

SEL0433/SEL0336/SEL0614

APLICAÇÃO DE MICROPROCESSADORES



Roteiros de Aulas, Atividades e Projetos

Parte 2 – Microcontroladores PIC e Programação em Linguagem C

Motivação e objetivos: será abordado a programação de periféricos em microcontroladores de 8 bits da família PIC18F (instruções otimizadas RISC/Harvard para programação em linguagem C), tais como: temporizadores, para a contagem de tempo; interrupções para otimização de códigos; e conversor analógico-digital para leituras analógicas de variáveis de interesse em problemas reais.

Instruções e datas para entrega de atividades/projetos: todas as atividades podem ser realizadas em grupos de três pessoas. Não é permitido realizá-las de forma individual. Instruções específicas e datas de entrega estarão contidas em documento específico na tarefa correspondente atribuída no e-Disciplinas (Moodle).

Material de apoio: além dos links disponibilizados diretamente no presente documento, o material principal estará também disponível no e-Disciplinas.

Instruções para as aulas: As ferramentas computacionais utilizadas no curso estão disponíveis nos computadores do laboratório. Durante o horário da aula, os computadores são de uso exclusivo para realização das atividades pertencentes à disciplina e não devem ser utilizados para outros fins. **Importante:** ao final da aula ou ao término do uso dos computadores, não esqueça de fazer logout de contas pessoais que eventualmente tenha acessado durante seu uso (email, e-Disciplinas etc.), desligar os computadores e organizar mesas e cadeiras que utilizou.

Recomendações importantes: recomenda-se que a utilização do kit EasyPICv7 durante as aulas seja revezada entre as três pessoas, possibilitando a cada aluno(a) a oportunidade de reproduzir as atividades, tais como manipular a plataforma e realizar configurações, gravar o programa no kit etc., em razão da limitação de 13 unidades. Recomenda-se cuidado e atenção ao manipular as placas, pois seus componentes são extremamente sensíveis a danos. Não alterar configurações de chaves e jumpers sem uma necessidade evidente ou sem conhecimento pleno das ligações envolvidas na placa, evitando afetar o funcionamento da plataforma (exceto em situações específicas, como correções necessárias ou modificações solicitadas pelo professor). Em caso de dúvidas, sempre perguntar antes de prosseguir com alguma ação. Mantendo esse cuidado, será possível que outras turmas também utilizem os kits em boas condições. Durante as aulas, o monitor estará à disposição para ajudar a resolver problemas de uso do kit. Ao término da aula, procure devolver o kit da mesma forma que o recebeu: guardado na embalagem junto ao cabo USB.

Monitoria: será possível agendar esclarecimentos de dúvidas e atendimento para auxílio nas atividades/projetos em outros horários, com o professor e com monitores. Da mesma forma, podem utilizar os laboratórios (LEI Maior ou Lab. de Microprocessadores) em horários que não estão sendo usados para aulas para realizar atividades da disciplina (caso necessitem de computadores, uso das ferramentas computacionais, kits etc.), preferencialmente agendando previamente com técnicos responsáveis, monitores e/ou com professor.

Roteiros

Aula 6	4
Pré-aula	4
Pós-aula	4
Aula 7	5
Pré-aula	5
Pós-aula	6
Aula 8	7
Pré-aula	7
Pós-aula	7
Projeto 2 - Cronômetro Digital com Timer e Interrupções	8
Aula 9	13
Pré-aula	13
Pós-Aula	13
Projeto 3 – Termômetro Digital com LM35 e Conversor A/D.....	14

Aula 6

Pré-aula

- **Material de aula:** “[Cap. 5 – Microcontroladores PIC e Kit EasyPIC](#)” (**disponível no e-Disciplinas**).
- **Objetivos da aula:** conhecer famílias de microcontroladores PIC, especialmente à linha de 8 bits; estudar arquitetura, organização de memória, e recursos da linha PIC18F e modelo usado no curso: PIC18F4550.
- [Página da microchip](#); [Microchip 8-bit PIC Microcontrollers](#)
- [Datasheet](#) e [página](#) do microcontrolador PIC18F4550
- **Kit EasyPIC v7:** [manual](#) do kit; [diagrama](#) esquemático; [página do fabricante](#)

Pós-aula

Pontos importantes:

- Considerações sobre família PIC: arquitetura Harvard, set de instruções RISC (maior eficiência e economia); compatibilidade de pinos entre modelos; otimização para programação em linguagem C; ciclo de máquina e instruções no PIC ($4T - F_{osc}/4$); linha de 8 bits (baseline, mid-range e high performance); memória flash x memória EEPROM;
- PIC18F: instruções de 16 bits, contador de programa/pilha de 21 bits para endereçamento (16 bits da instrução + 5 bits do Stack pointer para pilha de 32 níveis de endereços); áreas da memória de programa e memória de dados; memória flash até 2 MBytes ou 1K x 16 words; memória SRAM até 4 KB – 16 bancos de 256 bytes e barramento de endereço de 12 bits (sendo 8 bits para operando e 4 para selecionar os 16 bancos de registradores)
- Pinagem do modelo PIC18F4550; identificação das portas I/O (PORTA, B, C, D, E), pinos VDD e VSS, MCLR (master clear), USB, CLK, canais analógicos ANx, etc.
- Programação de entradas e saídas: registradores PORT, TRIS, LAT, ADCON; formas de reset e tipo de cristais, método de gravação de programas ISP.
- Kit EasyPIC v7 Mikroe: compatibilidade para microcontroladores PIC, alimentação, gravador, cristal oscilador, periféricos de I/O e recursos para projetos (displays, buzzer, sensores de temperatura, switches e bancos de LEDs, reset, potenciômetros, comunicação USB etc.).

Aula 7

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

Atividade 1

- 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 “.mcppl”)
- 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_tecla_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.

Atividade 2

- Ilustrar no **SimulIDE** (abrir em “*C – Arquivos de Programas – SimulIDE – Bin – Simulide.exe*”) a conexão de um botão no pino B0 do microcontrolador PIC18F4550 na configuração *Pull-Up* (externo), o qual ao ser pressionado deverá mudar o estado de um LED conectado ao pino D0, com base no exemplo da Atividade 1. Simule o circuito carregando o arquivo *hex* gerado na compilação do programa em Linguagem C que atende essa lógica, realizada no software no software MikroC PRO for PIC. Configurar o valor do resistor *Pull-Up* para 10 kΩ e a frequência do clock do microcontrolador para 8 MHz.

Atividade 3

- 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 de código, verificar ciclos de máquina na opção *View – Debug Windows – Stopwatch*); *View – assembly; listing, statistics* (ver estatísticas de um programa compilado).

Atividade 4

- Abrir o programa do **Exemplo 2** no MikroC, fechando outros projetos abertos, compilar e gravar no KitEasyPIC. Verificar o algoritmo utilizado para tratar o efeito *bounce* presente no programa do Exemplo 1.
- Abrir o programa do **Exemplo 3** no MikroC, fechando outros projetos abertos, compilar e gravar no KitEasyPIC. Habilitar o Display de 7 segmentos do kit por meio dos DIP switch SW4. Verificar a lógica de varredura do Display de 7 Segmentos conectado ao PORTD do microcontrolador PIC18F4450 e as ligações deste módulo no Manual do Kit EasyPICv7.

Pós-aula

Exercícios práticos para revisão

(não é necessário a entrega)

- 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.
- Faça um programa em linguagem C para o microcontrolador PIC18F4550 utilizando o compilador MikroC PRO for PIC que leia a porta B e escreva o resultado na porta D com um intervalo de 1 segundo entre leitura e escrita. Certifique-se de pré-definir um valor para a porta B para que seja transferido para porta D.
- 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.

Aula 8

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

- Abrir no MikroC PRO for PIC o programa do **Exemplo 5**. Compilar e gravar no kit EasyPICv7. Verificar o resultado (um LED deve piscar a cada 1segundo com uso do Timer0 do PIC18F4550)
- Abrir no MikroC PRO for PIC o programa do **Exemplo 8**. Compilar e gravar no kit EasyPICv7. Verificar o resultado (uso de interrupção externa – comparar com o programa do Exemplo 1)
- Abrir no MikroC PRO for PIC o programa do **Exemplo 9**. Compilar e gravar no kit EasyPICv7. Verificar o resultado (uso de interrupção interna – timer0 – comparar com o programa do Exemplo 5).
- Se houver tempo, reproduzir os exemplos acima no SimulIDE, realizando as devidas ligações dos circuitos e configurações necessárias.

Pós-aula

(próxima página)

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 MikroC PRO for PIC, 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.
- **Material relacionado:** indicado nas Aulas 6 e 7;

Esta atividade prática versa sobre o desenvolvimento do mesmo projeto do cronômetro digital. 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).

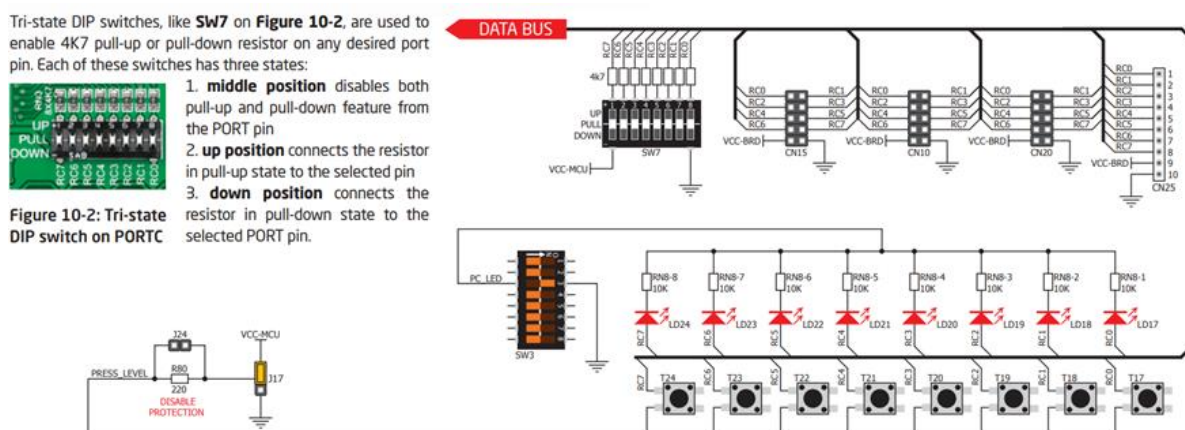
III - Atividade prévia (tarefa):

- Estudar os exemplos de códigos disponibilizados anteriormente para download (**Exemplos 3, 5, 8 e 9**), revisando conceitos e testes realizados durante as **aulas 7 e 8** e implementações dos exemplos nos softwares MikroC PRO for PIC e SimulIDE. Além dos programas disponibilizados, o material principal de consulta está indicado nas aulas 6, 7 e 8.
- 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.
- 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.
- **Contato dos monitores da disciplina:**
 - Pedro Fiorese Machado Netto - pedrofmnetto@usp.br;
 - Johnny Caselato Guimarães - johnny.caselatoguimaraes@usp.br;
 - Marco Tulio Reggiani Cardoso- marcoreggianicard@usp.br;

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



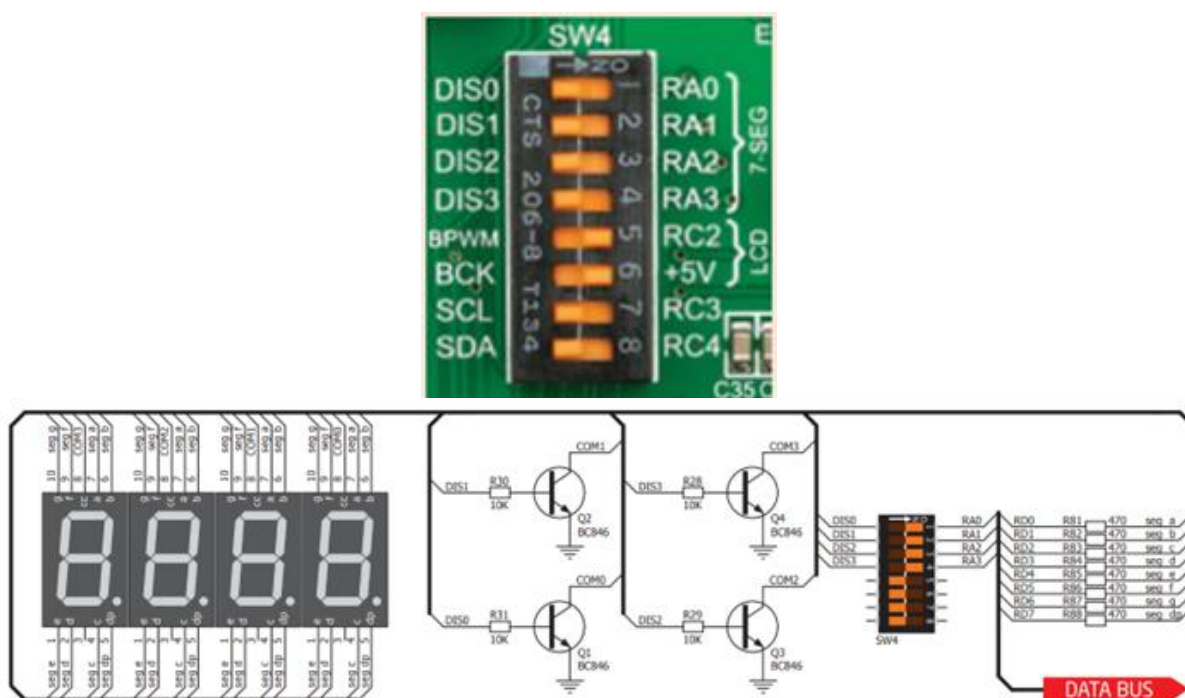
- **Display de 7 segmentos:** O controle deste display é feito pelas quatro primeiras chaves de SW4 (SW4.1, SW4.2, SW4.3 e SW4.4), de forma que cada um serve para ligar um dos display de 7 segmentos. Os segmentos do display estão conectados à Porta D da seguinte forma:

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 1 página 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 PD.
 - 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)
- Além do documento acima referido, enviar também o código fonte desenvolvido no mikroC PRO for PIC (“arquivo.c”) e 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 apenas 4 arquivos avulsos** (“documentação”, “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 de todas as pessoas 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 integrantes (lembrando que não é necessário que cada integrante do grupo faça envio no e-Disciplinas, bastando apenas uma pessoa fazer a entrega pelo grupo).
 - Fazer 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	1
Boas práticas de programação: programa organizado e com as linhas de código devidamente comentadas; explicação e discussão textual sobre o programa, suportada com diagramas solicitados	2
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.	7

OBS.: não será considerado como entrega somente o envio dos arquivos do projeto, sem a documentação contendo discussão e explicação do que o código faz.

Aula 9

Pré-aula

- **Material de aula:** “Cap. 8 – Conversor A/D”
- **Objetivos da aula:** programação do conversor analógico-digital e display LCD com especial atenção para os registradores envolvidos na configuração e nas bibliotecas do compilador que permitem otimização do código.
- Códigos de exemplo - programas em Ling. C usados na aula (**disponível no e-Disciplinas**).

Durante a aula

- Explorar recursos do MikroC PRO for PIC: **biblioteca ADC**, recursos para display LCD (menu *Tools – ascii chart*; *Tools – LCD custom character*; **biblioteca Lcd**)
- Abrir no MikroC PRO for PIC os programas do *Exemplo_10_ADC_LCD* e, posteriormente, do *Exemplo_11_ADC_LCD*. Compilar e gravar sequencialmente os programas no KitEasyPICv7 (gravar e testar primeiramente o Exemplo 10, depois fazer o mesmo para o exemplo 11).
- Os programas acima irão permitir realizar a leitura analógica do valor de tensão de um potenciômetro (Vref. 0-5V usando o Trimpot do Kit EasyPIC v7). Para tanto, conectar o potenciômetro no pino **RA3 ou RA5** (canal analógico AN0 ou AN4- ver Fig. 2 a seguir).
- Ligar o módulo LCD do kit no DIP Switch **SW4**, por meio das chaves **SW4.5** e **SW4.6**. No kit EasyPICv7, o display está em sua configuração de 4 bits, de forma que só são utilizados os quatro bits mais significativos (D4-D7) para a escrita, e os outros quatro bits menos significativos estão ligados ao GND.
- É importante ressaltar que no caso do Kit, o display LCD está conectado ao PORTB. Por essa razão, sugere-se utilizar os pinos do PORTA para testar os exemplos acima (já que os pinos do PORTB estão sendo usados na configuração digital para escrita no display). Essa recomendação é específica para o kit, devido às ligações físicas permanentes da placa nesta configuração. No entanto, ao testar os exemplos no SimulIDE, o display pode ser conectado a qualquer outro PORT, não sendo obrigatório também manter os mesmos pinos para testar o conversor A/D. É fundamental, porém, respeitar os pinos aos quais estão conectados os 13 canais analógicos do microcontrolador.

Pós-Aula

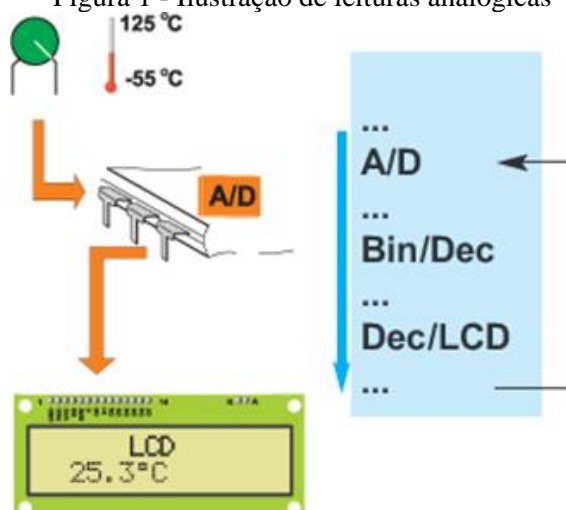
Projeto 3 – Termômetro Digital com LM35 e Conversor A/D

Objetivos

- Familiarização com compiladores e programação básica em linguagem C de registradores SFR, I/O, e periféricos de microcontroladores de 8 bits.
- Utilizar o compilador MikroC PRO for PIC para desenvolver aplicações que lidam com dados analógicos, com foco especialmente voltado para módulo ADC (conversor analógico-digital ou conversor A/D) e interface com display LCD no PIC18F4550.
- Implementar projetos no kit EasyPIC v7 e realizar simulações por meio do software Simul IDE.
- **Material relacionado:** indicado nas Aulas 6, 7 e 8;

A utilização de leituras analógicas possui grande importância para sistemas embarcados, tendo em vista que, em geral, as variáveis de interesse dos problemas reais não possuem discretização (como temperatura, pressão, luz etc.), sendo necessário o uso de conversores A/D para a conversão de sistemas analógicos para discretizados para serem tratados pelo sistema de controle digital. Uma forma de conexão pode ser o uso de módulos ADC ou, por exemplo, como visto nesta prática, o uso de microcontroladores com módulos ADC internos capazes de cumprir esse propósito sem a necessidade de componentes externos. O módulo ADC presente internamente nos microcontroladores é configurável por meio de registradores e interfaces de programação em alto nível (neste caso, por exemplo, o uso do PIC18F e a programação em linguagem C por meio do compilador MikroC PRO for PIC já conta com bibliotecas e funções pré-configuradas para uso do módulo ADC). Outra vantagem é a flexibilidade em termos de tensões de referências ajustáveis e a precisão do conversor que irá proporcionar sensibilidade adequada conforme a aplicação (no caso do PIC18F, a resolução é de 10 bits)

Figura 1 - Ilustração de leituras analógicas



I - Roteiro

Parte 1 (familiarização com conversor A/D e display LCD – Tarefa)

- Estudar o programa do Exemplo 11 disponibilizado para download (“*Exemplo_11_ADC_LCD*”), bibliotecas ADC e Lcd do compilador MikroC PRO for PIC, revisando conceitos e testes realizados durante a aula 9, bem como material indicado nesta aula.
- Testar o projeto do **Exemplo 11** no **SimulIDE**: montar o circuito com microcontrolador PIC18F4550, potenciômetro de 0-5V, display LCD etc; carregar o firmware “.hex” do programa compilado no MikroC PRO for PIC; simular o projeto. A frequência do clock deve ser de **8 MHz** (cristal externo HS presente no kit).
- A simulação irá permitir realizar a leitura analógica do valor de tensão de um potenciômetro (V_{ref} . 0-5V). Para tanto, conectar o potenciômetro no pino **RA3 ou RA5** (canal analógico AN0 ou AN4- ver Fig. 2 a seguir). Ligar o módulo LCD no SimulIDE com base na configuração presente no kit EasyPICv7, ou seja, o display está em sua configuração de 4 bits, de forma que só são utilizados os quatro bits mais significativos (D4-D7) para a escrita, e os outros quatro bits menos significativos estão ligados ao GND, conforme Fig. 2.

Figura 2 - Manual Kit EasyPIC v7 pág. 24

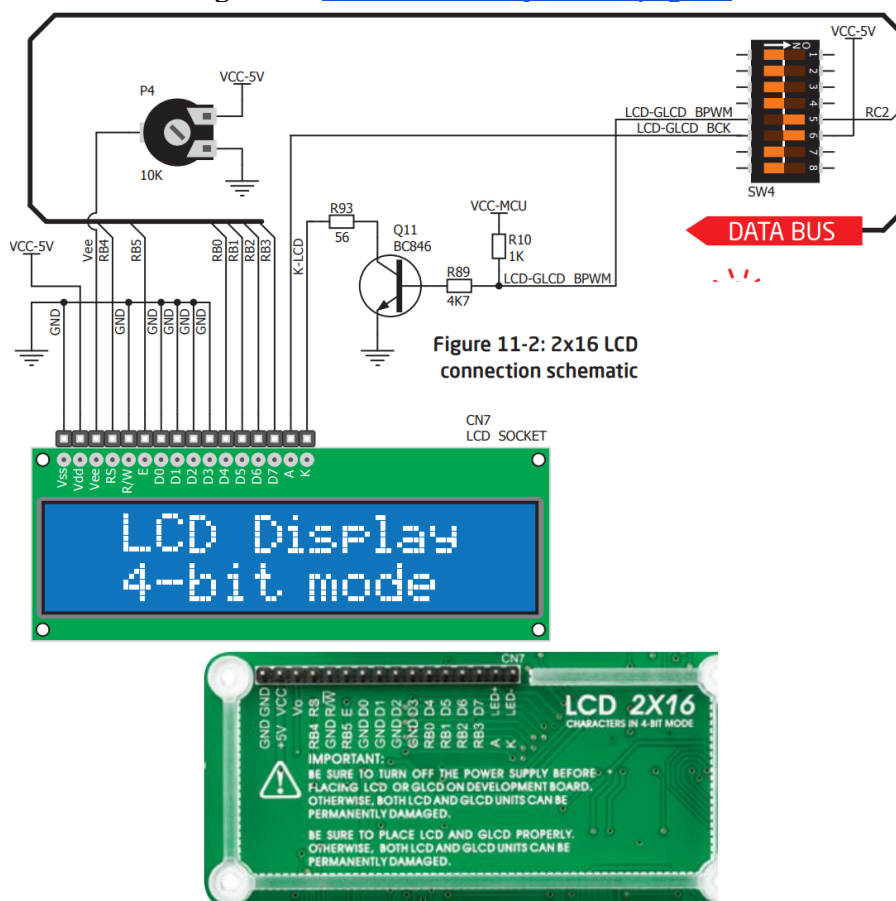
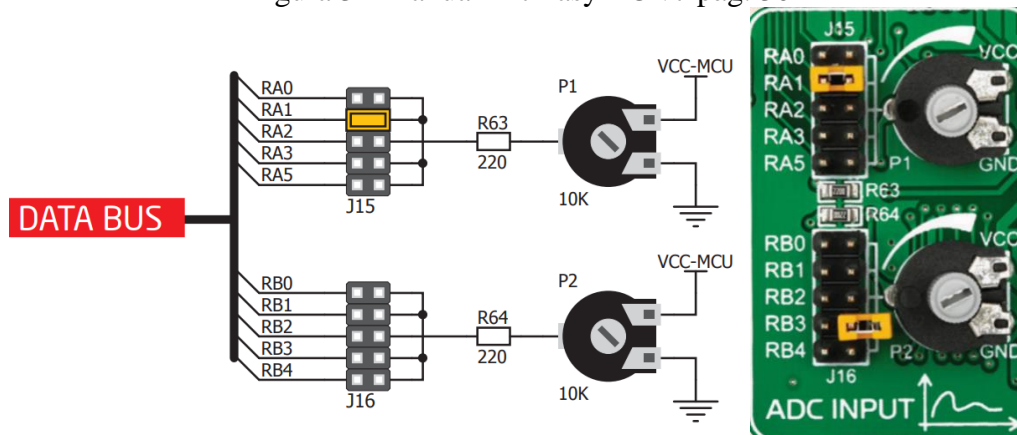


Tabela 1 - Ligação das Portas e o display LCD

LCD	Pino
D4	RB0
D5	RB1
D6	RB2
D7	RB3
RS	RB4
E	RB5

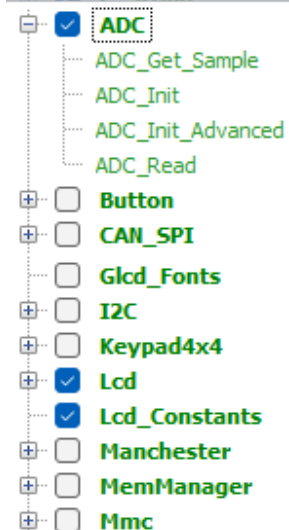
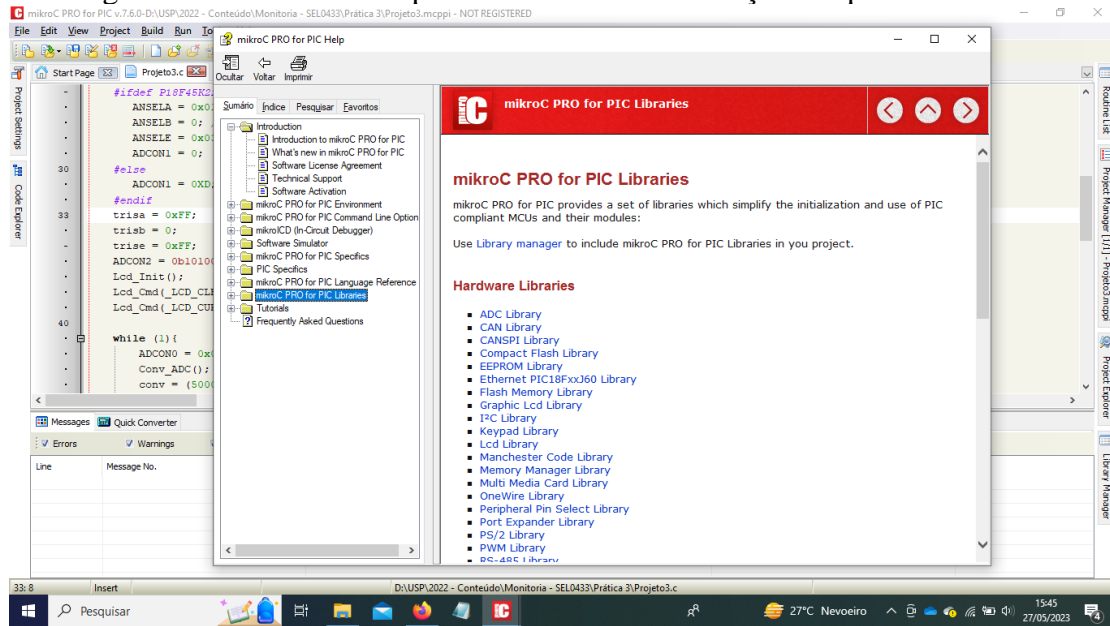
- **Potenciômetro:** o kit possui dois potenciômetros que podem ser ligados nas entradas do conversor A/D, o potenciômetro **P1** pode ser conectado em qualquer um dos pinos de **RA0** à **RA3** e o pino **RA5**, sua seleção é feita pelo jumper **J15**, e o potenciômetro **P2** pode ser conectado em qualquer um dos pinos de **RB0** à **RB4** e sua seleção é feita pelo jumper **J16**. Para esta prática **P1** será conectado ao pino **RA0** (ver [Manual Kit EasyPIC v7](#) para mais detalhes), usando somente os canais analógicos do **PORTA**, pois o **PORTB** deverá ser definido como digital, já que será utilizado para o Display LCD.

Figura 3 - Manual Kit EasyPIC v7 pág. 30



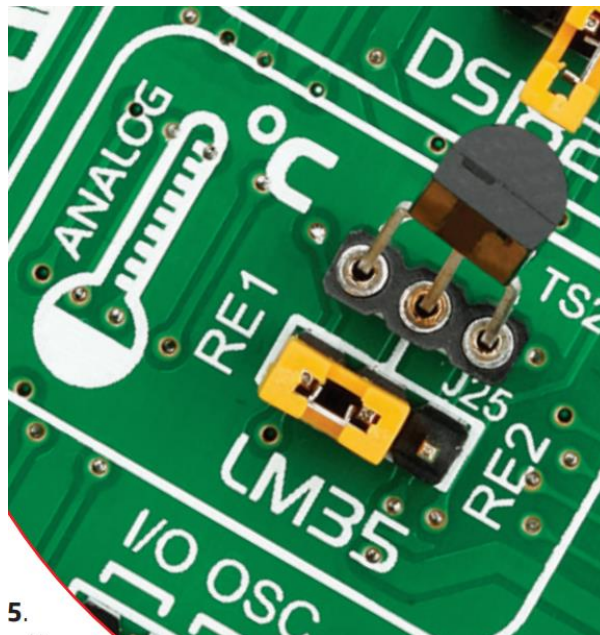
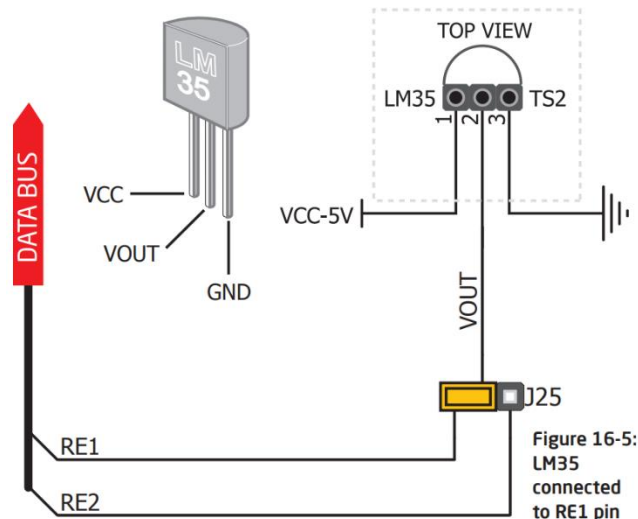
- **Bibliotecas:** Para facilitar a utilização do conversor A/D e do display LCD é possível fazer uso das bibliotecas ADC, e Lcd disponíveis em **Library Manager** (acessar no menu *View > Library Manager*). No Help do **MikroC PRO for PIC Libraries** é possível ver as bibliotecas disponíveis e suas funções.

Figura 4- Terminal de Help com as bibliotecas e funções disponíveis



- **Exemplo de aplicação prática com LM35:** sensor de temperatura que apresenta uma tensão analógica com sensibilidade de +10.0 mV/°C (range: +2° C a +150°C). Neste kit é possível conectar este sensor no PIC nos pinos RE1 e RE2. Essa escolha é feita pelo jumper **J25**. Para esta prática ele será conectado no pino **RE1** (ver [Manual Kit EasyPIC v7](#) para mais detalhes).

Figura 5 - Manual Kit EasyPIC v7 pág. 29



[Datasheet LM35](#)

Parte 2 – Projeto (Tarefa)

- Modifique o programa do **Exemplo 11** de forma que ele simule a leitura de temperatura por um sensor LM35 na faixa de 0 a 100 °C e mostre o resultado em um display LCD.
- Implementar o projeto no SimulIDE. Como o simulador não dispõe deste sensor, assim como no Exemplo 11, o projeto continuará utilizando um potenciômetro de forma que a variação de tensão no potenciômetro simule a variação de temperatura. A diferença principal é que a agora a variação no potenciômetro deverá exibir valores de 0 a 100 no display LCD.
- Outra diferença em relação ao Exemplo 11, é que agora deve ser utilizado, no SimulIDE, uma alimentação externa de 1V conectada nos canais analógicos AN2 e

AN3 (ver config. do registrador ADCON1 no datasheet do PIC18F4550) para simular a variação de 0 a 100 °C do LM35 (Vref. de 0 a 1V irá se adequar melhor à sensibilidade do LM35 do que usar Vref. 0-5V). É importante lembrar que a configuração acima é feita no registrador ADCON1, ou seja, é possível escolher entre alimentação interna (a qual usa a própria alimentação VDD e Vss do microcontrolador, neste caso de 0 a 5V, como foi o caso do **Exemplo 11**), ou alimentação externa (a qual deverá ser feita agora), na qual o microcontrolador irá ler uma alimentação externa conectada aos pinos AN2 e AN3 para ajustar Vref do conversor A/D.

- Após a leitura, os valores de temperatura em °C (0 a 100) devem ser escritos no display LCD (Fig. 2). Considerar pelo menos três algarismos significativos da seguinte forma: “XX.X °C”
- A frequência do clock do PIC deve ser de 8 MHz (cristal externo HS).
- Utilizar as funções da biblioteca do ADC e LCD, seguindo os moldes do Exemplo 11. Consultar material de apoio (Aula 9), manual do kit e datasheet do microcontrolador PIC18F4550.
- Não será necessário testar o projeto no Kit EasyPICv7 em razão de não termos no laboratório uma fonte alimentação externa de 1V para conectar ao kit. Portanto, para entrega, será considerado apenas o programa desenvolvido no MikroC PRO for PIC e o projeto testado no SimulIDE (apresentar circuito montado e arquivo de simulação).
- É 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.
- **Contato dos monitores da disciplina:**
 - Pedro Fiorese Machado Netto - pedrofmnetto@usp.br;
 - Johnny Caselato Guimarães - johnny.caselatoguimaraes@usp.br;
 - Marco Tulio Reggiani Cardoso- marcoreggianicard@usp.br;

II - Formato de entrega (entregar somente a Parte 2)

- Apresentar em um documento uma breve discussão textual (no máximo 1 página de texto) explicando de forma objetiva os resultados obtidos, bem como os principais conceitos envolvidos na atividade (configuração de funções do programa desenvolvido, tais como conversor A/D, configuração dos registradores para as leituras, ajuste de Vref. e uso das funções da biblioteca, acionamento do display LCD etc).

- Para complementar a explicação textual, apresentar a imagem do circuito montado no SimulIDE.
- Enviar também na tarefa o arquivo de simulação “**arquivo.simu**” correspondente a montagem e simulação funcional no SimulIDE, e os códigos **.hex** e em “**c**” do programa desenvolvido e compilado no MikroC PRO for PIC. A entrega consistirá, portanto, no envio de 4 arquivos avulsos.
- Fazer o upload dos arquivos na respectiva tarefa atribuída no e-Disciplinas até a data especificada. A atividade poderá ser feita em grupos de 3 integrantes.
- Qualquer dúvida sobre o formato de envio ou sobre a implementação da atividade prática, entrar em contato com o professor ou com o monitor.

Critérios de avaliação

Item	Pontuação
<u>Entrega no formato correto</u>	1
<u>Programa com as linhas de código devidamente comentadas</u> <u>Explicação e discussão textual sobre o programa, suportada com</u> <u>diagramas</u>	2
<u>Correção lógica do programa: atendimento ao enunciado e uso</u> <u>dos recursos solicitados, como interfaces I/O (botões, display</u> <u>LCD), módulo ADC, alimentação externa aos canais analógicos,</u> <u>potenciômetro, leitura de temperatura de 0 a 100° etc.</u>	7