

# SEL0433 APLICAÇÃO DE MICROPROCESSADORES



## Parte 2 – Microcontroladores PIC e Programação em Linguagem C

### Projeto 2 - Cronômetro Digital com Timer e Interrupções

#### Objetivos

- Explorar conceitos relacionados aos microcontroladores de 8 bits de arquitetura Harvard e set de instruções RISC.
- Exercitar o uso do software compilador e a programação básica de registradores SFR, I/O, e periféricos dos microcontroladores PIC18F em linguagem C.
- Desenvolver um projeto prático implementando funcionalidades de microcontroladores em linguagem C, tais como timers, interrupções, I/O e interface com display de 7 segmentos, utilizando o PIC18F4550.
- Implementar o projeto no kit EasyPIC v7 e realizar simulações por meio do software SimulIDE.

Esta atividade prática versa sobre o desenvolvimento do mesmo projeto do cronômetro digital (Projeto 1). A diferença é que agora o projeto deverá ser desenvolvido em Linguagem C, utilizando o microcontrolador PIC18F4550. É importante também destacar que, ao contrário do projeto anterior, temporizadores e interrupções serão utilizados para contagem de tempo e mudança das bases de contagem a partir do botão pressionado. As semelhanças que se mantiveram são o uso de display de 7 segmentos, a contagem de 0 a 9 em loop, uso de duas bases de tempo, e a mudança de bases de tempo por meio de botões. O projeto proposto permitirá comparações entre as características de microcontroladores de 8 bits CISC vs. RISC, linguagem de baixo nível Assembly com programas em linguagem C, e implementação prática vs. simulador.

#### Roteiro

#### I - Aplicação prática

Desenvolvimento de um projeto em linguagem C para PIC18F que explore os seguintes recursos: interrupções, temporização, portas de entradas e saídas, e periféricos (botões, LEDs e displays de 7 segmentos).

#### II - Requisitos do projeto

Criar um projeto no compilador MikroC PRO for PIC e escrever um programa em linguagem C que implemente os requisitos a seguir:

- Quando um botão conectado na porta **RB0** for pressionado, um display de **7 segmentos** ligado na **Porta D** deve contar (de 0 a 9 em loop) com período de **1s**.
- Quando um segundo botão, conectado na porta **RB1**, for pressionado, o mesmo display de 7 segmentos deve contar com período de **0,25s**.
- O display se inicia desligado e a contagem começa somente quando qualquer um dos botões é pressionado.

- Utilizar o temporizador **TMR0** com **Interrupção externa (nos botões)** para gerar as bases de tempo da contagem e provocar suas mudanças. Importa destacar que os botões (chaves) considerados agora são aqueles do tipo momentâneo, ou seja, deve-se pressionar e soltar logo em seguida pois o que conta para acionamento da interrupção é a mudança de borda.
- A frequência do clock do PIC deve ser de **8 MHz** (cristal externo HS).

### III - Atividade prévia (tarefa):

- Estudar os exemplos de códigos disponibilizados anteriormente para download e realizar os exercícios da Atividade Semanal revisando conceitos e testes realizados durante as **aulas sobre Temporizadores e Interrupções**, implementações dos exemplos nos softwares MikroC PRO for PIC e SimulIDE.
- Desenvolver o programa solicitado no item “II – Requisitos do Projeto” no MikroC PRO for PIC, compilar para o **PIC18F4550** e testar o projeto no **SimulIDE** (montar no SimulIDE o circuito com microcontrolador, 2 botões, e 1 display de 7 segmentos etc; carregar o firmware “.hex” do programa compilado no MikroC PRO for PIC; simular o projeto e testar o programa).
- **Implementação no kit EasyPIC v7 do laboratório:** a implementação prática no kit poderá ser considerada durante uma das aulas futuras e somente após a realização da atividade prévia, ou seja, somente quando projeto estiver funcional, após ter sido desenvolvido e testado no Simul IDE. **Para entrega no e-Disciplinas, no entanto, será considerada a versão testada no SimulIDE.**
- **OBS:** Devido a uma falha no SimulIDE, restrita ao Timer0, recomenda-se utilizar a menor razão possível do Prescaler do Timer0 para obter a contagem de tempo desejada. Dessa forma, a simulação apresentará valores de frequência de contagem mais próximos dos reais.
- Será possível agendar esclarecimentos de dúvidas e atendimento para auxílio neste projeto em outros horários, com o professor e/ou com monitores da disciplina. Da mesma forma, podem utilizar os laboratórios (LEI Maior ou Lab. de Microprocessadores) em horários que não estão sendo usados para aulas para testar o funcionamento do projeto nos softwares MikroC PRO for PIC, SimulIDE e no Kit EasyPICv7, desde que agendado previamente com técnicos responsáveis pelos laboratórios, monitores da disciplina e/ou com professor.

### IV – Configuração

- **Botões:** Para utilizar os botões da Porta B do kit EasyPIC é preciso selecionar em **SW** se os pinos usados terão resistores de pull-up (chave para cima) ou pull-down (chave para baixo), e dependendo da escolha conectar o jumper J17 em VCC (se usar pull-down) ou GND (se usar pull-up).

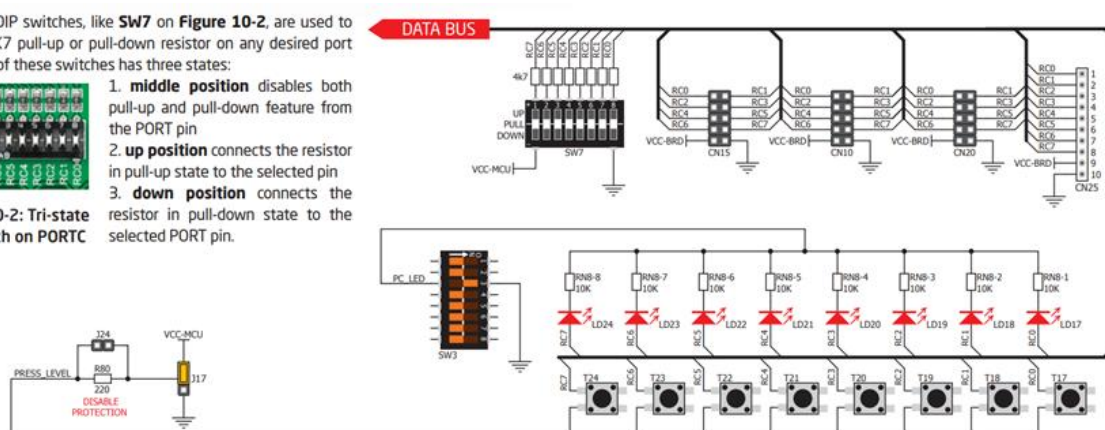
Figura 1 - Manual Kit EasyPIC v7 pág. 22

Tri-state DIP switches, like **SW7** on **Figure 10-2**, are used to enable 4K7 pull-up or pull-down resistor on any desired port pin. Each of these switches has three states:



**Figure 10-2: Tri-state DIP switch on PORTC**

1. **middle position** disables both pull-up and pull-down feature from the PORT pin
2. **up position** connects the resistor in pull-up state to the selected pin
3. **down position** connects the resistor in pull-down state to the selected PORT pin.



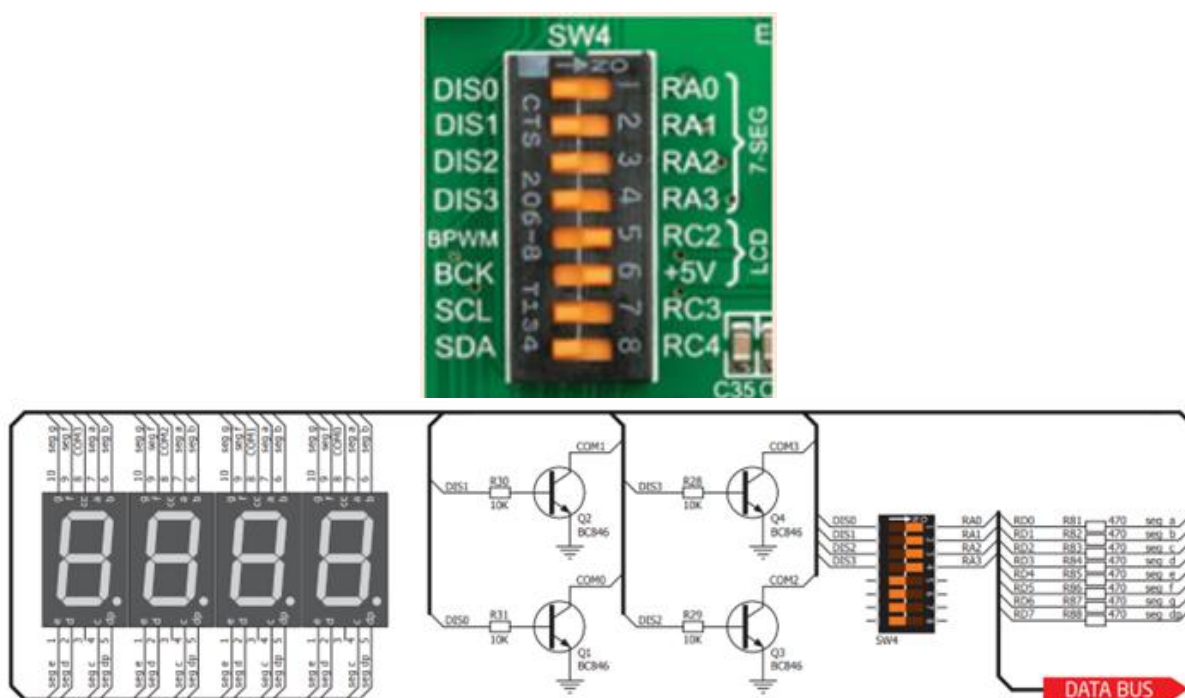
- **Display de 7 segmentos:** O controle deste display é feito pelas quatro primeiras chaves de SW4 (SW4.1, SW4.2, SW4.3 e SW4.4), de forma que cada um serve para ligar um dos displays de 7 segmentos. Os segmentos do display estão conectados à Porta D da seguinte forma:

Tabela 1 - Ligação Porta D e display de 7 segmentos

Segmento	Pino
a	RD0
b	RD1
c	RD2
d	RD3
e	RD4
f	RD5
g	RD6
dp	RD7

O display está conectado na configuração cátodo comum (os segmentos se acendem com nível lógico 1).

Figura 2 - Manual Kit EasyPIC v7 pág. 27



## V - Formato de entrega

- Apresentar em um documento o programa desenvolvido em Linguagem C devidamente comentado.
- Apresentar uma breve discussão textual explicando os conceitos envolvidos nesta prática, os resultados obtidos e como é feita a configuração dos principais blocos e funções do programa desenvolvido, tais como timers, interrupções, acionamento do display de 7 segmentos etc. (no máximo 2 páginas de texto considerando um documento em PDF, sem considerar o código fonte do programa ou imagens).
- Nas discussões descritas acima, compare o projeto atual com o projeto anterior (já que se trata da mesma aplicação), segundo o seu ponto de vista, comentando sobre as vantagens e desvantagens deste programa em linguagem C usando PIC18F em relação ao programa em Assembly usando 8051 do projeto anterior.
- Para complementar a explicação textual e o programa apresentado, apresentar o diagrama (microcontrolador e circuito montado no SimulIDE).
- Apresentar as partes solicitadas acima em único arquivo PDF.
  - Ou, caso preferir, em arquivo README.md do GitHub (neste caso enviar na tarefa o link para repositório do GitHub em que conta tal arquivo - pode ser enviado um arquivo txt contendo salvo este link - OBS. Não enviar diretamente o arquivo README.md na tarefa)
- Um vídeo com gravação da tela do computador demonstrando o funcionamento do programa no SimulIDE poderá substituir o relatório solicitado acima. O vídeo deve exibir a simulação em execução e comprovar o uso de temporizadores e interrupções. Para isso, é necessário mostrar o código em linguagem C no compilador, destacando as partes relacionadas a temporizadores e interrupções, além de realizar a compilação e o carregamento do firmware no SimulIDE em tempo real. O aluno(a) deve explicar o funcionamento do programa desenvolvido (que irá substituir do relatório que solicita a apresentação do programa comentado), abordando os pontos principais do código. A

comparação do projeto atual com o projeto anterior também deve ser comentada brevemente no vídeo, destacando as diferenças. O vídeo deve ser sucinto e direto. Caso opte pelo vídeo, enviar link de acesso ao conteúdo ou realizar o upload do arquivo na tarefa.

- Também deve-se enviar na tarefa o código fonte desenvolvido no mikroC PRO for PIC (“**arquivo.c**”), o “**arquivo.hex**”, bem como o arquivo de simulação “**arquivo.simu**” correspondente a montagem e simulação funcional (com código .hex testado no projeto) realizada no Simul IDE.
- Portanto, a **entrega consistirá no envio de 4 arquivos** (**‘documentação’ (relatório ou vídeo), ‘arquivo.c’, ‘arquivo.hex’, e ‘arquivo.simu’**). Sendo assim:
  - Não enviar os 4 arquivos acima em uma pasta compactada (arquivo zip, rar etc.).
  - Não enviar a pasta com todos os arquivos do projeto que são gerados na compilação do programa no MikroC PRO for PIC
  - Identificar o documento devidamente com nome e NUSP do grupo.
  - O atendimento ao formato de entrega solicitado é um item que será avaliado.
- Em razão do número limitado de kits, a atividade poderá ser feita em grupos de 3 ou 4 integrantes (basta que apenas uma pessoa efetue a entrega pelo grupo).
- Realizar o upload dos arquivos na respectiva tarefa atribuída no e-Disciplinas até a data especificada.
- Qualquer dúvida sobre o formato de envio ou sobre a implementação da atividade prática, entrar em contato com o professor ou com o monitor.

## VI - Critérios de avaliação

Item	Pontuação
Entrega no formato solicitado	<b>1</b>
Boas práticas de programação: programa organizado e com as linhas de código devidamente comentadas nas partes importantes; explicação e discussão textual sobre o programa, suportada com apresentação das partes solicitadas na documentação (ou vídeo)	<b>2</b>
Correção lógica do programa: atendimento ao enunciado e uso dos recursos solicitados, como interfaces I/O (botões, display de 7 segmentos), utilização de interrupções e de temporizadores, projeto e simulação realizada no Simul IDE etc.	<b>7</b>

**OBS.:** apesar da pontuação de correção lógica, não será considerado como entrega somente o envio código fonte, sem a documentação no formato solicitado.

