



**Departamento de Engenharia  
Elétrica**  
**Universidade Federal do  
Ceará *Campus Sobral***

# (SBL0082) Microprocessadores

**Prof. Me. Alan Marques da Rocha**  
**Prof. Dr. Marcelo Marques Simões de Souza**

# Prática 02: Interface LCD 16x2 com PIC18F4520 em Assembly

---

## 1.1 Introdução

A presente prática tem como finalidade integrar conceitos fundamentais de microcontroladores, eletrônica digital e instrumentação computacional por meio da comunicação entre um microcontrolador PIC18F4520 e um *display* LCD alfanumérico de 16 colunas por 2 linhas (LCD 16×2). O experimento realiza a escrita de duas mensagens no *display* a partir de um programa desenvolvido integralmente em linguagem *Assembly*.

O LCD utilizado segue o padrão de controladores compatíveis com o HD44780, amplamente empregado em aplicações embarcadas devido à sua simplicidade de interface paralela. O barramento de dados é de 8 bits e é conectado diretamente ao PORTD do PIC18F4520, enquanto os sinais de controle *RS* (*Register Select*) e *E* (*Enable*) são conectados respectivamente aos pinos RB0 e RB1 do microcontrolador. O sinal *RW* do LCD é conectado permanentemente ao terra, de forma que a comunicação é feita apenas no sentido "microcontrolador → *display*", isto é, apenas escrita.

Além da programação, a prática aborda aspectos essenciais de *hardware*: alimentação do circuito em +5 V, uso adequado do pino de *reset* (MCLR) com *pull-up* de 10 kΩ, ajuste de contraste do LCD via potenciômetro e uso de um cristal externo de aproximadamente 4 MHz para definir o *clock* do PIC. Também é explorada a simulação completa do sistema no *software* Proteus, que permite validar tanto o *firmware* quanto o *hardware* sem necessidade imediata de montagem física em bancada.

Dessa forma, esta atividade funciona como um primeiro "Hello World" em sistemas embarcados: o aluno escreve código *Assembly*, gera o arquivo executável em formato *.hex*, associa esse código a um microcontrolador em um ambiente de simulação e observa um resultado visual direto no *display*. Esse fluxo corresponde exatamente ao ciclo típico de desenvolvimento embarcado profissional (desenvolver → compilar → gravar → testar).

---

---

## 2.2 Objetivos

- Compreender o mapeamento de pinos entre o microcontrolador PIC18F4520 e um *display* LCD 16x2 no modo de 8 bits.
- Desenvolver, em *Assembly*, uma rotina capaz de:
  - inicializar o LCD (configuração de interface, liga *display*, limpa tela);
  - posicionar o cursor no início da primeira e da segunda linha;
  - escrever sequências de caracteres armazenadas na memória de programa (*Flash*).
- Configurar corretamente os *Configuration Bits* (ou “bits de configuração”) do PIC18F4520, definindo tipo de oscilador (OSC = XT para cristal externo de aproximadamente 4 MHz), desabilitando o *watchdog* (WDT = OFF) e habilitando o uso de MCLR externo.
- Gerar o arquivo .hex a partir do código *Assembly* utilizando o ambiente MPLAB X.
- Montar e simular o circuito completo no Proteus, associando o arquivo .hex ao componente PIC18F4520, fornecendo alimentação de +5 V, *clock* e *reset* adequados, e verificando a exibição de mensagens pré-definidas.
- Consolidar o fluxo de desenvolvimento típico: projeto → compilação → gravação → simulação funcional.

## 2.3 Lista de Materiais

- Um kit da placa UFC PICLAB-4520;
- 1 gravador PICKit2 para programação do microcontrolador;
- Computador com a IDE de desenvolvimento MPLABX e *software* Proteus;
- *display* LCD 16x2;
- Potenciômetro de 5K $\Omega$  ou 10k $\Omega$ .

## 2.4 Procedimento Experimental

### 2.4.1 Etapa 1: Criação do projeto no MPLAB X e configuração do dispositivo

1. Abrir o MPLAB X IDE.
-

---

2. Criar um novo projeto (*New Project*):

- Selecionar "*Microchip Embedded*" → "*Standalone Project*".
  - Escolher o dispositivo **PIC18F4520**.
  - Selecionar o *assembler* apropriado (MPASM / PIC18 *toolchain*).
  - Criar o projeto em uma pasta conhecida (por exemplo, PIC\_LCD\_MICROS).
3. Dentro do projeto, criar um novo arquivo fonte Assembly (*New* → *Assembly File*).
4. Escreva o código disponível em (<https://abre.ai/nWxp>) no arquivo criado na etapa anterior. Faça as considerações e observações necessárias. Após a criação, compile o projeto clicando no ícone do martelo ou utilize o atalho F11. Caso o código seja compilado sem erro, irá aparecer a seguinte mensagem no Output: **BUILD SUCCESSFUL (total time: 1s)**.

#### 2.4.2 Etapa 3: Compilação e geração do arquivo .hex

1. No MPLAB X, selecionar o projeto e executar *Build Project* (ou *Clean and Build*).
2. O assembler irá:
  - montar o código Assembly;
  - aplicar os *CONFIG bits*;
  - gerar um arquivo objeto e, ao final, gerar um arquivo **.hex**.
3. Ao término da compilação, localizar o arquivo **.hex** na pasta de saída do projeto. Em projetos MPLAB X típicos, esse arquivo aparece em um subdiretório parecido com:

dist/default/production/<nome\_do\_projeto>.production.hex

4. Esse arquivo **.hex** contém o programa pronto para ser executado pelo PIC18F4520. É esse arquivo que será carregado no Proteus como “firmware” do microcontrolador.

#### 2.4.3 Etapa 4: Montagem do circuito no Proteus

1. Abrir o Proteus e criar um novo projeto/schematic (**.pdsprj**).
-

2. Inserir os seguintes componentes:

- PIC18F4520;
- LCD 16x2 compatível com HD44780 (por exemplo LM016L);
- Cristal (aprox. 4 MHz);
- Dois capacitores cerâmicos  $\approx 22$  pF (um de cada lado do cristal para GND);
- Potenciômetro de 10 k $\Omega$  (ajuste de contraste do LCD);
- Resistor de 10 k $\Omega$  entre MCLR e VDD (pull-up de reset);
- Fonte DC de +5 V e referência GND;
- (Opcional) **Resistor** em série com o backlight do LCD (típico 220  $\Omega$ ) se o módulo tiver pinos de LED.

O circuito simulado deverá seguir a montagem apresentada na Figura 5.1, conforme ilustrado a seguir:

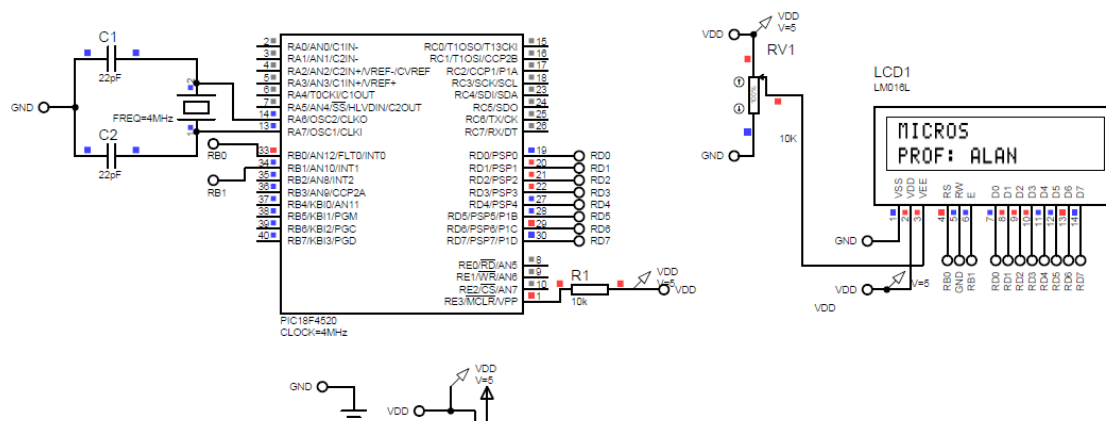


Figure 5.1: Circuito esquemático da prática 02.

## 2.4.4 Etapa 5: Carregar o firmware .hex no PIC dentro do Proteus

1. No *schematic* do Proteus, dar *duplo clique* no componente **PIC18F4520**.
2. Na janela de propriedades do PIC:

- Em *Program File*, selecionar o arquivo **.hex** gerado na Etapa 3.
- Ajustar o campo *Clock Frequency* para **4 MHz**, coerente com o cristal externo e com CONFIG OSC = XT.

3. Confirmar que o PIC está alimentado: VDD ligado a +5 V e VSS a GND. Se o nó VDD aparecer “cinza” durante a simulação, significa que a fonte de 5 V ainda não está realmente inserida no circuito ou conectada corretamente.
4. Ajustar o potenciômetro de contraste no LCD até que os caracteres fiquem visíveis. Se o contraste estiver muito alto ou muito baixo, a tela pode parecer completamente vazia mesmo que o código esteja rodando.

### 2.4.5 Etapa 6: Execução da simulação

1. Clicar em *Run* (Play) no Proteus.
2. Se todas as conexões e o arquivo .hex estiverem corretos, o PIC18F4520 executará:

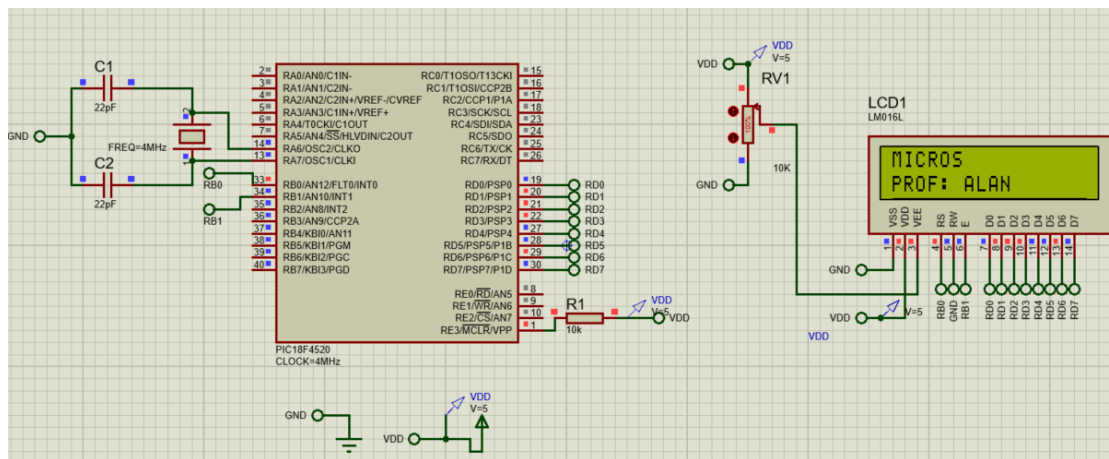


Figure 5.2: Circuito simulado no proteus da prática 02.

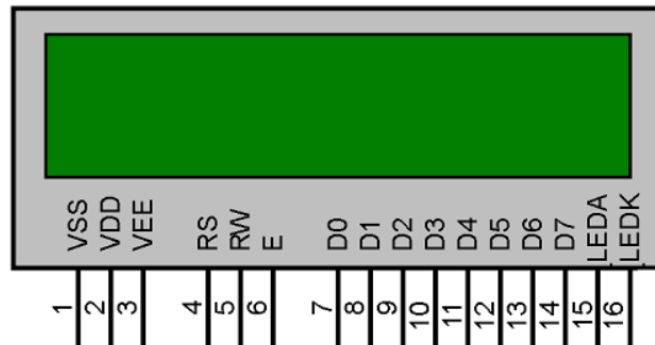
### 2.4.6 Etapa 7: Montagem do circuito físico

1. Monte o circuito em *proto board* replicando o esquemático anteriormente validado em simulação no Proteus.
2. Ao utilizar a placa UFC PICLAB-4520 não se faz necessário realizar a montagem do MCLR ligado ao VDD por meio de um resistor de 10kΩ.
3. Também não se faz necessário a montagem do cristal externo com os capacitores de desacoplamento ao terra.
4. Ligue o *display* LCD 16×2 conforme definido (barramento D0..D7 no PORTD, RS no RB0, E no RB1, RW em GND), ajuste o contraste do LCD com o

---

potenciômetro entre 5 V e GND e alimente tanto o PIC quanto o LCD com a mesma fonte regulada de 5 V. Certifique-se de reproduzir fielmente a mesma fiação usada na simulação.

5. Verifique a pinagem do *display* LCD, realizando a montagem seguindo as informações apresentadas na Figura 5.3. Para mais informações sobre o *display* LCD, visite: (<https://abre.ai/nWxX>)



**Figure 5.3:** Pinagem do LCD 16×2.

6. Verifique as conexões e observe a mensagem que aparece no *display*. Registre o resultado para a inclusão no relatório.

## 2.5 Questionário

1. Explique qual é o papel do sinal RS, do sinal *Enable* (E) e do sinal *Read/Write* (RW) no LCD 16×2. Indique quais desses sinais são conectados ao PIC18F4520 e quais são forçados a um nível fixo no circuito desta prática.
  2. No código em *Assembly*, o barramento de dados do LCD (D0–D7) é conectado em qual porta do PIC18F4520? Justifique por que essa porta foi configurada como saída usando o registrador TRIS correspondente.
  3. Qual é a função da sub-rotina LCD\_Init? Descreva os comandos enviados ao LCD (0x38, 0x0C, 0x01, 0x06) e o efeito que cada um deles provoca no *display*.
  4. Por que o potenciômetro (entre 5 V e GND) precisa estar ligado ao pino de contraste (VEE/V0) do LCD? O que acontece visualmente no *display* se o contraste estiver muito alto ou muito baixo?
-

- 
5. A prática utiliza um cristal externo de aproximadamente 4 MHz ligado aos pinos OSC1 e OSC2 do PIC18F4520, junto com capacitores de aproximadamente 22 pF para GND. Explique a importância desse cristal para o funcionamento do programa e justifique por que o bit de configuração CONFIG OSC foi ajustado para XT.
  6. Na simulação no Proteus, por que é necessário carregar manualmente o arquivo .hex no componente PIC18F4520 e ajustar a frequência de *clock* no campo *Clock Frequency*? Descreva o que aconteceria se o PIC estivesse sem programa ou com *clock* incorreto.
  7. Mostre, no código em *Assembly*, onde as strings "MICROS" e "PROF: ALAN" estão declaradas. A seguir, explique como a rotina LCD\_PrintString percorre essas strings e envia caractere por caractere ao LCD. Use o trecho abaixo como referência:

```
                ORG      0x300
MsgLinha1:
                DB       "MICROS",0x00
MsgLinha2:
                DB       "PROF: ALAN",0x00
```

e

```
LCD_PrintString:
NextChar:
    TBLRD*+      ; TABLAT <- [TBLPTR], TBLPTR++
    MOVF        TABLAT, W, ACCESS ; W = caractere lido
    BNZ         SendChar          ; se W != 0x00, imprime
    RETURN       ; se W == 0x00, acabou a string

SendChar:
    CALL        LCD_Data          ; manda caractere em W
    BRA         NextChar
```

8. Quais são as diferenças práticas entre simular o circuito no Proteus e montar fisicamente o circuito em bancada (por exemplo, utilizando a placa UFC PICLAB-4520)? Comente sobre: alimentação de +5 V, *reset* (MCLR), oscilador, ligação do LCD e necessidade de ajustes físicos (potenciômetro de contraste).
-



---

9. Reproduza em esquema o diagrama mínimo de ligação entre o PIC18F4520 e o LCD 16×2 utilizado na prática, indicando:

- VDD e VSS (alimentação +5 V / GND);
- MCLR com *pull-up* de 10 kΩ;
- Cristal externo e capacitores de 22 pF;
- RS → RB0, E → RB1, RW → GND;
- D0–D7 → RD0–RD7;
- Potenciômetro no pino de contraste do LCD.

10. Refaça a prática utilizando linguagem C em vez de *Assembly*. O comportamento final deve ser o mesmo: escrever "MICROS" na primeira linha e "PROF: ALAN" na segunda linha do LCD. Para isso:

- (a) Crie um novo projeto em C no MPLAB X para o PIC18F4520.
- (b) Implemente em C funções equivalentes a `LCD_Init()`, `LCD_Command()`, `LCD_Data()` e `LCD_PrintString()`.
- (c) Armazene as duas mensagens (linha 1 e linha 2) e envie-as ao LCD após a inicialização, posicionando o cursor corretamente.
- (d) Gere o arquivo `.hex` a partir do código C, carregue esse arquivo no PIC no Proteus e valide o funcionamento.

Inclua no relatório final:

- o código em C (em ambiente L<sup>A</sup>T<sub>E</sub>X, use `\begin{verbatim} ... \end{verbatim}` para apresentá-lo);
  - o arquivo `.hex` gerado;
  - uma captura de tela do Proteus exibindo as mensagens no LCD.
-