

SEL0614/SEL0433
APLICAÇÃO DE MICROPROCESSADORES



Parte 2 – Microcontroladores PIC e Programação em Linguagem C
Atividade Semanal 8

Formato de entrega:

- A atividade pode ser feita em grupos.
- Apresentar as respostas das questões da seção “Pós-aula” (texto ou linhas de código comentadas) em editor de texto (arquivo pdf) ou em repositório do GitHub (neste caso enviar o link para o arquivo Readme.md em um arquivo de texto – não enviar diretamente o arquivo Readme.md na tarefa).

Pré-aula

- **Material de aula:** “Cap. 7 – Temporizadores e Interrupções”
- **Objetivos da aula:** programação de temporizadores e interrupções utilizando PIC18F4550, kit EasyPICv7 e SimulIDE, com atenção especial na configuração dos registradores (bit e byte) envolvidos.
- Códigos de exemplo - programas em Ling. C usados na aula (**disponível no e-Disciplinas**).

Durante a aula

- Conectar o **KitEasyPIC** ao PC por meio do cabo USB e ligar a placa (verificar se os jumpers de alimentação estão conectados à USB e ao 5V na placa).
- Abrir o MikroC PRO for PIC e acompanhar os programas que serão desenvolvidos durante a aula de Temporizadores e Interrupções.
- Implemente no SimulIDE um programa para piscar um LED (conectado à um dos pinos do PORTC) a cada 1 segundo, utilizando o temporizador Timer0 (TMR0) do PIC18F4550. Ajustar o clock do microcontrolador PIC18F4550 para 8 MHz no SimulIDE, carregar o firmware (arquivo hex resultante da compilação no software MikroC PRO for PIC) e realizar a simulação. Apresentar na resposta o programa em linguagem C comentado (somente as partes mais importantes) e print do circuito montado e simulação realizada no SimulIDE.

Pós-aula

Exercício 1

- Implemente no SimulIDE um programa para acionar uma saída (representada por um LED que irá piscar) a cada intervalo de tempo correspondente a contagem de tempo máxima do Timer3 (TMR3) do PIC18F4550 (ajustando todos os parâmetros com valores máximos para contagem, preescaler etc., e carregando valor inicial zero nos registradores acumuladores do TMR3). Ajustar o clock do microcontrolador PIC18F4550 para 8 MHz no SimulIDE, carregar o firmware (arquivo hex resultante da compilação no software MikroC PRO for PIC) e realizar a simulação. Apresentar na resposta o programa em linguagem C comentado e print do circuito montado e simulação realizada no SimulIDE. Este programa não faz o uso de interrupções.
- Implemente no SimulIDE um programa para acionar uma saída (representada por um LED que irá piscar) a cada intervalo de tempo correspondente a contagem de tempo máxima do Timer2 (TMR2) do PIC18F4550 (ajustando todos os parâmetros com valores máximos para contagem, preescaler, postscaler etc., e definindo valor máximo no registrador PR2 * observar a diferença do Timer2 para os demais temporizadores). Ajustar o clock do microcontrolador PIC18F4550 para 8 MHz no SimulIDE, carregar o firmware (arquivo hex resultante da compilação no software MikroC PRO for PIC) e realizar a simulação. Apresentar na resposta o programa em linguagem C comentado e print do circuito montado e simulação realizada no SimulIDE. Este programa não faz o uso de interrupções.

Exercício 2

- Implemente no SimulIDE um programa para mudar o estado de uma saída (representada por um LED conectado a um dos pinos do PORTD) sempre que for sinalizado um evento (representado por um botão quando é pressionado, o qual também pode ilustrar a leitura de um sensor conectado a uma entrada do microcontrolador). Neste caso, a implementação será realizada com uso de interrupções externas. Basicamente, o programa é um aprimoramento do Exemplo 1 demonstrado nas aulas anteriores (leitura de entrada e acionamento de saída – I/O digital), no qual um botão, ao ser pressionado, muda o estado de um LED. No entanto, ao invés de implementar linhas de código no programa principal para verificar se o botão foi pressionado, utiliza-se interrupções (evento por hardware que faz com que o programa salte automaticamente para a sub-rotina de tratamento somente quando o botão for pressionado). Deve-se utilizar a interrupção externa INT2. Portanto, o botão deve ser conectado na configuração pull-up (desta vez usar habilitar pull-up interno) ao pino do PIC18F4550 correspondente à INT2 (observar qual deve ser este pino por meio da pinagem do microcontrolador disponível no datasheet e no material de aula). Deve-se configurar o uso de interrupções por meio da ativação das chaves globais e específicas da INT2 (flags IE, IF, IP) nos registradores relacionados a esta interrupção (ver datasheet – seção Interrupts). Na sub-rotina de tratamento da interrupção (para onde o programa saltará quando o botão for pressionado), o LED deverá mudar seu nível lógico (inversão) e a INT2IF deve ser zerada (no PIC, este processo é

feito manualmente, isto é, as flags IF nunca são zeradas automaticamente). Ajustar o clock do microcontrolador PIC18F4550 para 8 MHz no SimulIDE, carregar o firmware (arquivo hex resultante da compilação no software MikroC PRO for PIC) e realizar a simulação. Apresentar na resposta o programa em linguagem C comentado e print do circuito montado e simulação realizada no SimulIDE.

- Implemente no SimulIDE um programa para acender um LED (conectado à um dos pinos do PORTD) após 5 eventos (representados por um botão na configuração pull-up pressionado por 5 vezes) utilizando o Timer1(TMR1) no modo **contador** (modo 16 bits). Para tanto, o botão deve ser conectado ao pino T13CKI do PIC18F4550 (ver qual é por meio da pinagem disponível no datasheet e no material de aula). Configurar o uso de interrupção (interrupção interna) do TMR1 (habilitar a chave de interrupções de periféricos e ativar flag IE e zerar IF referente ao TMR1). Neste contexto, a interrupção do TMR1 é um tipo especial de exceção interna, cuja sub-rotina de tratamento (para onde o programa irá saltar após a contagem de 5 eventos) será responsável por acender o LED, recarregar os registradores acumuladores do TMR1 e zerar a flag TMR1IF. Ajustar o clock do microcontrolador PIC18F4550 para 8 MHz no SimulIDE, carregar o firmware (arquivo hex resultante da compilação no software MikroC PRO for PIC) e realizar a simulação. Apresentar na resposta o programa em linguagem C comentado e print do circuito montado e simulação realizada no SimulIDE.

Exercício 3

O que acontece se o PIC18F estiver ocupado com uma interrupção de baixa prioridade e uma interrupção de alta prioridade ocorrer?

- a) O microcontrolador sempre atenderá primeiro às interrupções de alta prioridade, independentemente de uma interrupção de baixa prioridade já estar em andamento.
- b) O PIC18F4550 não possui um mecanismo de priorização de interrupções, e todas as interrupções são tratadas na ordem em que são geradas, sem considerar a importância ou a prioridade das interrupções.
- c) O microcontrolador pode continuar processando a interrupção de baixa prioridade até que ela seja concluída, mesmo que uma interrupção de alta prioridade seja gerada.
- d) As interrupções de alta prioridade podem interromper a execução de interrupções de baixa prioridade, mas após o processamento da interrupção de alta prioridade, o microcontrolador retorna automaticamente à interrupção de baixa prioridade.

Não será necessária a entrega dos códigos fontes em arquivos separados (“c” ou “hex”), referente aos programas gerados para cada exercício acima. Sendo assim, basta apresentar os programas em linguagem C e um print do circuito montado e simulação realizada no Simul IDE diretamente na resposta dos exercícios acima.

OBS: Devido a uma falha no SimulIDE, restrita a programas que usam o Timer0, recomenda-se utilizar a menor razão possível do Prescaler do Timer0 para obter a contagem de tempo. Por exemplo, para contagem de 1s, o código de exemplo disponibilizado no material de aula mostra a configuração para uma razão de 1:128. Neste caso, recomenda-se utilizar a razão de 1:32 (razão mínima para contagem de 1s) ao invés de 1:128. Dessa forma, devido a um erro do simulador, simulação apresentará valores de frequência de contagem mais próximos dos reais ao configurar o clock do microcontrolador para 8 MHz no simulador. Para testar os programas no kit EasyPIC, no entanto, pode-se usar qualquer razão que garanta a contagem, pois trata-se apenas de uma falha do simulador. Caso desejar, outros simuladores, como PICSIMLAB ou um software profissional como PROTEUS, podem ser usados no lugar do SimulIDE, sendo opcional. Como mencionado, erro acima é uma falha de suporte restrita ao Timer0 da família PIC18F. Os demais recursos do SimulIDE funcionam normalmente, sendo um recurso bastante acessível para testar os programas sem a necessidade da placa física.