

Prática 4:

Usando display LCD tipo 16x2

4.1 – Introdução e objetivos

Frequentemente, o PIC recebe algum tipo de dado e faz algum tipo de tratamento neste dado (segundo sua programação) e depois deve exibi-lo ao seu usuário através de algum tipo de interface. Há vários dispositivos diferentes que podem fazer esta tarefa; contudo o display 16x2 é uma das formas mais comuns de se mostrar, através de texto, ao usuário alguma informação. A Figura 5.1 ilustra um exemplo deste tipo de display que comporta 16 caracteres por cada uma de suas duas linhas. Sua vantagem é seu custo que é relativamente barato e sua relativa facilidade de uso que são ao microcontrolador a capacidade de exibir letras e números para seu usuário.



Figura 5.1 - Ilustração de um display 16x2.

Deste modo, o objetivo desta prática é ilustrar como o display 16x2 pode ser usado com uma biblioteca de comunicação para este tipo de dispositivo e demonstrar como se trabalha com este dispositivo para exibir texto e números. O display 16x2 é ensinado neste capítulo pois ele também deverá ser usado em práticas futuras para enriquecer a aplicação e ilustrar valores e textos de interação com o usuário. Por este motivo, o entendimento desta prática é fundamental.

4.2 – Funcionamento do display 16x2

A Figura 5.2a ilustra o padrão de pinagem do display 16x2. Este padrão pode as vezes ser encontrado na ordem sequencial, o que facilita seu manuseio. Contudo, é também comum encontra-lo na ordem ilustrada na Figura 5.2a que não é sequencial. Geralmente, os números que indicam a ordem é indica na cor branca na placa que dá suporte ao display e o usuário deve tomar muito cuidado na hora de montar o circuito e obedecer fielmente esta referência e os mapas de pinos ilustrados na Figura 5.2b. A função destes pinos é padronizada e deve ser obedecida. Fique atento aos pinos 1 (terra) e 2 (alimentação +5V) que não podem ser ligados invertidos e nem podem receber uma tensão acima de 5v pois podem danificar permanentemente o display.

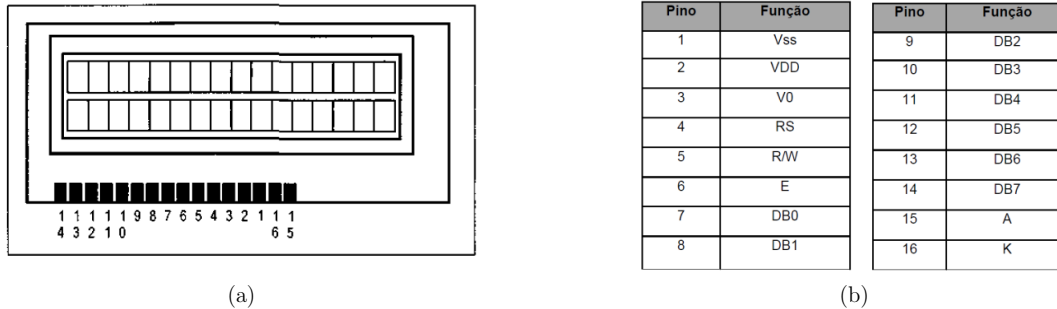


Figura 5.2 - (a) Arranjo dos pinos do display 16x2 visto em ordem não-sequencial. (b) Na tabela são indicados os pinos. Destaca-se também que em muitos casos o padrão de pinagem pode também ser sequencial. Consulte o datasheet do fabricante.

Segundo a tabela da Figura 5.2b, o pino V0 é utilizado para controle de contraste dos caracteres exibidos e pode ser ligado a um potenciômetro de 10k ohms para controle do contraste. Os demais pinos são utilizados para uma lógica de controle onde o PIC envia códigos para que sejam exibidos caracteres neste display. Nesta lógica de controle, o pino RS (register select) é indicado para definir se o dado enviado para os pinos 7 a 14 tratam de um comando (Rs=0) ou se trata de um dado (Rs=1). O pino R/W muda o estado do LCD entre leitura (R/W=1) e escrita (R/W=0) pois o LCD permite que sejam lidos e escritos dados nele por possuir uma pequena memória RAM interna para guardar os dados a serem exibidos. É comum ver este pino ligado diretamente ao terra pois a operação de escrita na memória é a mais comum. O pino E (enable) indica em sua borda de descida que há um dado novo no barramento de dados que correspondem os pinos DB0 a DB7. Estes dados podem ser transmitidos com 8 bits de uma vez ou somente 4. Neste último caso, os dados são enviados por 2 pacotes de 4 bits cada. Por fim, os pinos A (anode) e K (katode) são usados para ligar a luz de fundo do display.

Antes de exibir caracteres no display, é necessário inicializá-lo. Para isto deve ser criada uma rotina que deve, resumidamente, fazer:

- (i) esperar 15ms para que o display tenha sua energização garantida;
- (ii) enviar o código 0x30 e esperar pelo menos 5ms;
- (iii) repetir a operação anterior outras duas vezes;
- (iv) enviar um código indicando se a transmissão é de 8 ou 4 bits e o tipo de mapa de caracteres será usado. Este mapa de caracteres é armazenado em sua memória local e esperar por 40us.
- (v) enviar o comando de limpar a tela e posicionar o cursor na primeira linha da primeira coluna e esperar por 1,8ms;
- (vi) várias outras operações acontecem como o envio de comando de deslocamento automático do cursor. O display tem uma série de comandos de controle que podem ser consultados em uma literatura específica sobre o assunto.

Neste roteiro estes detalhes do protocolo de comunicação entre microcontrolador e display não serão abordados pois será usada uma biblioteca de software que já implementa automaticamente esta

comunicação com seus códigos e suas restrições de tempo. O objetivo deste capítulo é utilizar o display 16x2 da forma correta ainda que não se conheça todas as complexidades deste dispositivo que são abstraídas pela biblioteca de software indicada aqui.

4.3 – Aplicação

Nesta aplicação, um visor 16x2 é ligado ao PIC para exibir o texto "X vale: " seguido de um valor inteiro de 2 dígitos que representa o conteúdo de uma variável inteira no PIC que é incrementada e reiniciada constantemente para que o display possa exibir constantemente estas variações. Como poderá ser constatado, o display só aceita exibir caracteres. Assim, o conteúdo da variável inteira deve ser antes convertido para caractere para que possa ser exibida. Detalhes são tratados na sequência.

4.3.1 – O circuito

Para o exemplo de uso do display 16x2, monte o circuito da Figura 5.3. Destaca-se que a pinagem do display pode variar de modelo para modelo.

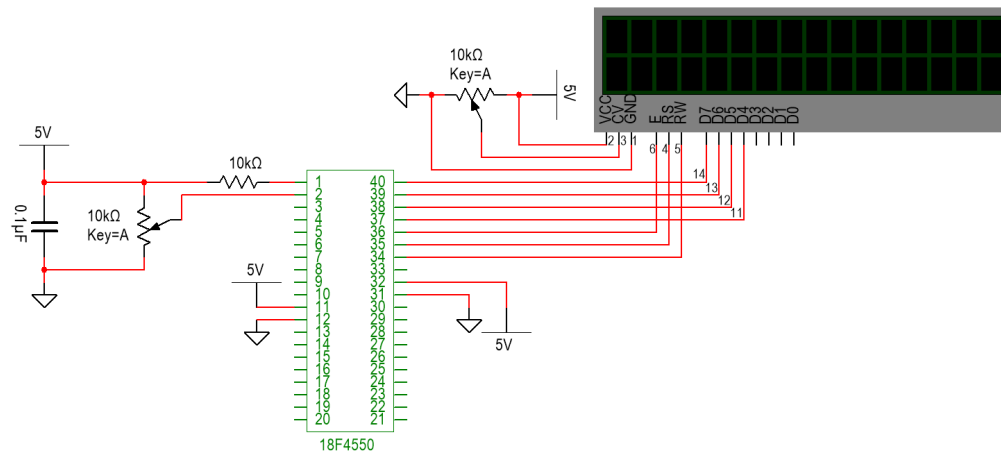


Figura 5.3 - Circuito ilustrando a ligação do display 16x2 com o PIC 18F4550.

4.3.2 – O código

Para facilitar o uso do display, será usada uma biblioteca contida no arquivo "mod_lcd.c" (que pode ser encontrado no site da disciplina) para facilitar a comunicação com o display de tal forma que suas peculiaridades e códigos sejam invisíveis ao seu usuário que terá apenas a preocupação de enviar texto para o display.

Antes de criar o firmware da prática, vá até a biblioteca "mod_lcd.c" (que pode ser obtida no site da disciplina) e em suas primeiras linhas, defina os pinos do PIC que serão ligados aos pinos do display conforme layout elétrico da Figura 5.3. O código da sequência ilustra como deve ficar esta parte do código para que ele seja compatível com as designações de pinos da Figura 5.3. Se o usuário desejar mudar alguns pinos no esquemático, ele também deve proceder com a correspondente mudança nesta parte do código do arquivo "mod_lcd.c". Neste exemplo, o pino lcd_enable (que segundo tabela da Figura 5.2b é o pino 6 do display) é segundo a Figura 5.3 ligado ao pino B3 (pino 36) do 18F4550. Analogias podem ser feitas aos outros pinos que são designados pelas linhas 2 a 8 do código 5.1.



Código 5.1 - Código parcial da biblioteca "mod_lcd.c" utilizada para enviar caracteres para exibição em display 16x2.

```
1 #ifndef lcd_enable
2 #define lcd_enable    pin_b3    // pino enable do LCD
3 #define lcd_rs       pin_b2    // pino rs do LCD
4 #define lcd_rw       pin_b1    // pino rw do LCD
5 #define lcd_d4       pin_b4    // pino de dados d4 do LCD
6 #define lcd_d5       pin_b5    // pino de dados d5 do LCD
7 #define lcd_d6       pin_b6    // pino de dados d6 do LCD
8 #define lcd_d7       pin_b7    // pino de dados d7 do LCD
9 #endif
10 ...
```

Uma vez tendo montado o circuito, analise o código da sequência onde é ilustrado o emprego da biblioteca de controle do display.

Código 5.2 - Código da aplicação que emprega o display 16x2.

```
1 #include "main.h"
2 #include <mod_lcd.c>
3 #include <stdlib.h>
4
5 void main()
6 {
7     int x, tamanho;
8     char texto[4];
9     int i;
10    ...
11    lcd_ini();
12    x = 90; //inicializa em qq valor
13
14    while(true)
15    {
16        itoa(x,10, texto);
17        tamanho = strlen(texto);
18        for (i=0; i<tamanho;i++)
19        {
20            lcd_pos_xy(i+1,1);
21            lcd_escreve(texto[i]);
22        }
23        if (x > 110)
24            x = 90;
25        x++;
26        delay_ms(1000);
27    }
28 }
```

Neste código, a variável x é inicializada com o valor 90 (linha 12). A cada 1 segundo (linha 26), a variável é incrementada (linha 25) e seu valor é exibido no display (linhas 20 e 21). Quando ela atinge o valor 110 (linha 23), ela é reiniciada (linha 24). Assim o display mostra uma contagem de 90 a 110. Ainda, nota-se que usamos a função "itoa" (linha 16) para converter os dígitos da variável inteira "x" em caracteres que serão armazenados no vetor "texto". Logo após a conversão, é estimado na linha 17 quantos caracteres tem a variável convertida em texto.

Nota-se que a função "lcd_escreve" exhibe somente 1 caractere por vez e por isto temos que varrer uma posição por vez do vetor texto. Esta tarefa é feita pelo laço da linha 18. Além disto, toda vez que for escrito um novo caractere, existe a necessidade de se indicar a posição deste caractere na



tela através da função `lcd_pos_xy`. Faz-se importante destacar que antes de exibir qualquer caractere no display, é necessário fazer sua inicialização e estabelecer uma comunicação inicial que no caso desta biblioteca é feita pela função `lcd_ini` (linha 11). Uma vez chamada esta função de inicialização, a tela já se encontra pronta para receber caracteres.