FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA
ENSINANDO E APRENDENDO

Aula 12
Portas de Entrada e Saída – Parte IV
(Display LCD)

Microcontroladores PIC18 – Programação em C

Prof. Ítalo Jáder Loiola Batista
Universidade de Fortaleza - UNIFOR
Centro de Ciências Tecnológicas - CCT
E-mail: italoloiola@unifor.br

Jan / 2011

Display LCD

- Cristal Líquido é um estado da matéria entre o estado sólido e o líquido;
- Alfanuméricos ou Gráficos;
- Número de Linhas e Colunas;
- Resoluções variadas;
- Número de pinos para conexão;
- Com ou sem *backlight;*
- Tipo de caracteres;
- Tecnologias e Temporizações diversas.

Display LCD

- Displays LCD comerciais;

Número de Colunas	Número de Linhas	Quantidade de pinos
8	2	14
12	2	14/15
16	1	14/16
16	2	14/16
16	4	14/16
20	1	14/16
20	2	14/16
20	4	14/16
24	2	14/16
24	4	14/16
40	2	16
40	4	16

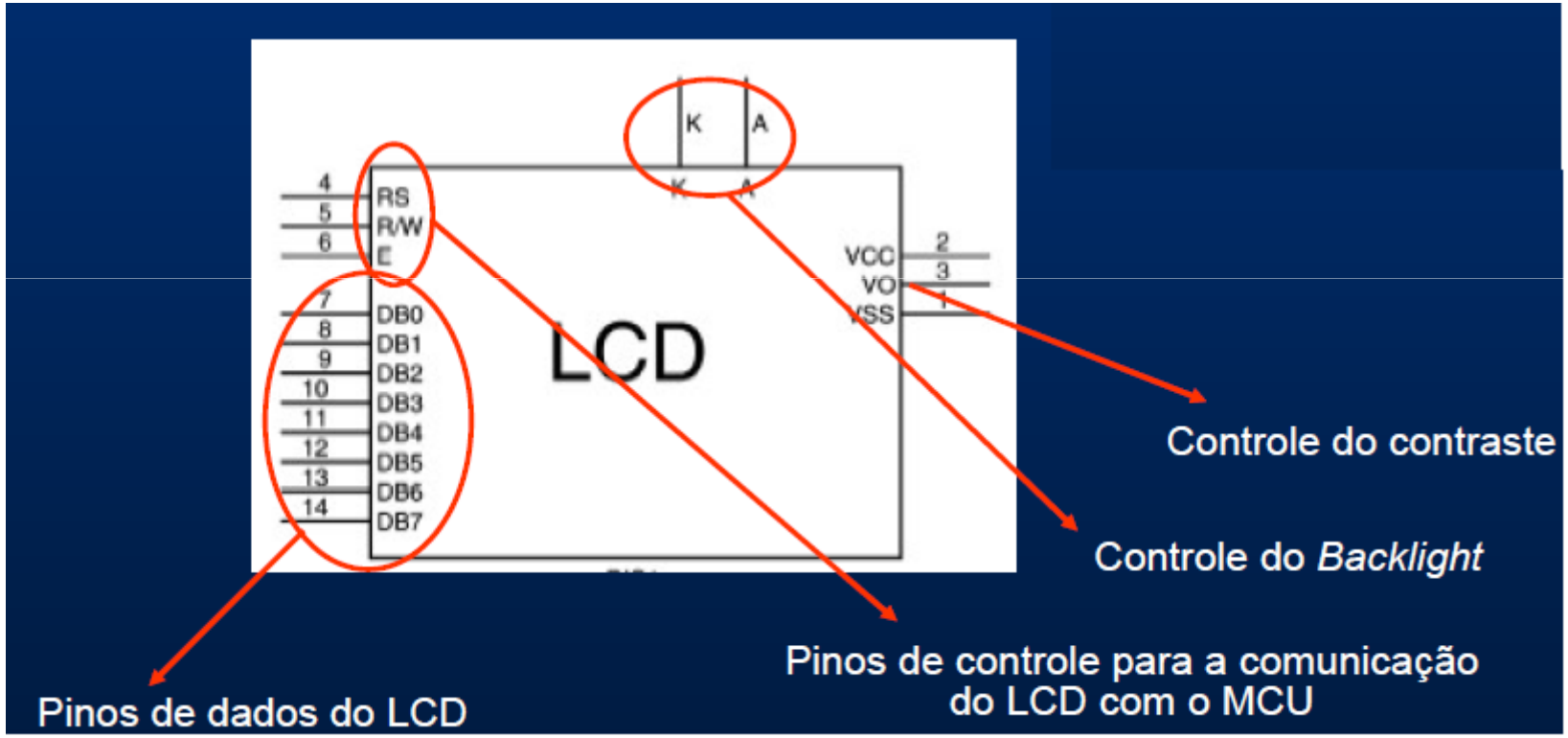
Display LCD

- Pinagem do Displays LCD 16x2;

Pino	Função	Descrição
1	Alimentação	Terra ou GND
2	Alimentação	VCC ou +5V
3	V0	**Tensão para ajuste de contraste (ver Figura 1)**
4	RS Seleção:	1 - Dado, 0 - Instrução
5	R/W Seleção:	1 - Leitura, 0 - Escrita
6	E Chip select	1 ou (1 → 0) - Habilita, 0 - Desabilitado
7	B0 LSB	Barramento de Dados
8	B1	
9	B2	
10	B3	
11	B4	
12	B5	
13	B6	
14	B7 MSB	
15	A (qdo existir)	Anodo p/ *LED backlight*
16	K (qdo existir)	Catodo p/ *LED backlight*

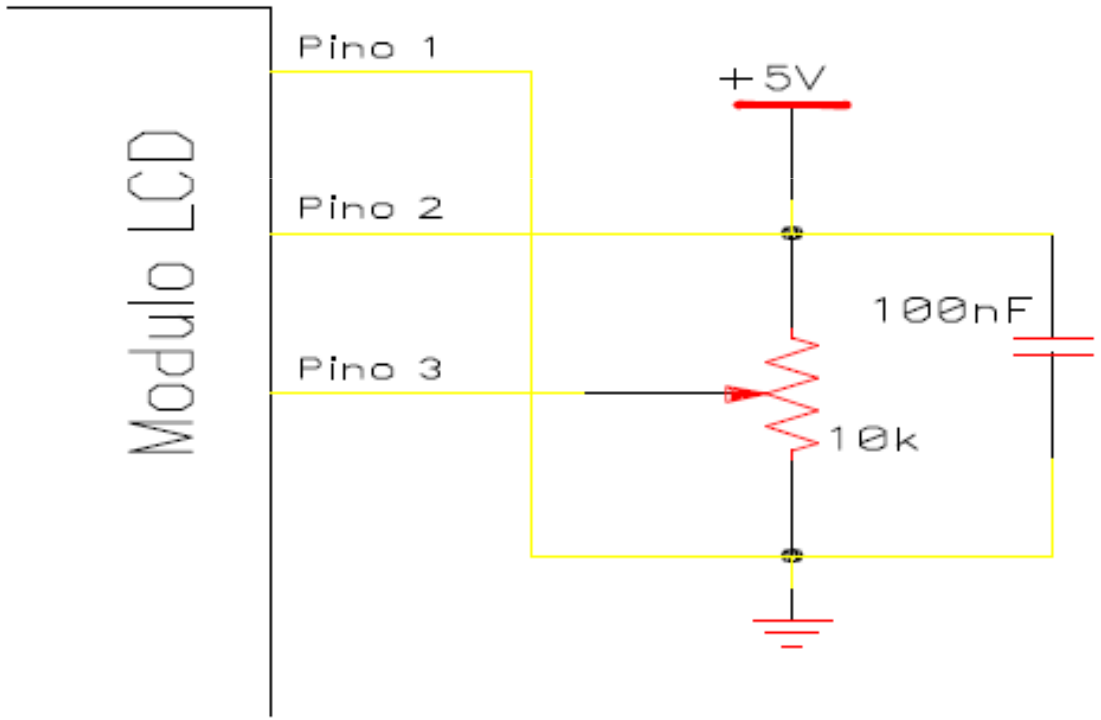
Display LCD

- Pinagem do Displays LCD 16x2;



Display LCD

- Detalhe do controle de contraste do display LCD;



Display LCD

- A comunicação com o LCD pode se dar de duas formas, o LCD pode receber:
  - Dados;
  - Instruções;
- O pino 4 (RS) tem a função de informar ao LCD se o byte que se encontra na via de dados (DB7:DB0) é uma dado ou uma instrução;
- O pino 5 (R/W) tem a função de ativar o ciclo de leitura ou de escrita;
- Para dar início a um ciclo de leitura ou de escrita, é necessário aplicar um pulso no início no pino 6 (E);

Display LCD

- O pino 4 (RS)

RS	Significado
0	Instrução
1	Dado

- O pino 5 (R/W)

R/W	Significado
0	Escrita
1	Leitura

Display LCD - Inicialização

1. Início da Inicialização
2. Aguardar 15ms após VDD atingir 4,5V
3. Instrução = 30h
4. Delay maior que 4,1ms
5. Instrução = 30h
6. Delay maior que 100ums
7. Instrução = 30h
8. Delay maior que 40us
9. Instrução = 38h (8 bits de dados; Formato 5x7 mais cursor; Duas linhas de caracteres)
10. Delay maior que 39us

Display LCD - Inicialização

11. Instrução = 0Fh (Display ativo com cursor e piscando)
12. Delay maior que 39us
13. Instrução = 06h (Desloca cursor à direita com a entrada de um novo caractere)
14. Delay maior que 1,53ms
15. Instrução = 01h (Limpa display e retorna cursor à 1ª posição da 1ª coluna)
16. Delay maior que 1,53ms
17. Fim de inicialização

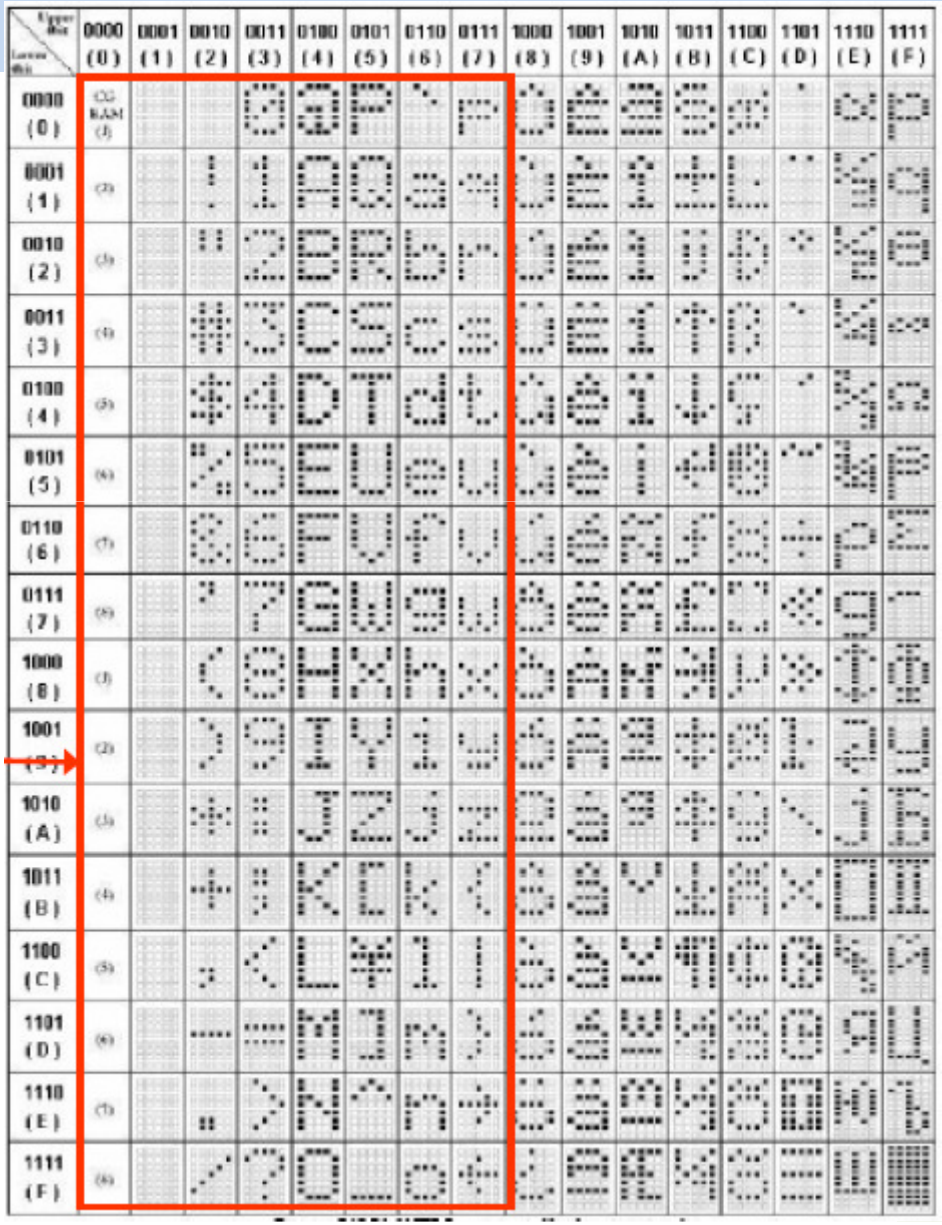
Display LCD

- Instruções mais comuns

DESCRIÇÃO	MODO	RS	R/W	Código (Hexa)
Display	Liga (sem cursor)	0	0	0C
	Desliga	0	0	0A / 08
Limpa Display com Home cursor		0	0	01
Controle do Cursor	Liga	0	0	0E
	Desliga	0	0	0C
	Desloca para Esquerda	0	0	10
	Desloca para Direita	0	0	14
	Cursor Home	0	0	02
	Cursor Piscante	0	0	0D
	Cursor com Alternância	0	0	0F
Sentido de deslocamento do cursor ao entrar com caracter	Para a esquerda	0	0	04
	Para a direita	0	0	06
Deslocamento da mensagem ao entrar com caracter	Para a esquerda	0	0	07
	Para a direita	0	0	05
Deslocamento da mensagem sem entrada de caracter	Para a esquerda	0	0	18
	Para a direita	0	0	1C
End. da primeira posição	primeira linha	0	0	80
	segunda linha	0	0	C0

Display LCD

- Caracteres que podem ser exibidos no display;



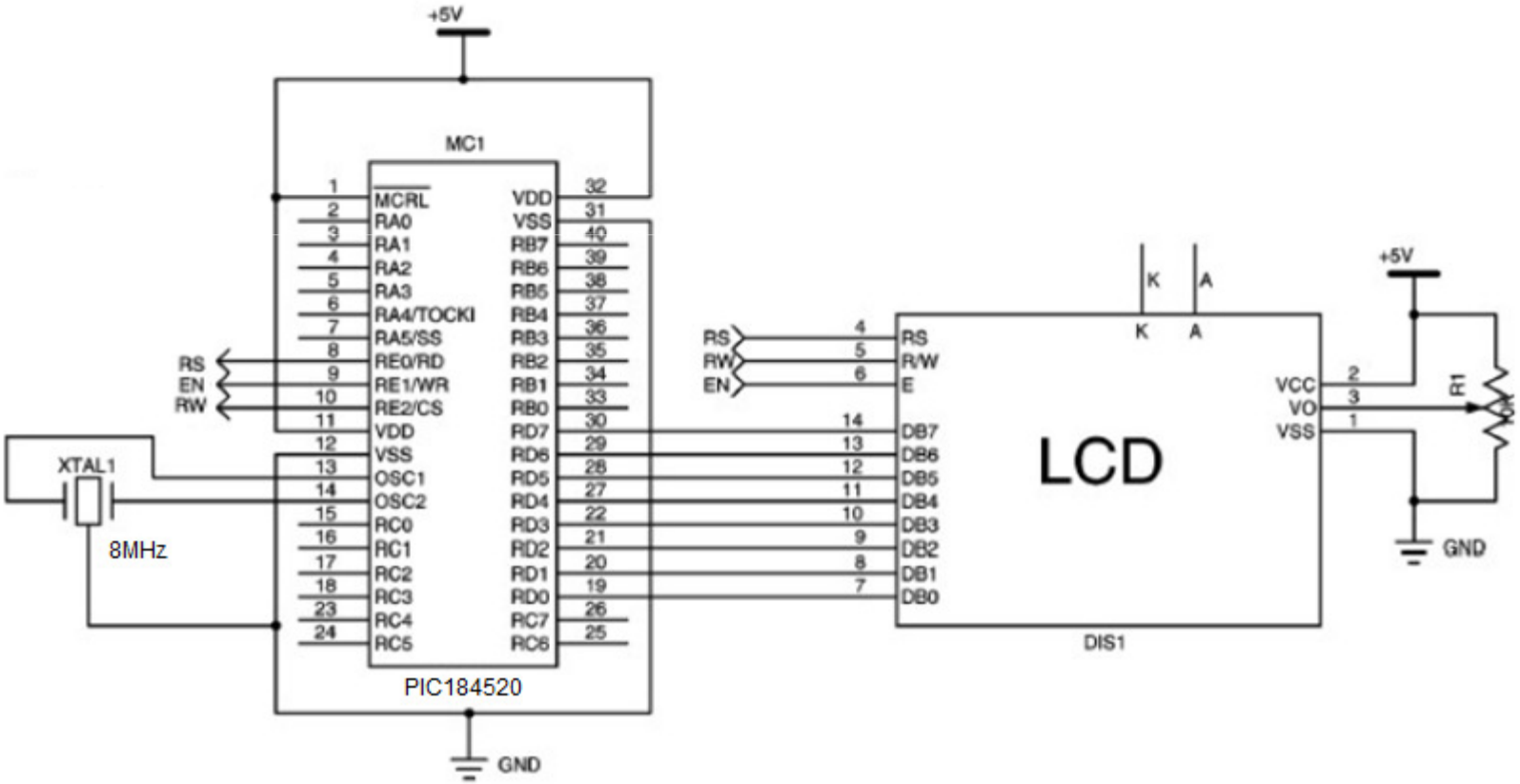
Display LCD

- Tabela de endereços dos caracteres

LCD 16x2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
linha 1	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
linha 2	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

Display LCD

- Esquema Elétrico de Conexão de um LCD com um uC;



Programação de LCDs:

- Programar displays de LCDs
  - Configurar PORTs adequadas ao uso de LCDs;
  - Configurar (inicializar) o LCD;
  - Enviar dados de comandos para o LCD;
  - Enviar dados de leitura ou escrita (LCD).
- Para escrever um caractere em uma determinada posição do LCDs:
  - Envia-se ao LCD um comando com o endereço da posição onde se quer escrever;
  - Envia-se logo a seguir o caractere a ser escrito na posição.

Display LCD / Código-fonte

- LCD_8bits.h

  Arquivo cabeçalho com as definições dos pinos utilizados como via de dados, vias de controle e os protótipos das funções;

- LCD_8bits.c

- Main_32.c

  Arquivo principal responsável por efetuar testes no display e exibir uma mensagem;

Display LCD / Funções

Função	Descrição
IniciaLCD	Inicializa LCD *controller*
TestPixelsLCD	Acende todos os *pixels* do LCD
EscInstLCD	Envia instrução para o LCD
EscDataLCD	Escreve um caractere na posição apontada pelo cursor
EscStringLCD	Escreve uma string lida na memória de dados a partir da posição apontada pelo cursor
EscStringLCD_ROM	Escreve uma string lida na memória de programa a partir da posição apontada pelo cursor
TesteBusyFlag	Verifica se o LCD *controller* está ocupado executando alguma instrução
Pulse	Aplica pulso de para leitura ou escrita no LCD
_Delay100us	Delay de 100us
_Delay5ms	Delay de 5ms
DelayFor20TCY	Delay de 20 ciclos de instrução do oscilador

LCD_8bits.h

Display LCD / Código-fonte

```
8  #ifndef __LCD_8BITS_H
9  #define __LCD_8BITS_H
10 //*************************************************************
11 //definições do port ligado no LCD
12 //*
13 #define PORT_LCD       PORTD          //LCD ligado no PORTD
14 #define PORT_CONT_LCD    PORTE          //PORT de controle do LCD
15 #define TRIS_PORT_LCD    TRISD          //direção dos pinos
16 #define TRIS_CONT_LCD    TRISE          //direção dos pinos
17 //*************************************************************
18 //definições dos pinos de controle
19 #define _RS PORTEbits.RE0              //pino dado/instrução
20 #define _EN PORTEbits.RE1              //pino enable
21 #define _RW PORTEbits.RE2              //pino escrita/leitura
22 //*************************************************************
23 //protótipos de funções
24 void IniciaLCD (unsigned char NL);
25 void Pulse(void);
26 void _Delay100us(void);
27 void _Delay5ms(void);
28 void TestPixelsLCD(void);
29 void DelayFor20TCY( void);
30 void DelayFor18TCY( void);
31 unsigned char TesteBusyFlag(void);
32 void EscDataLCD(char _data);
33 void EscInstLCD(unsigned char _inst);
34 void EscStringLCD(char *buffer);
35 void EscStringLCD_ROM(const rom char *buffer);
36 //*************************************************************
37 #endif
38
```

Identificador que impede a definição a seguir seja duplicada se o arquivo cabeçalho foi incluído em outro arquivo-fonte associado ao projeto.

LCD_8bits.c

Display LCD / Código-fonte – 1

```
/*****************************************************************
Esta biblioteca contém um conjunto de funções que permitem ao microcontrolador
se comunicar com o LCD controller HD44780.
//****************************************************************/
#include <p18cxxx.h>                  //diretiva de compilação
#include<delays.h>                     //diretiva de compilação
#include "LCD_8bits.h"                 //diretiva de compilação
/*****************************************************************
A função IniciaLCD() recebe como argumento um valor que irá inializar o LCD com:

valor = 1 -> inicializa o LCD com uma linha
valor != 1 -> inicializa o LCD com linha dupla

Quando o programa retorna ao ponto de chamada, o LCD mostra o cursor piscando
na primeira posição da primeira linha.
//****************************************************************/
```

LCD_8bits.c

Display LCD / Código-fonte - 2

NL: Define o número de linhas que estarão ativas;

```
25  void IniciaLCD (unsigned char NL)
26  {
27     const unsigned char Seq_Inic[3] = {0x0F, 0x06, 0x01};    //declaração de vetor
28     unsigned char i;                                         //declaração de variável local
29     char x;                                                  //declaração de variável local
30     _EN = 0;                          //envia intrução
31     _RS = 0;                          //limpa pino enable
32     _RW = 0;                          //ativa ciclo de escrita
33     ADCON1 = 0x0F;                      //configura PORT de controle com digital
34     TRIS_CONT_LCD = 0;                   //configura PORT de controle como saída
35     TRIS_PORT_LCD = 0;                   //configura PORT de dados como saída
36  //************* envia para o LCD o comando 0x30 três vezes
37     for(i=0;i<3;i++)
38        {
39        PORT_LCD = 0x30;                 //comando 0x30
40        Pulse();                        //aplica pulso enable no LCD
41        _Delay5ms();                     //delay 5ms
42        }
43  //************ configura linha simples ou linha dupla
44     if(NL == 1)PORT_LCD = 0x30;          //se NL=1, ativa uma linha
45     else PORT_LCD = 0x38;                //se NL!=1, ativa duas linhas
46     Pulse();                        //aplica pulso enable no LCD
47     _Delay5ms();                     //delay 5ms
48  //*********************************
49     for(i=0;i<3;i++)
50        {
51        PORT_LCD = Seq_Inic[i];           //LCD recebe comando
52        Pulse();                        //aplica pulso enable no LCD
53        _Delay5ms();                     //delay 5ms
54        }
55     TRIS_PORT_LCD = 0xFF;               //configura PORT de dados como entrada
56  }                                  //final  da função IniciaLCD
```

LCD_8bits.c

Display LCD / Código-fonte - 3

São utilizadas para gerar a base de tempo exigida pelo LCD

Precisam que o arquivo cabeçalho delay.h seja incluído no projeto.

Desenvolvida para freqüência de clock de 8Mhz.

```
58  //esta função escreve comando/dado no LCD
59  void Pulse(void)
60     {
61     DelayFor20TCY();                //delay de 20 ciclos de clock
62     _EN = 1;                       //seta pino enable
63     DelayFor20TCY();                //delay de 20 ciclos de clock
64     _EN = 0;                       //limpa pino enable
65     }
66  //***************************************************************************
67  //                          funções de delay
68  //***************************************************************************
69  //delay de 100us
70  void _Delay100us(void)
71  {
72     Delay100TCYx(2);                //delay 100us
73  }
74  //***************************************************************************
75  //delay de 5ms
76  void _Delay5ms(void)
77  {
78     Delay10KTCYx(1);                //delay 5ms
79  }
80  //***************************************************************************
81  //delay 20 ciclos do oscilador principal
82  void DelayFor20TCY( void )
83  {
84     Nop(); Nop();Nop(); Nop(); Nop();
85     Nop(); Nop();Nop(); Nop(); Nop();
86     Nop(); Nop();Nop(); Nop(); Nop();
87  }
```

LCD_8bits.c

Display LCD / Código-fonte - 4

Verifica se o LCD está ocupado executando alguma instrução ou se ele está livre;

```
96  void DelayFor18TCY( void )
97  {
98      Nop(); Nop();Nop(); Nop(); Nop(); Nop(); Nop();
99      Nop(); Nop();Nop(); Nop(); Nop(); Nop(); Nop();
100 }
101 //*****************************************************************
102 //             funções de acesso ao LCD
103 //*****************************************************************
104 /*esta função fica aguardando o LCD controller terminar de executar
105 a instrução atual. ela retorna o  valor 0 quando a instrução terminar.*/
106 unsigned char TesteBusyFlag(void)
107 {
108 //   TRIS_PORT_LCD = 0xFF;            //configura PORT de dados como entrada
109
110     _RW = 1;                         //ativa ciclo de leitura
111     _RS = 0;                       //ciclo de intrução
112     DelayFor20TCY();                 //delay de 20 ciclos de clock
113     _EN = 1;                               //seta pino enable
114     DelayFor20TCY();                 //delay de 20 ciclos de clock
115     if(PORT_LCD&0x80)                      //leitura do bit busy flag
116     {                                        //se bit busy == 1, LCD ocupado
117         _EN = 0;                           //reseta pino enable
118         _RW = 0;                           //reseta linha de escrita
119         return 1;                           //LCD ocupado, retorna 1
120     }
121     else                                   //se busy flag == 0, LCD livre
122     {
123         _EN = 0;                           //reseta pino enable
124         _RW = 0;                         //ativa ciclo de escrita
125         return 0;                        //LCD livre, retorna 0
126     }
```

LCD_8bits.c

Display LCD / Código-fonte - 5

Verifica se o LCD está ocupado executando alguma instrução ou se ele está livre;

Verifica se o LCD está ocupado executando alguma instrução ou se ele está livre;

```
128  //esta função escreve um caractere na pocição apontada pelo cursor.
129  void EscDataLCD(char _data)
130  {
131      TRIS_PORT_LCD = 0;              //configura PORT de dados como saída
132      PORT_LCD = _data;              //escreve dado
133      _RS = 1;                      //envia dado
134      _RW = 0;                      //ativa ciclo de escrita
135      Pulse();                      //aplica pulso enable no LCD
136      _RS = 0;                      //envia instrução
137       DelayFor20TCY();               //delay de 20 ciclos de clock
138      TRIS_PORT_LCD = 0xFF;           //configura PORT de dados como entrada
139  //*****************************************************************
140  }                                 //final da função EscDataLCD
141  //*****************************************************************
142  //esta função envia uma instrução para o LCD.
143  void EscInstLCD(unsigned char _inst)
144  {
145      TRIS_PORT_LCD = 0;              //configura PORT de dados como saída
146      PORT_LCD = _inst;              //escreve instrução
147      _RS = 0;                      //envia intrução
148      _RW = 0;                      //ativa ciclo de escrita
149      Pulse();                      //aplica pulso enable no LCD
150      _RS = 0;                      //envia dado
151       DelayFor20TCY();               //delay de 20 ciclos de clock
152      TRIS_PORT_LCD = 0xFF;           //configura PORT de dados como entrada
153  //*****************************************************************
154  }                                 //final da função EscInstLCD
```

LCD_8bits.c

Display LCD / Código-fonte - 6

Envia para o LCD a *string* lida na memória de dados que será exibida no *display* a partir da posição apontado pelo cursor;

Envia para o LCD a *string* lida na memória de programa que será exibida no *display* a partir da posição apontado pelo cursor;

```
155  //*****************************************************************/
156  /*esta função escreve no LCD uma string lida da memória RAM a partir
157  da posição aponntada pelo cursor.*/
158  #pragma code My_codigo = 0x200
159  void EscStringLCD(char *buff)
160  {
161      while(*buff)                          //escreve caractere até econtrar null
162      {
163         while(TesteBusyFlag());           //espera LCD terminar de executar instruçã
164         EscDataLCD(*buff);               //escreve no LCD caractere apontado por buff
165           buff++;                          // Incrementa buffer
166      }
167  //*****************************************************************
168  }                                  //final da função EscStringLCD
169  #pragma code
170  //*****************************************************************
171  /*esta função escreve no LCD uma string lida da memória de programa
172  a partir da posição aponntada pelo cursor.*/
173  void EscStringLCD_ROM(const rom char *buff)
174  {
175      while(*buff)                          // Write data to LCD up to null
176      {
177         while(TesteBusyFlag());           //espera LCD controller terminar de executa
178           EscDataLCD(*buff);              //escreve no LCD caractere apontado por buf
179           buff++;                          // Incrementa buffer
180      }
181           return;
182  //*****************************************************************
183  }                                  //final da função EscStringLCD_ROM
```

LCD_8bits.c

Display LCD / Código-fonte - 7

Função que acende todos os pixels do display do LCD;

Escreve cursor na primeira linha

Posiciona cursor na segunda linha

Caractere com todos os pixels acesos

```
185  //esta função testa o LCD acendendo todos os pixels do display.
186  void TestPixelsLCD(void)
187  {
188  unsigned char BffCheio[32];              //declaração de vetor
189  unsigned char i;                        //declaração de variável local
190     EscInstLCD(0x80);                     //posiciona cursor na primeira posição da primeira linha
191     while(TesteBusyFlag());                  //espera LCD controller terminar de executar instrução
192
193     for(i=0;i<32;i++)                     //laço de iteração
194     {
195        if(i<16)                          //i < 16?
196        {                                 //sim, executa bloco de código a seguir
197           EscDataLCD(0xFF);              //escreve caractere na posição pantada pelo  cursor
198           while(TesteBusyFlag());           //espera LCD controller terminar de executar instrução
199        }
200        else    if(i==16)                 //i==16?
201        {                                 //sim, executa bloco de código a seguir
202           EscInstLCD(0xC0);              //posiciona cursor na primeira posição da segunda linha
203           while(TesteBusyFlag());           //espera LCD controller terminar de executar instrução
204
205           EscDataLCD(0xFF);              //escreve caractere na posição pantada pelo  cursor
206           while(TesteBusyFlag());           //espera LCD controller terminar de executar instrução
207        }
208        else                              //se i !=16 executa bloco de c[odigo a seguir
209        {
210           EscDataLCD(0xFF);              //escreve caractere na posição pantada pelo  cursor
211           while(TesteBusyFlag());           //espera LCD controller terminar de executar instrução
212        }
213     }
214  //*****************************************************************
215  }
```

Main_32.c

Display LCD / Código-fonte - 1

```
8  #include <p18f4520.h>          //diretiva de compilação
9  //#include <delays.h>          //diretiva de compilação
10 #include "Lcd_8bits.h"         //diretiva de compilação
11 //*****************************************************************
12 //protótipos de funções
13  void Inic_Regs (void);
14  //*****************************************************************
15 void main(void)                          //função main
16 {
17 char buf [17] = {"Seja bem vindo!"};      //declaração de vetor inicializado
18 char buf02 [17] = {"   LCD 16 x 2"};     //declaração de vetor inicializado
19 int dly=0;                              //declaração de variável local inicializada
20 //***********************************
21    Inic_Regs ();                        //configurar SFRs
22    IniciaLCD (2);                        //inicializar LCD controller HD44780
23    TestPixelsLCD();                     //teste no LCD - acende todos os pixels.
24 //***********************************
25 //delay de 3 segundos
26    for(dly=0;dly<600;dly++)              //comando de iteração
27    {
28       _Delay5ms();                      //delay de 5ms
29    }
```

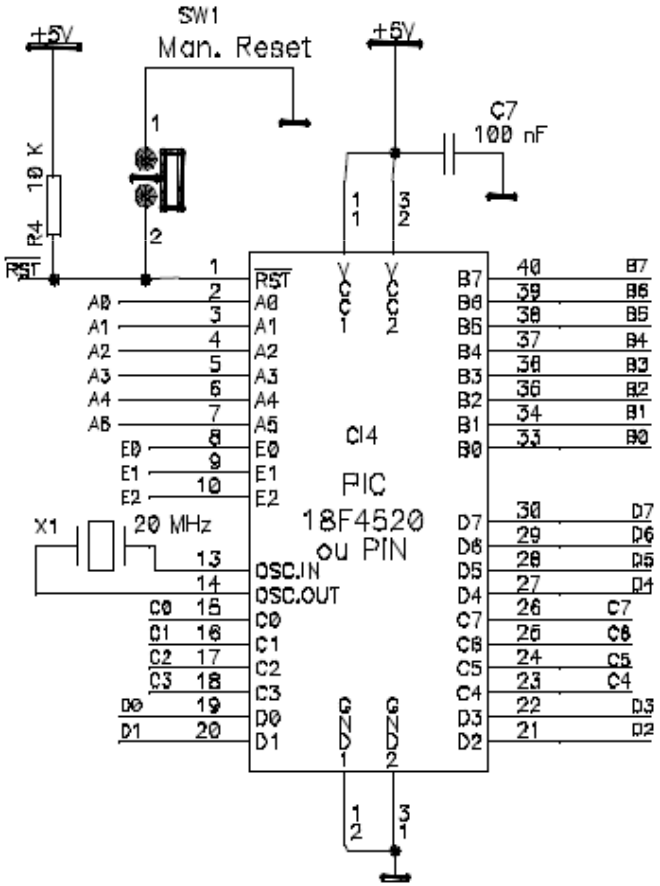
Main_32.c

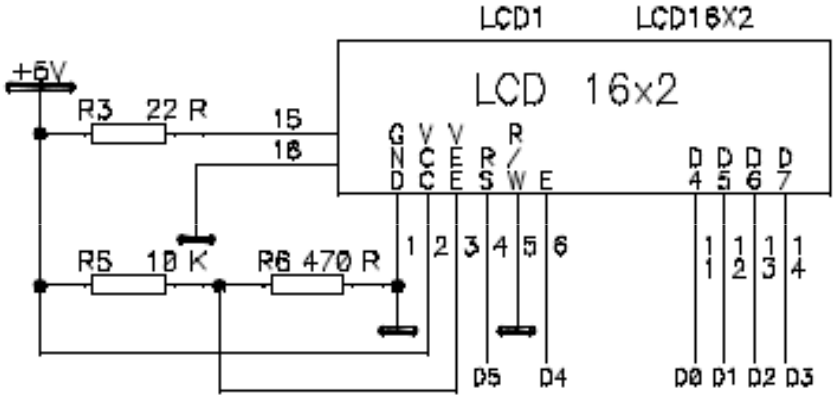
Display LCD / Código-fonte - 2

```
31      EscInstLCD(0x01);  //limpa display e mostra cursor piscando na 1a posição da 1a linha
32      while(TesteBusyFlag());      //espera LCD controller terminar de executar instrução
33
34      EscStringLCD(buf);           //escreve string no LCD
35      while(TesteBusyFlag());      //espera LCD controller terminar de executar instrução
36
37      EscInstLCD(0xC0);           //posiciona cursor na primeir aposicção  da segunda linha
38      while(TesteBusyFlag());     //espera LCD controller terminar de executar instrução
39
40      EscStringLCD(buf02);         //escreve string no LCD
41      while(TesteBusyFlag());      //espera LCD controller terminar de executar instrução
42
43      EscInstLCD(0x0C);            //desativa cursor
44      while(TesteBusyFlag());      //espera LCD controller terminar de executar instrução
45      while(1);                    //loop infinito
46  }
47  /*********************************************************************
48  Esta funcao inicializa os resgistradores SFRs.*/
49   void Inic_Regs (void)
50  {
51      TRISA = 0x00;          //PORTA saída
52      TRISB = 0x00;          //PORTB saída
53      TRISC = 0x00;          //PORTC saída
54      TRISD = 0x00;          //PORTD saída
55      TRISE = 0x00;          //PORTE saída
56      ADCON1 = 0x0F;          //configura pinos dos PORTA e PORTE como digitais
57      PORTA = 0;             //limpa PORTA
58      PORTB = 0;             //limpa PORTB
59      PORTC = 0;             //limpa PORTC
60      PORTD = 0x00;          //apaga displays
61      PORTE = 0;             //limpa PORTE
62  }
```

Display LCD

- Esquema Elétrico de conexão de um LCD com um uC da placa RF_Explorer;





Display LCD

- No exemplo de demonstração da placa RF_Explorer pode ser encontrado um código-fonte para um Display LCD utilizando 4 bits.

Display LCD – Placa RF_Explorer - 1

```
1  #include <p18f4520.h>
2  #include <delays.h>
3  // The lower 4 bits of this port will be used
4  #define LCD_DATA      LATD            // as RF Explorer Smart Radio Board
5  #define LCD_EN          LATDbits.LATD4
6  #define LCD_RS          LATDbits.LATD5
7  #define TRIS_LCD_DATA   TRISD
8  #define TRIS_LCD_EN    TRISDbits.TRISD4
9  #define TRIS_LCD_RS    TRISDbits.TRISD5
10
11 // Commands for Hitachi LCD
12 #define CLEAR_DISPLAY   0x01
13 #define FOUR_BIT      0b00101111  /* 4-bit Interface               */
14 #define EIGHT_BIT     0b00111111  /* 8-bit Interface               */
15 #define LINE_5X7      0b00110011  /* 5x7 characters, single line   */
16 #define LINE_5X10     0b00110111  /* 5x10 characters               */
17 #define LINES_5X7     0b00111011  /* 5x7 characters, multiple line */
18
19 #define DISPLAY_ON  0b00001111  /* Display on      */
20 #define DISPLAY_OFF 0b00001011  /* Display off     */
21 #define CURSOR_ON   0b00001111  /* Cursor on       */
22 #define CURSOR_OFF  0b00001101  /* Cursor off      */
23 #define BLINK_ON    0b00001111  /* Cursor Blink    */
24 #define BLINK_OFF   0b00001110  /* Cursor No Blink */
25 #define UNDERLINE_ON    0b00001111
26 #define UNDERLINE_OFF   0b00001101
27
28 #define INCREMENT   0b00000111   /* Entry mode increment */
29 #define DECREMENT   0b00000101   /* Entry mode decrement */
30 #define SHIFT_ON   0b00000111   /* Display shift on      */
31 #define SHIFT_OFF   0b00000110   /* Display shift off   */
32 #define PulseEnable { LCD_EN = 1; Nop(); LCD_EN = 0; }
```

Display LCD – Placa RF_Explorer - 2

```
34  void LCDInit(void);
35  void LCDCmd(unsigned char);
36  void LCDChar(unsigned char);
37  // TODO: Add all prototypes
38  void Delay5ms(void) {
39      Delay1KTCYx(60); // Delay of 5ms
40      // Cycles = (TimeDelay * Fosc) / 4
41      // Cycles = (5ms * 8MHz) / 4
42      // Cycles = 10,000
43      return;
44  }
45
46  void Delay15ms(void) {
47      Delay1KTCYx(180); // Delay of 15ms
48      // Cycles = (TimeDelay * Fosc) / 4
49      // Cycles = (15ms * 8MHz) / 4
50      // Cycles = 30,000
51      return;
52  }
53
54  void LCDInit(void) {
55      // Set port directions
56      TRIS_LCD_DATA &= 0xf0;    // Clear lower 4 bits
57      TRIS_LCD_EN = 0;
58       TRIS_LCD_RS = 0;
59
60      //LCDCmd(FOUR_BIT);          // Nigel's code sends this befo
61      LCDCmd(FOUR_BIT & LINES_5X7);
62      LCDCmd(INCREMENT & SHIFT_OFF);
63      LCDCmd(DISPLAY_ON & BLINK_OFF & UNDERLINE_OFF);
64      LCDCmd(CLEAR_DISPLAY);
65  }
```

Display LCD – Placa RF_Explorer - 3

```
67  void LCDCmd(unsigned char c) {
68      LCD_RS = 0;
69
70      LCD_DATA &= 0xf0;
71      LCD_DATA |= (c >> 4);
72      PulseEnable;
73
74      LCD_DATA &= 0xf0;
75      LCD_DATA |= (c & 0x0f);
76      PulseEnable;
77
78      Delay5ms();
79  }
80
81  void LCDChar(unsigned char c) {
82      LCD_RS = 1;
83
84      LCD_DATA &= 0xf0;
85      LCD_DATA |= (c >> 4);
86      PulseEnable;
87
88      LCD_DATA &= 0xf0;
89      LCD_DATA |= (c & 0x0f);
90      PulseEnable;
91
92      Delay5ms();
93  }
```

Display LCD – Placa RF_Explorer - 4

```
95  void Inicio_2Linha (void)
96  {
97  LCDCmd (0xC0);
98  }
99
100
101
102 void LCD_Welcome (void) {
103
104 unsigned char i;
105    TRISD = 0x00; //LCD Port
106
107    for (i = 0; i < 2 ; i++)
108       {
109       Delay15ms();
110       }
```

Display LCD – Placa RF_Explorer - 5

```
112     LCDInit();
113
114     LCDChar(' ');
115     LCDChar(' ');
116     LCDChar(' ');
117     LCDChar('R');
118     LCDChar('F');
119     LCDChar(' ');
120     LCDChar('E');
121     LCDChar('x');
122     LCDChar('p');
123     LCDChar('l');
124     LCDChar('o');
125     LCDChar('r');
126     LCDChar('e');
127     LCDChar('r');
128     Inicio_2Linha (); //    Move to  2nd line
129     LCDChar(' ');
130     LCDChar(' ');
131     LCDChar(' ');
132     LCDChar('S');
133     LCDChar('m');
134     LCDChar('a');
135     LCDChar('r');
136     LCDChar('t');
137     LCDChar(' ');
138     LCDChar('R');
139     LCDChar('a');
140     LCDChar('d');
141     LCDChar('i');
142     LCDChar('o');
143 }
```

Próxima Aula

Aula 13

Módulos de Suporte à CPU