

LISTA DE EXERCÍCIOS PARA A TERCEIRA AVALIAÇÃO.

Os programas solicitados são em linguagem C do compilador CCS C. Para facilitar, considere a utilização de um cristal de quartzo de 4 MHz (Fosc = 4 MHz).

- 1) Desenhe um circuito com o decodificador 7448 e um demultiplexador para acionar quatro displays de 7 segmentos de catodo comum, utilizando a porta D. A partir do desenho, faça o programa em C.
- 2) Utilizando a porta B e a porta D, elabore um programa em C para acionar quatro displays de anodo comum com quatro transistores, tal como mostrado no slide de página 24. Nos displays serão apresentados os algarismos da contagem de 0 a 9999, sendo esta contagem considerada para cada borda de subida de um sinal externo conectado no pino RB0/INT.
- 3) Altere o programa anterior, utilizando o TIMER0 como contador externo via pino RA4.
- 4) Idem, utilizando agora o TIMER1 como contador externo via pino RC0.
- 5) Utilizando o TIMER0 como temporizador com clock interno, faça um programa para verificar constantemente se o pino RD7, configurado como entrada, se mantém em nível alto em pelo menos 50 ms. Caso afirmativo, o estado do RD6 é colocado em nível alto, caso contrário, o seu estado é baixo.
- 6) Utilizando o TIMER0, faça um programa para ler o estado do pino RD7 a cada 26 us durante 10 vezes. A primeira leitura estará armazenada numa flag (Ex: #bit flag1 = 0x20.0). A segunda leitura estará no bit menos significativo de uma variável de 8 bits, enquanto que a terceira leitura estará no segundo bit menos significativo e assim seguindo essa lógica até o oitavo bit mais significativo da variável (Sugestão: Utilize o comando >> da linguagem C). A décima leitura do pino estará em uma outra flag (Ex: #bit flag10 = 0x20.1). Essas 10 leituras serão sempre realizadas após o pino RD7 estiver em nível alto durante 50 ms.
- 7) Altere a solução da quarta questão para utilizar a interrupção do TIMER0 no intuito de gerar a temporização de 4 ms para a multiplexação de cada display.
- 8) Utilizando a interrupção do TIMER0, faça um programa para gerar uma onda quadrada de 10 kHz, 1 kHz, 100 Hz, 10 Hz e 1 Hz respectivamente nos pinos RD7, RD6, RD5, RD4 e RD3.
- 9) Utilizando a interrupção do TIMER1, faça um programa para implementar um relógio com segundos (variável **segundos**), minutos (variável **minutos**) e horas (variável **horas** que deve estar entre 0 a 23). Utilize um cristal de quartzo de 32768 Hz para conectar entre os pinos RC0 e RC1 para ser a fonte de clock para o TIMER1 e facilitar a temporização precisa de 1 segundo.
- 10) Utilizando a interrupção do TIMER2, faça um programa para comutar o estado do pino RD7 a cada 800 us.
- 11) Utilizando as interrupções dos três timers, configure o TIMER0 para gerar uma interrupção a cada 100 us, o TIMER1 a cada 200 us e o TIMER2 a cada 300 us. A ordem de prioridade das interrupções é: TIMER0, TIMER1 e TIMER2, sendo a interrupção do TIMER0, a interrupção de maior prioridade e a do TIMER2, a de menor prioridade. A rotina de tratamento de interrupção do TIMER0 comuta o

estado do pino RD7, a do TIMER1 comuta o estado do pino RD6 e por fim a do TIMER2 comuta o estado do pino RD5. Nesse caso, é possível duas dessas interrupções ocorrerem ao mesmo tempo? Se sim, quais? É possível também a ocorrência dessas três interrupções ao mesmo tempo? Se sim, quais e por que?

- 12) Escreva um programa em C para o PIC16F877A a fim de gerar um sinal PWM na frequência de 20 kHz. A cada 0,1 segundos, a razão cíclica é aumentada em 10 %. Quando for atingido o *duty* no tempo de 100 %, o *duty* reinicia em zero, repetindo o ciclo até 100%. No mesmo programa, faça a medição do tempo em nível alto do sinal PWM. Se este tempo for superior a 25 us, o estado do pino RD0 permanece em nível alto, caso contrário em nível baixo.
- 13) Faça um programa para gerar um sinal PWM em 20kHz de frequência no pino RC1, no qual o seu *duty* varie conforme a largura de um pulso, conectado no pino RC2 do microcontrolador.
- 14) Faça um programa para colocar em nível alto o estado do pino RC2/CCP1 toda vez que a contagem do TIMER1 for 10. Após, isso o estado deverá estar em nível baixo.
- 15) Escreva um programa generalizado capaz de medir o período de um sinal, podendo este possuir qualquer valor de período entre 100 us e 50000 us. O período deste sinal será armazenado sempre em uma variável do programa.
- 16) Faça o mesmo do item anterior, modificando apenas o tipo de evento a ser medido. No caso deste, para medir o tempo em nível alto do sinal ao invés do período.
- 17) Implemente um PWM com frequência de 20kHz cujo duty cycle varie de forma senoidal com o tempo em 100Hz de frequência. O duty central deve ser de 50%. Utilize um dos módulos CCP.
- 18) Faça o mesmo da questão 17, desde que a frequência do PWM senoidal possa variar na frequência de um sinal conectado em um dos pinos relacionados ao módulo CCP.
- 19) Faça um programa para gerar três ondas quadradas de 60 Hz de frequência cada, defasados entre si de 120° com duty de 50 %.
- 20) Faça um programa para configurar a porta serial do microcontrolador com os seguintes parâmetros: Baud rate: 9600 bps, 8 bits de dado, sem paridade e 1 stop bit, pino de transmissão RC6 e de recepção o RC7. Em seguida, crie uma função que receba dois vetores de inteiros, sendo os inteiros de 8 bits. O primeiro vetor, V1, tem sempre 2 bytes e o segundo vetor, V2, tem sempre 4 bytes. Na função, os dois bytes do vetor V1 são transmitidos pela porta serial e logo em seguida do envio desses bytes, a mesma função preenche o vetor V2 com os 4 primeiros bytes recebidos. Após isso, se a primeira posição do vetor 1 for igual a do vetor 2, então a função retorna 1, senão 0. Dica: Utilize ponteiro para apontar para cada vetor, ou vejam <https://www.youtube.com/watch?v=SAhR1h3LpDY>.
- 21) Aproveitando o programa anterior e o alterando, faça um programa para enviar e receber dados do PIC a partir do terminal serial do computador¹. Da seguinte forma:
 - I. O PIC ao receber a letra 'a' e em seguida a letra 'n', os quatro bytes de V2 serão transmitidos do PIC ao computador. Como mostrado na aula, utilize também a função `printf("%u, ", V2[i])`.

¹

II. Ao receber a letra 'b' e '0', nessa sequência, o vetor V1 é preenchido com o valor 0x81 e 0xAA nesta ordem, ou seja, $V1 = \{0x81, 0xAA\}$; Se receber a letra 'b' e '1', nessa sequência, o vetor V1 fica $V1 = \{0x81, 0x55\}$;

III. Ao receber a letra 'c' e um caracter qualquer (x), nessa sequência, o vetor V1 é $V1 = \{0x82, x\}$;

¹ Utilize o terminal serial do Proteus para fazer essa verificação.

22) Explique com detalhes como funciona a comunicação serial UART (modo assíncrono full-duplex) entre três ou mais microcontroladores PIC, sendo esses microcontroladores conectados entre si utilizando seus pinos Rx(RC7) e Tx(RC6) (além do sinal de terra, Vss, obviamente). Explique, por exemplo, como um PIC pode se comunicar apenas com um específico e em um outro dado momento, conseguir se comunicar com o outro. A sua explicação deve constar como ocorre a detecção de endereço. Ver página 119 do *datasheet* do microcontrolador.

FIM DA LISTA!