Sugestão: Considerando a carga RL, a senóide de frequência f e amplitude A e o intervalo de condução de 0 a ϕ , tem-se que a potência na carga é dada por:

$$P = \frac{A^2}{RL} \int_0^{\phi} \sin^2 2\pi f t \cdot dt$$