

SEL0433 APLICAÇÃO DE MICROPROCESSADORES



Parte 2 – Microcontroladores PIC e Programação em Linguagem C

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 neste projeto, 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 PIC18F4550, a resolução é de 10 bits).

Figura 1 - Ilustração de leituras analógicas

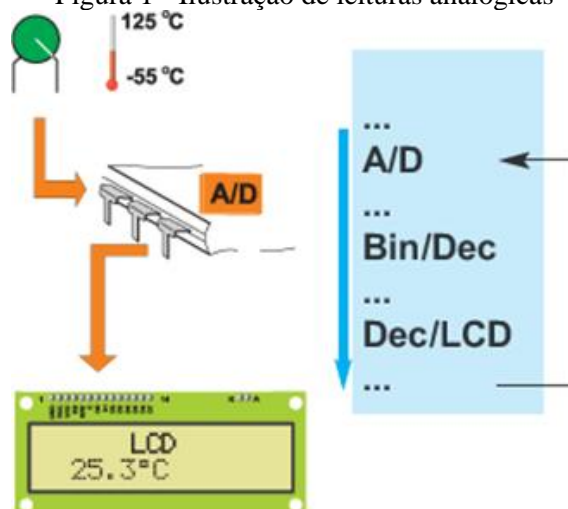
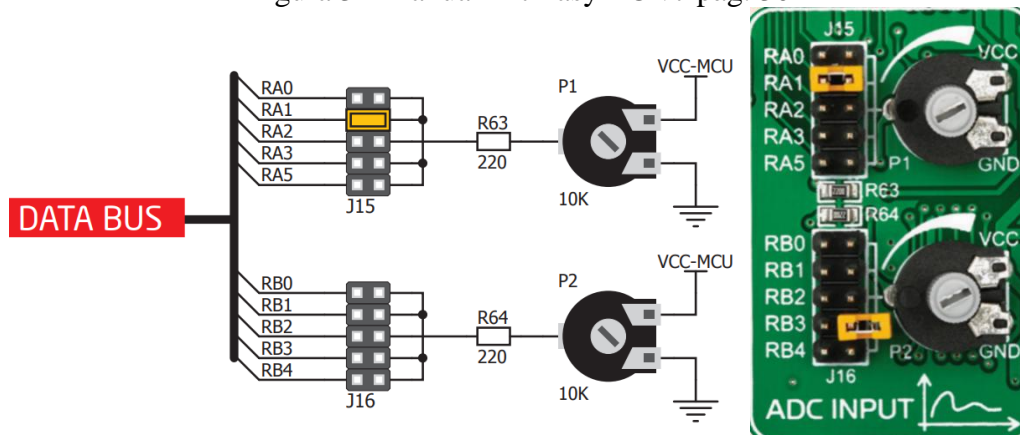


Tabela 1 - Ligação das Portas e o display LCD

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

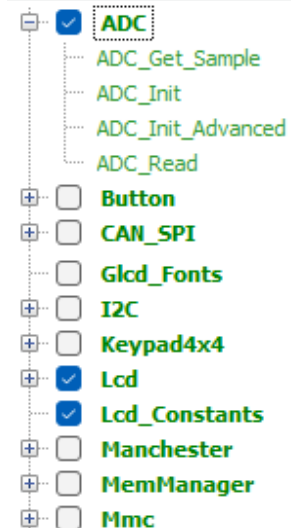
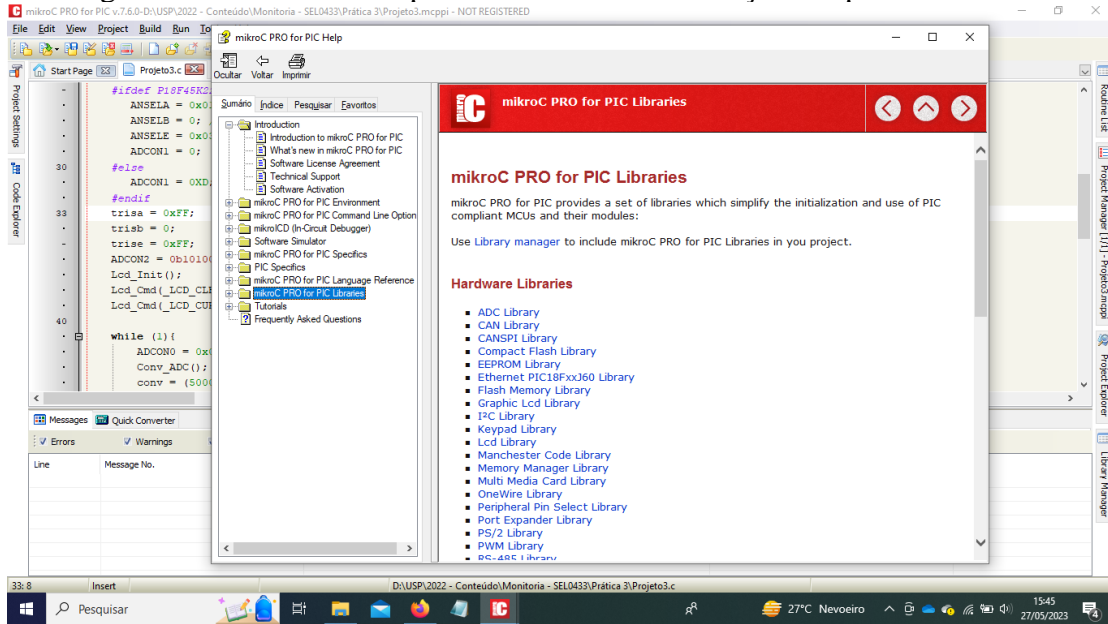
- **Potenciômetro:** o kit possui dois potenciômetros que podem ser ligados nas entradas do conversor A/D, o potenciômetro **P1** pode ser conectado em qualquer um dos pinos de **RA0** à **RA3** e o pino **RA5**, sua seleção é feita pelo jumper **J15**, e o potenciômetro **P2** pode ser conectado em 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



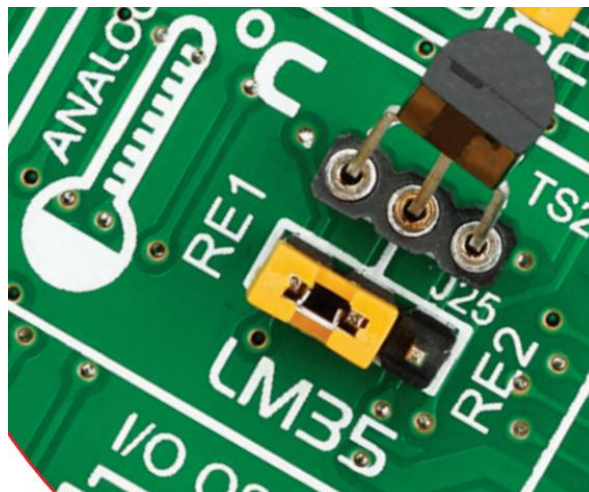
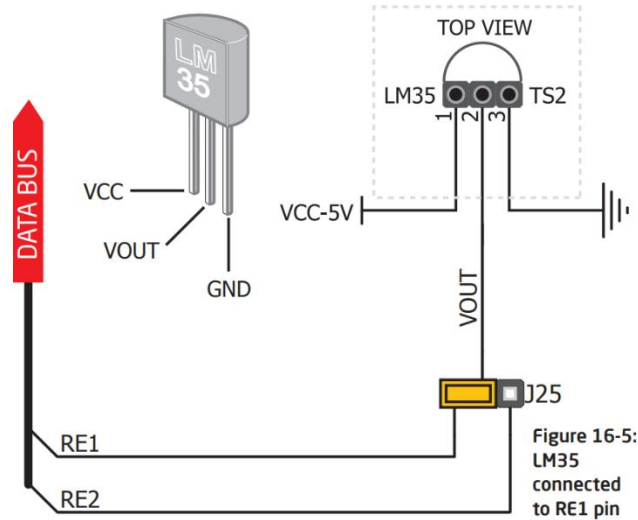
- **Bibliotecas:** Para facilitar a utilização do conversor A/D e do display LCD é possível fazer uso das bibliotecas ADC, e Lcd disponíveis em **Library Manager** (acessar no menu *View > Library Manager*). No **Help do MikroC PRO for PIC Libraries** é possível ver as bibliotecas disponíveis e suas 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 $+10.0 \text{ mV}/^{\circ}\text{C}$ (range: $+2^{\circ} \text{ C}$ a $+150^{\circ}\text{C}$). 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)

- Com base nas implementações da atividade semanal, elaborar um programa em linguagem C capaz de realizar a leitura analógica de temperatura por meio de um sensor LM35, na faixa de 0 a 100 °C, e mostrar o resultado em um display LCD. Utilizar as bibliotecas do módulo ADC do compilador MikroC PRO for PIC.
- Implementar o projeto no SimulIDE. Como o simulador não dispõe deste sensor, assim como nos exemplos abordados, 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 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 dos exemplos anteriores), 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 com essas tensões externas.

- 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”**. Formatar os valores sem usar dados do tipo *float* em razão deste formato bastante de memória de dados.
- A frequência do clock do PIC deve ser de 8 MHz (cristal externo HS).
- Utilizar as funções da biblioteca do ADC e LCD do compilador MikroC PRO for PIC. Consultar material de apoio, 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.

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.
- 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 mostrar o código em linguagem C no compilador, destacando as partes relacionadas as configurações do ADC e formatação dos valores (deve demonstrar 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. 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.

- 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.
- Realizar o upload dos arquivos na respectiva tarefa atribuída no e-Disciplinas até a data especificada. A atividade poderá ser feita em grupos.
- Qualquer dúvida sobre o formato de envio ou sobre a implementação da atividade prática, entrar em contato com o professor ou com o monitor.

III- 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 (ou vídeo).</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

IV- Solução para Erros no Projeto

Com relação às tensões de referência externas para o conversor A/D do microcontrolador PIC18F4550, seguem algumas considerações importantes:

- Ao utilizar a biblioteca ADC no software MikroC PRO For PIC ocorre o seguinte erro (do próprio compilador): a função *ADC_Init* coloca os bits 4 e 5 do registrador ADCON1 em 0 (por padrão). No entanto, estes dois bits devem estar em 1 para permitir ajustar Vref do conversor A/D como sendo a alimentação externa conectada aos pinos A2 e A3 do PIC (0-1V). Essa configuração estava causando erros ao testar o circuito no SimulIDE, pois os bits 4 e 5 sempre estavam em 0 (em zero, o microcontrolador utiliza as tensões de alimentação, isto é, tensões externas como referência Vref do A/D , algo que neste caso não queremos).
- Da mesma forma, a função *ADC_Read* não funciona corretamente para tensões de referência externas. Deve-se usar a função *ADC_Get_Sample(numero_canal)*:

- por exemplo: `ADC_Get_Sample(4);` para o canal AN4.
- Para resolver esses problemas, o módulo ADC deve ser inicializado primeiro e, em linhas posteriores, definir o registrador `ADCON1`, pois notamos que mesmo usando a função `ADC_Get_Sample`, o problema continua se não seguir a sequência a seguir:

```
void main() {

    ADC_Init(); // inicializa o módulo ADC (deve vir antes)

    ADCON1 = 0b00110000; // deve vir depois da linha anterior

    while(1)

    {
        ...
        Valor_ADC = ADC_Get_Sample(4); // Usar essa função ao invés de ADC_Read
        ....
    }
}
```

- Lembrando que na configuração acima, o potenciômetro deve ser ajustado para variar de 0 a 1V, simulando um LM35, que possui sensibilidade de 10 mV/°C e faz leitura de 0 a 100°C (por isso adota-se V_{ref} de 0 a 1V no conversor A/D).

V - Desafio (opcional – valendo 1 ponto a mais) - Projeto de um Voltímetro Digital com Microcontrolador PIC

Objetivo: desenvolver um voltímetro digital utilizando um microcontrolador PIC que leia uma tensão de entrada de 0 a 20 V e exiba o valor correspondente em um display LCD. Deve-se utilizar um microcontrolador PIC da família de 8 bits (pode ser PIC18F ou 16F, sendo necessário apenas que tenha número de pinos suficientes para acionar um display LCD e receber um sinal de entrada analógico, além das demais ligações: reset, clock etc.). O modelo escolhido, portanto, deve possuir conversor A/D interno com pelo menos um canal analógico disponível para uso. Será por meio deste canal analógico que o microcontrolador irá realizar a leitura da entrada referente ao sinal de tensão a ser medido como voltímetro. É de extrema importância enfatizar que o PIC somente pode lidar com sinal de entrada de 0 a 5V (caso o conversor A/D seja de 10 bits, esse valor será quantizado em 1024 níveis). Portanto, o sinal de entrada de 0 a 20V deve primeiro passar por um divisor de tensão, que o reduzirá para 0 a 5V. Deve-se também utilizar um amplificador operacional com realimentação para casamento de impedâncias e evitar carga em excesso no divisor de tensão. O valor digital lido pelo microcontrolador (de 0 a 5V) deverá ser formatado para ser exibido de 0 a 20V em um display LCD com uma casa decimal de precisão (sem usar float).

Requisitos:

- Projetar um divisor de tensão para adaptar a faixa de 0 a 20 V para 0 a 5 V (usar divisor resistivo e encontrar valores de resistores comerciais).
- Utilizar um amplificador operacional (AmpOp) para realizar o casamento de impedância, garantindo que a carga do divisor de tensão não afete a leitura da tensão pelo ADC do PIC.
- Escolher um pino do microcontrolador que tenha canal analógico (verificar qual é o canal analógico) para receber o sinal de entrada e realizar a leitura após o divisor resistivo e AmpOp.
- Escrever um programa em linguagem C para leitura da entrada analógica no canal escolhido. O valor lido pelo ADC será formatado para valor correspondente de 0 a 20 V e exibido no display LCD com uma casa decimal de precisão (sem recorrer ao float).
- Implementar o circuito no SimulIDE (usar uma fonte de tensão linear de 0 a 20 V à entrada do voltímetro, AmpOp, resistores, display LCD etc).
- Simular o circuito carregando o firmware gerado na compilação do programa. Variar a tensão de 0 a 20 V e verificar por meio de um terminal de prova se o valor na saída do divisor é de 0 a 5 V e se o valor exibido no display LCD corresponde à tensão de entrada de 0 a 20 V.
- O documento de entrega do projeto deve conter:
 - Cálculo do divisor de tensão
 - Diagrama esquemático
 - Código em linguagem C
 - Prints da simulação