

SEL0433
APLICAÇÃO DE MICROPROCESSADORES



Parte 2 – Microcontroladores PIC e Programação em Linguagem C
Atividade Semanal 7

Formato de entrega:

- Atividade individual.
- 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. 6 – Programação em Linguagem C para Microcontroladores PIC” (disponível no e-Disciplinas).
- **Objetivos da aula:** introdução a programação em linguagem C para PIC; programação da Família PIC18F e do modelo PIC18F4550 utilizando o compilador MikroC PRO for PIC, Kit EasyPIC e SimulIDE
- Página do compilador MikroC PRO for PIC (com link para download da versão demo)
- Tutorial de uso do compilador
- SimulIDE: página do simulador; link para download (utilizar a versão 0.4.15-SR10)
- Programas de exemplo utilizados na aula (**disponível no e-Disciplinas**).

Durante a aula

- Fazer o download da pasta com exemplos de códigos em linguagem C;
- Abrir o software **MikroC PRO for PIC** por meio do atalho na área de trabalho no PC do laboratório;
- Verificar se existem projetos abertos em “*Project*” no menu principal do software (se a opção “*Close Project*” estiver habilitada, clique sobre ela para fechar projetos abertos);
- Na opção “*Open Project*”, abrir o **Exemplo_1_tecla_LED** (arquivo na extensão “.mcppi”)
- 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).

- Compilar e gravar o programa do **Exemplo_1_tecle_LED** no kit e verificar seu funcionamento (pressionar botão RB0 e verificar LED RD0). Se atentar as configurações: manter RB0 em Pull-Up e o jumper J17 em GND, verificar se o PORTD está habilitado por meio dos DIP switches SW3. Para mais detalhes sobre as ligações consultar o Manual do Kit EasyPICv7 e seu diagrama esquemático.
- Revisar conceitos por meio dos códigos disponibilizados para download na pasta “Baseline” no que se refere à protótipo de funções, arquivo header “.h”, arquivo source “.c”, arquivo “.hex”, arquivo “.asm”, biblioteca padrão C, “linkagem”, compilação, diretivas do compilador, encapsulamento de funções para evitar duplicação de variáveis, escopo de variáveis (local/global), tipo de dados e suas implicações (char, int, float).
- Explorar a criação e configuração de um novo projeto no MikroC PRO for PIC, com especial atenção voltada para: seleção do “device”, configuração do clock, *configuration bits* (tipo de cristal, fonte de clock, resets, desabilita comparadores, watchdog timer). Digitar um programa para realizar alguns testes.
- Explorar recursos do MikroC PRO for PIC: *Tools – Start Debugger* (inicia depuração de um programa compilado – verificar depuração em *Watch Values* – add variáveis de interesse, executar linha por linha com F7, inserir breakpoints com duplo clique na linha. Explorar os exemplos 1, 2 e 3.
- Ilustrar no **SimulIDE** (para executá-lo no PC do laboratório: “C – Arquivos de Programas – SimulIDE – Bin – Simulide.exe”) a conexão de um botão no pino B0 do microcontrolador PIC18F4550 na configuração *Pull-down* (externo), o qual ao ser pressionado deverá mudar o estado de um LED conectado ao pino D0 (não esquecer do resistor limitador para o LED). Simule o circuito carregando o firmware (arquivo *hex*) gerado na compilação do programa em Linguagem C que atende essa lógica, a qual foi realizada anteriormente no software no software MikroC PRO for PIC. Configurar o valor do resistor e a frequência do clock do microcontrolador.

Pós-aula

- Altere a lógica do programa do Exemplo 1 para piscar o LED a cada 500 ms (usando a função *delay*) enquanto o botão se manter pressionado. Ao soltar o botão, o LED deve ser desligado. Realize a simulação no SimulIDE e utilize um osciloscópio digital (recurso disponível no simulador) para verificar o sinal de saída. Conectar o osciloscópio no pino correspondente ao LED no microcontrolador PIC.
- Conforme exemplo demonstrado em aula (Exemplo 2), implementar o algoritmo utilizado para tratar o efeito *bounce* presente no programa do Exemplo 1. Compilar o programa no MikroC PRO for PIC e implementar o circuito no Simul IDE carregando o firmware (arquivo *hex* gerado na compilação). Ajustar o clock do microcontrolador

PIC18F4550 para 8 MHz e o montar o botão na configuração pull-up (ajustar o valor do resistor de pull-up para 10 k Ω) no SimulIDE.

- Implemente no SimulIDE o programa no **Exemplo 3 – Display de 7 Segmentos**. Para tanto, realize as ligações de um display de 7 segmentos disponível no simulador no PORTD do microcontrolador. Ajustar o clock do microcontrolador PIC18F4550 para 8 MHz e o ligar o botão na configuração pull-up (ajustar o valor do resistor de pull-up para 10 k Ω) no SimulIDE.

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 diretamente na resposta dos exercícios acima os programas em linguagem C e um print do circuito montado comprovando a simulação realizada no SimulIDE.**