



FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA
ENSINANDO E APRENDENDO

Aula 11

Portas de Entrada e Saída – Parte III

(Displays de 7 seg. Multiplexados)

Microcontroladores PIC18 – Programação em C



Prof. Ítalo Jáder Loiola Batista

Universidade de Fortaleza - UNIFOR

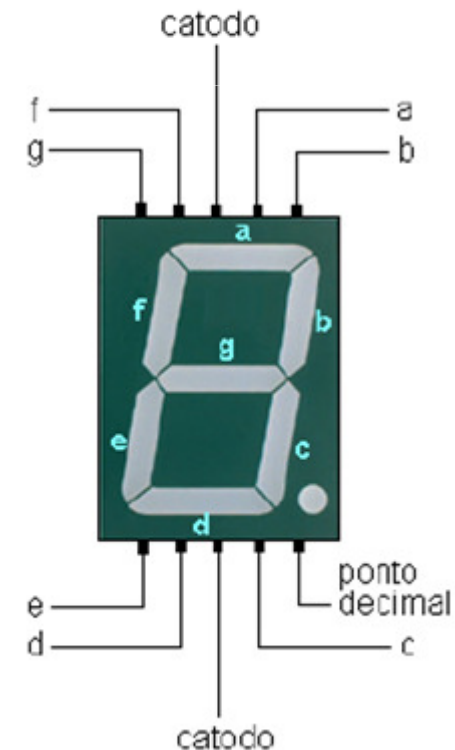
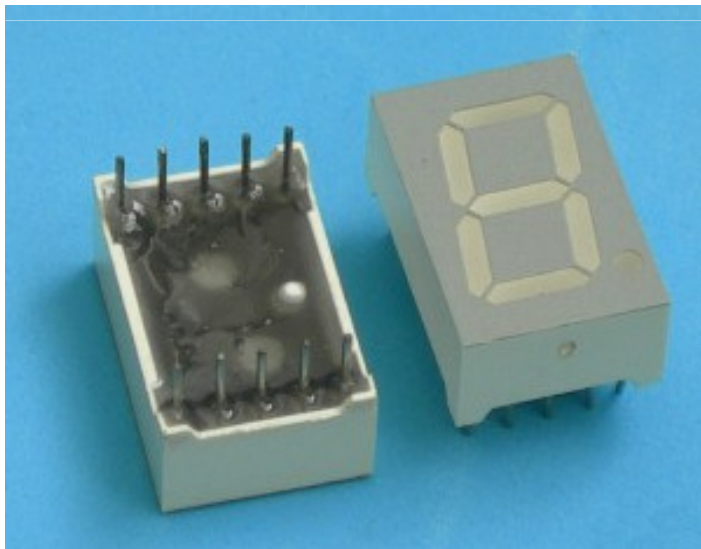
Centro de Ciências Tecnológicas - CCT

E-mail: italoloiola@unifor.br

Jan/2011

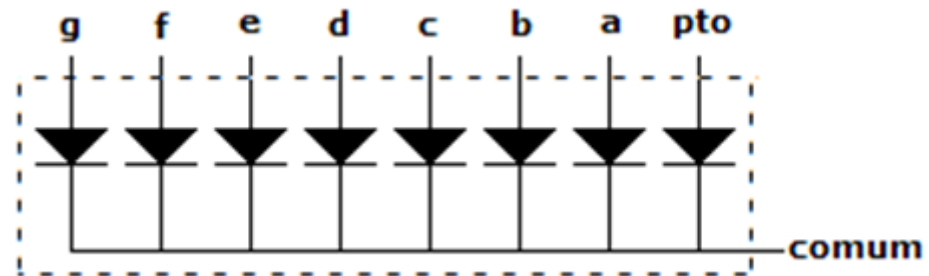
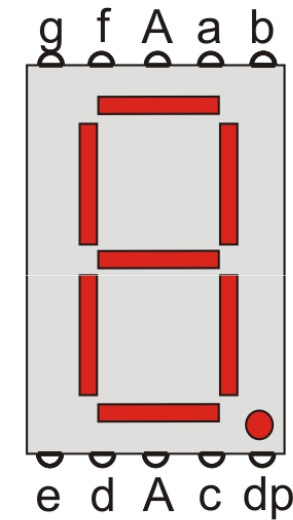
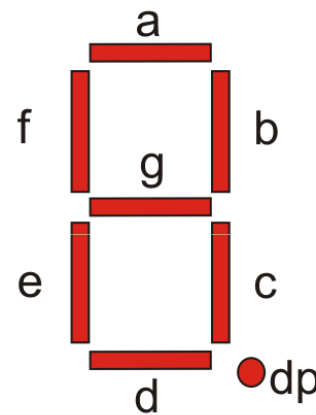
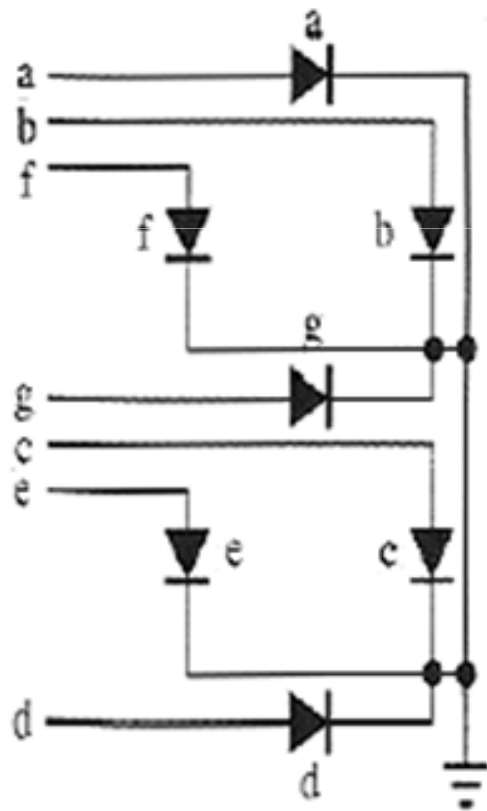
Display de Sete Segmentos

- O display de sete segmentos é um invólucro com sete leds com formato de segmento, posicionados de modo a possibilitar a formação de números decimais e algumas letras utilizadas no código hexadecimal.



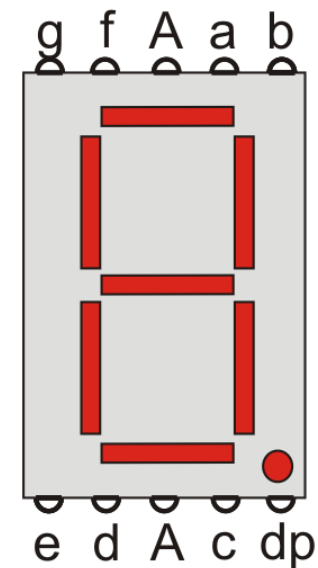
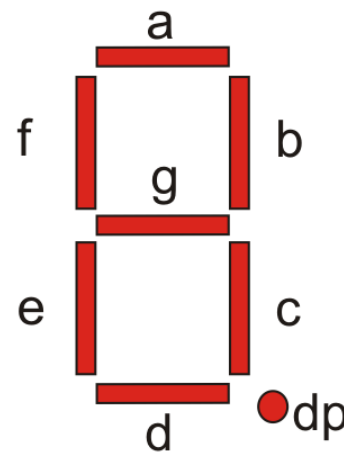
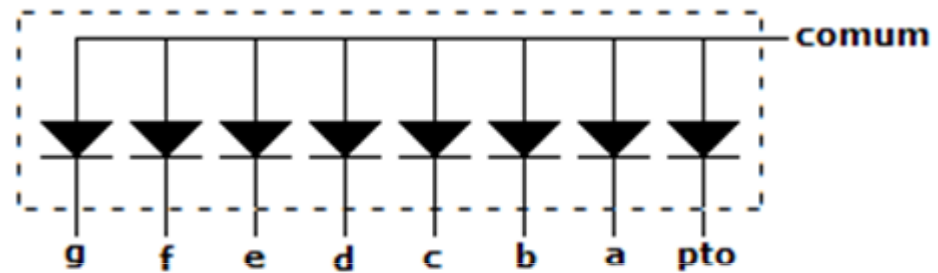
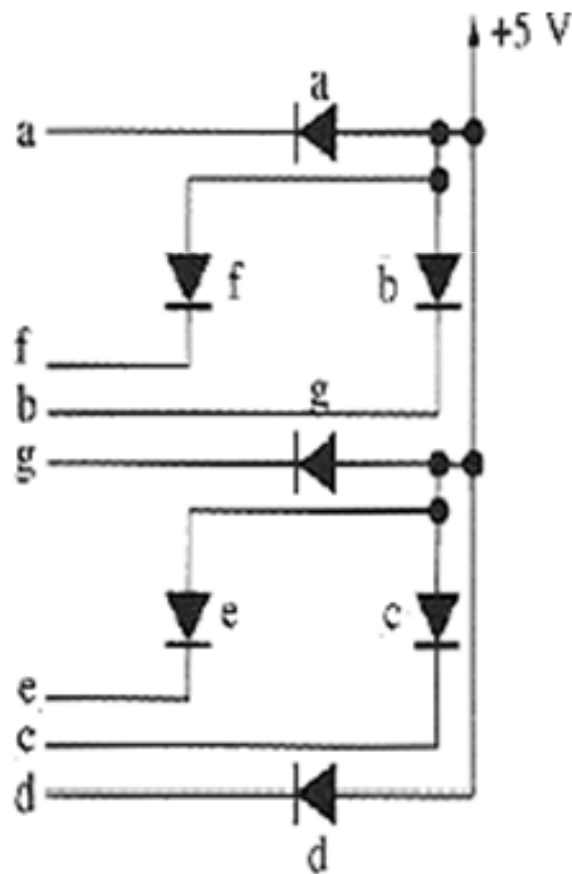
Display de Sete Segmentos

- Display **catodo** comum;



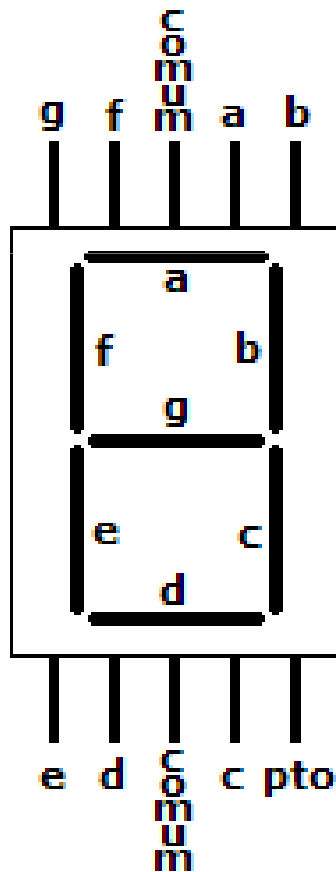
Display de Sete Segmentos

- Display **anodo** comum;



Display de Sete Segmentos

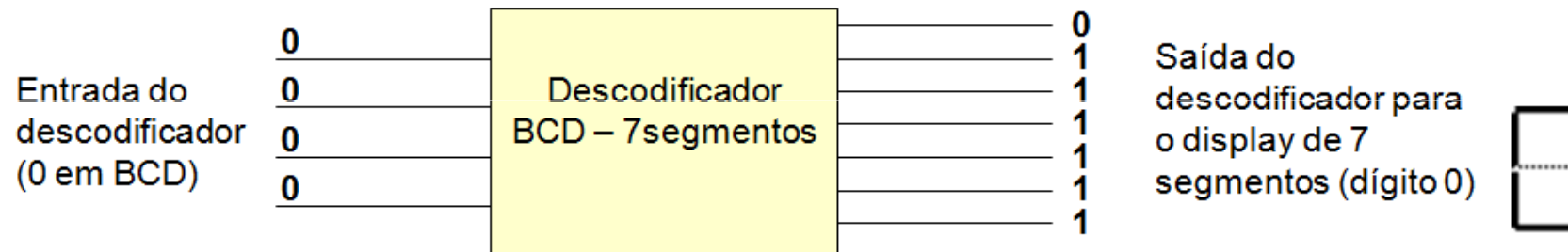
- Acionamento do Display **catodo** comum;



segmentos de saída							DISPLAY
a	b	c	d	e	f	g	
1	1	1	1	1	1	0	0
0	1	1	0	0	0	0	1
1	1	0	1	1	0	1	2
1	1	1	1	0	0	1	3
0	1	1	0	0	1	1	4
1	0	1	1	0	1	1	5
0	0	1	1	1	1	1	6
1	1	1	0	0	0	0	7
1	1	1	1	1	1	1	8
1	1	1	0	0	1	1	9

Display de Sete Segmentos

- O decodificador **BCD-7segmentos** tem a função de interpretar um código (BCD) e gerar os sinais para ligar o dígito correspondente a este código no display de 7 segmentos.

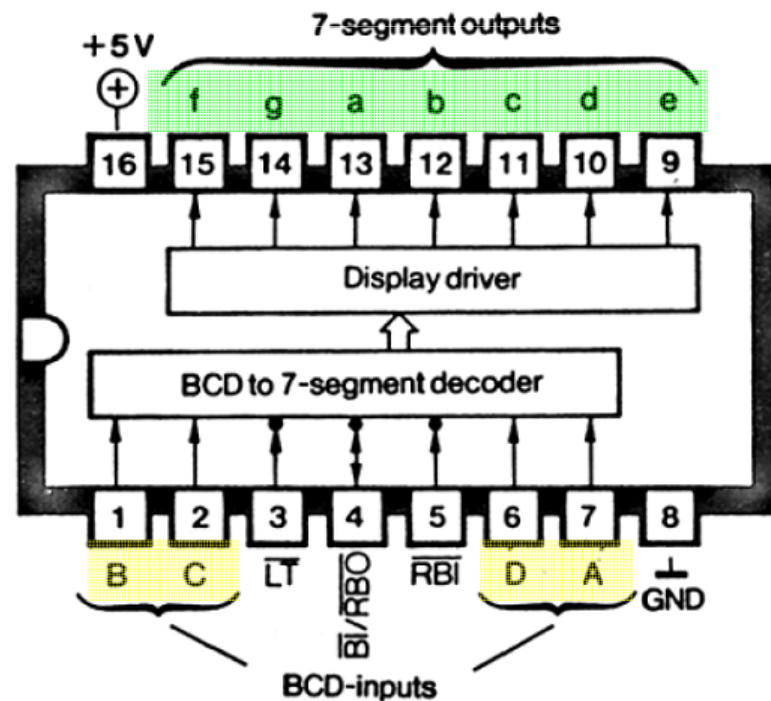
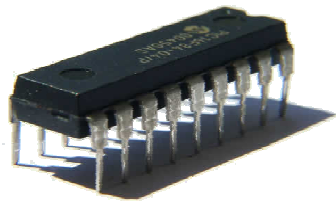


- Os decodificadores comerciais disponíveis são:
 - 7447 (ânodo comum)
 - 7448 (cátodo comum).

Display de Sete Segmentos

Descodificador BCD – 7 segmentos

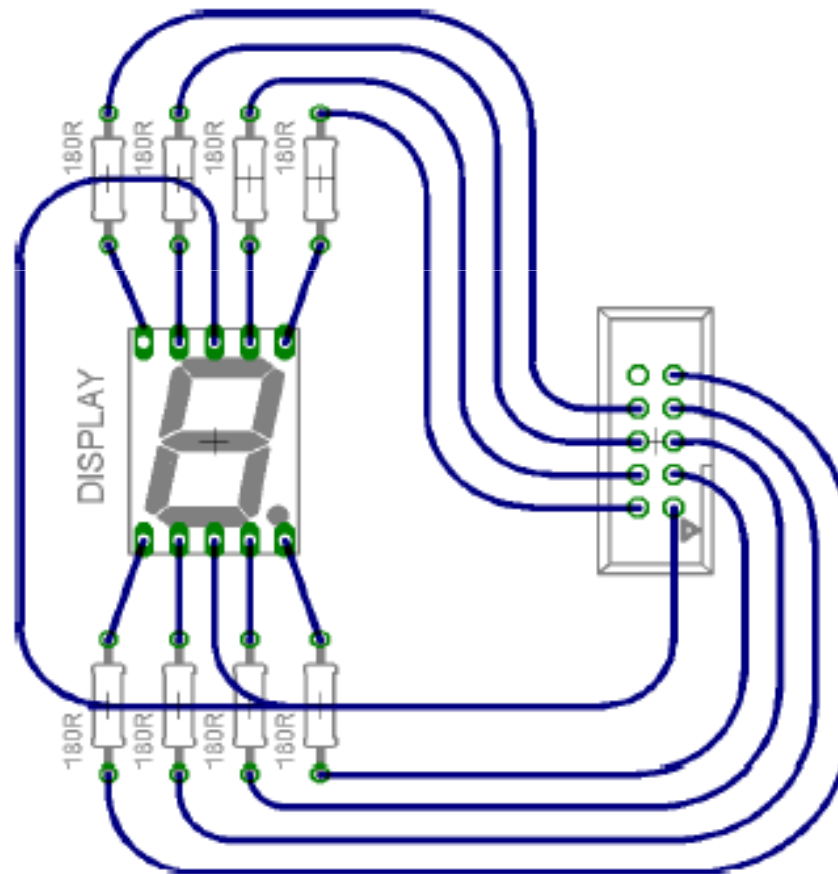
Circuito integrado 7448



entradas BCD				segmentos de saída							DISPLAY
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	0	0	1	1	9

Display de Sete Segmentos

- Teste do Display;



Display de Sete Segmentos

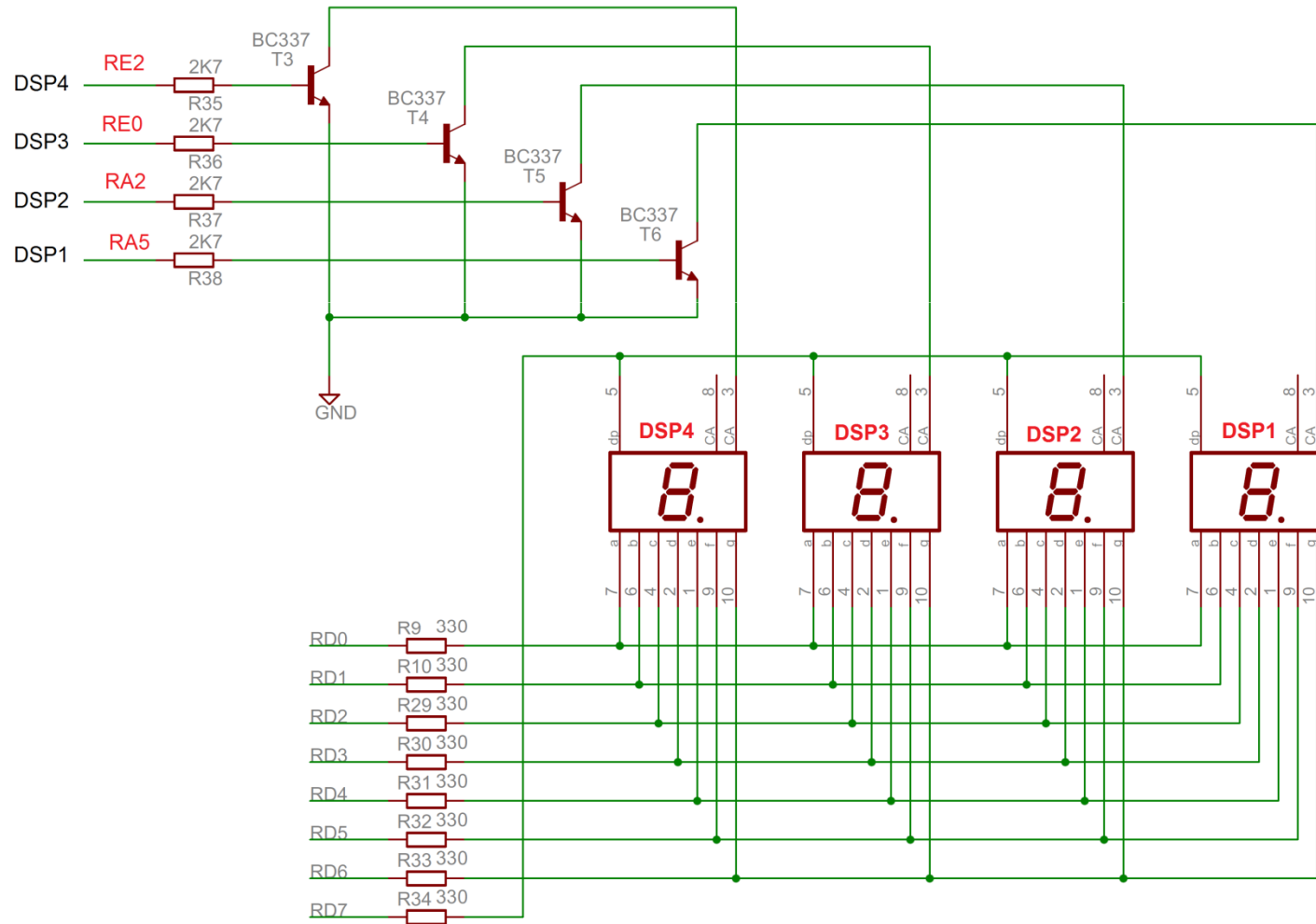
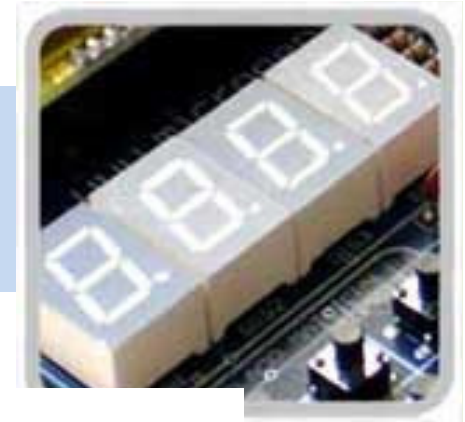
- Interfaceamento de um display **catodo** comum com o PIC18F4520;

Display de Sete Segmentos

- Interfaceamento de um display **anodo** comum com o PIC18F4520;

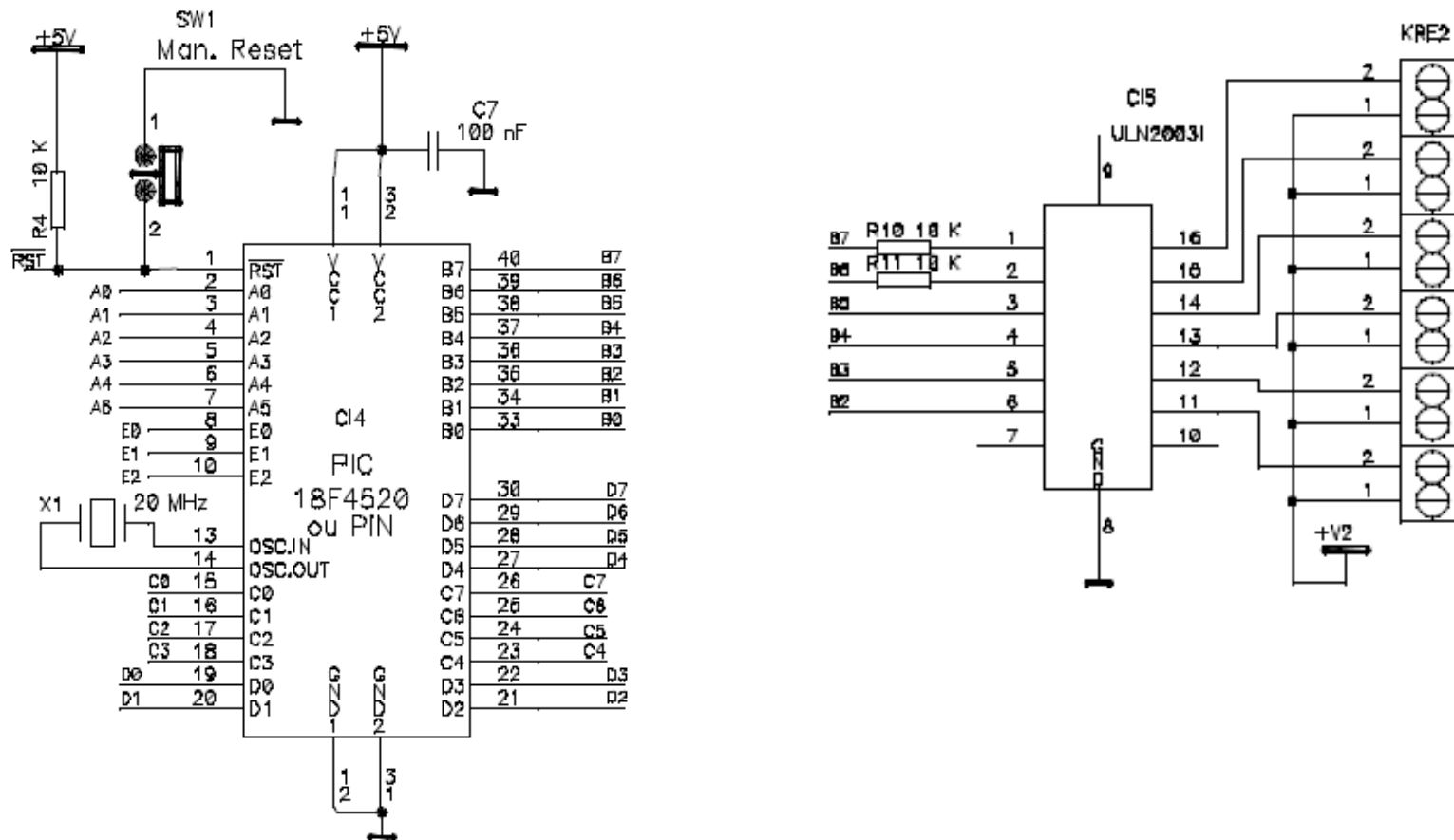
Display de Sete Segmentos

- Displays Multiplexados;



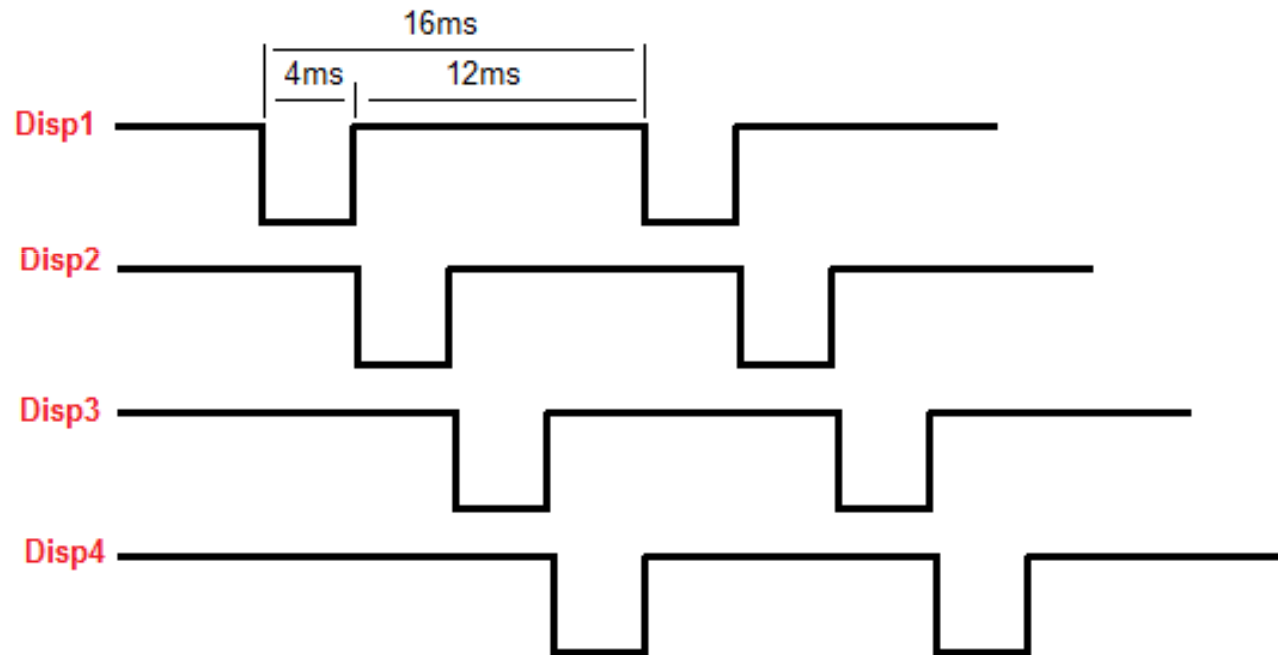
Display de Sete Segmentos

- Interfaceamento de quatro displays com o PIC18F4520 utilizando o driver ULN2803;



Temporização dos Displays

- Diagrama de tempo de acionamento dos displays



Display Multiplexado / Código-fonte

- **DSP_7Seg_x4.h**

Arquivo cabeçalho com as definições dos pinos nos quais serão conectados os pinos do display.

- **DSP_7Seg_x4.c**

Arquivo que compõe a biblioteca que contém a função que fará a atualização do display;

- **Main_31.c**

Arquivo principal responsável por realizar a aplicação de um contador de 0 a 9.999 utilizando uma interface de vídeo com quatro displays multiplexados;

Display Multiplexado / Código-fonte

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

```
8  #ifndef __DSP_7SEGx4_H
9  #define __DSP_7SEGx4_H
10 //*****
11 #include <p18cxxx.h> //diretiva de compilação
12 //*****
13 //definições
14 #define DSP_1    PORTAbits.RA5
15 #define DSP_2    PORTAbits.RA2
16 #define DSP_3    PORTEbits.RE0
17 #define DSP_4    PORTEbits.RE2
18 #define L_DADOS TRISD
19 #define DIR_A1   TRISAbits.TRISA5
20 #define DIR_A2   TRISAbits.TRISA2
21 #define DIR_A3   TRISEbits.TRISE0
22 #define DIR_A4   TRISEbits.TRISE2
23 //*****
24 void Aciona_DPS_7_seg (unsigned char Dsp4, unsigned char Dsp3, unsigned char Dsp2, unsigned char Dsp1);
25 #endif
```

DSP_7Seg_x4.c

Display Multiplexado / Código-fonte

```

8  #include <p18cxxx.h>           //diretiva de compilação
9  #include "DSP_7Seg_x4.h"       //diretiva de compilação
10 //*****
11 void Aciona_DPS_7_seg(unsigned char Dsp4, unsigned char Dsp3, unsigned char Dsp2, unsigned char Dsp1)
12 {
13     static unsigned char Atual_Dsp = 1;           //declaração de variável local static inicializa
14     const char tabela[] = {
15         0x3F, // número 0
16         0x06, // número 1
17         0x5B, // número 2
18         0x4F, // número 3
19         0x66, // número 4
20         0x6D, // número 5
21         0x7C, // número 6
22         0x07, // número 7
23         0x7F, // número 8
24         0x67, // número 9
25         0x00 //apaga display
26     };
27 //*****
28 //configuração dos pinos
29     L_DADOS = 0x00;           //configura pinos das linhas de dados como saída
30     ADCON1 = 0x0F;           //configura Port A e Port E como pinos digitais
31     DIR_A1 = 0;               //configura linha de endereço A1 como saída
32     DIR_A2 = 0;               //configura linha de endereço A2 como saída
33     DIR_A3 = 0;               //configura linha de endereço A3 como saída
34     DIR_A4 = 0;               //configura linha de endereço A4 como saída
35 //*****

```


DSP_7Seg_x4.c

Display Multiplexado / Código-fonte

```
36 //atualiza display
37 if (Atual_Dsp==1) //atualizar display 1
38 {
39     PORTD = tabela[Dsp1]; //atualiza display 1
40     DSP_1 = 1; //ativa linha A1
41     DSP_2 = 0; //desativa linha A2
42     DSP_3 = 0; //desativa linha A3
43     DSP_4 = 0; //desativa linha A4
44     Atual_Dsp = 2; //aponta endereço para o próximo display
45 }
46 else if (Atual_Dsp==2) //atualizar display 2
47 {
48     PORTD = tabela[Dsp2]; //atualiza display 2
49     DSP_1 = 0; //desativa linha A1
50     DSP_2 = 1; //ativa linha A2
51     DSP_3 = 0; //desativa linha A3
52     DSP_4 = 0; //desativa linha A4
53     Atual_Dsp = 3; //aponta endereço para o próximo display
54 }
```

Display Multiplexado / Código-fonte

```
55     else if (Atual_Dsp==3) //atualizar display 3
56     {
57         PORTD = tabela[Dsp3]; //atualiza display 3
58         DSP_1 = 0;           //desativa linha A1
59         DSP_2 = 0;           //desativa linha A2
60         DSP_3 = 1;           //ativa linha A3
61         DSP_4 = 0;           //desativa linha A4
62         Atual_Dsp = 4;       //aponta endereço para o próximo display
63     }
64     else if (Atual_Dsp==4) //atualizar display 4
65     {
66         PORTD = tabela[Dsp4]; //atualiza display 4
67         DSP_1 = 0;           //desativa linha A1
68         DSP_2 = 0;           //desativa linha A2
69         DSP_3 = 0;           //desativa linha A3
70         DSP_4 = 1;           //ativa linha A4
71         Atual_Dsp = 1;       //aponta endereço para o próximo display
72     }
73     //*****
74 }
```

Main_31.c

Display Multiplexado / Código-fonte

```
8  #include <p18f4520.h>      //diretiva de compilação
9  #include "DSP_7Seg_x4.h"   //diretiva de compilação
10 #include <delays.h>        //diretiva de compilação
11 //*****
12 //protótipos de funções
13 void Inic_Regs (void);
14 //*****
15 void main(void)             //função main
16 {
17     unsigned char Dsp1=0;    //declaração de variável local inicializada
18     unsigned char Dsp2=0;    //declaração de variável local inicializada
19     unsigned char Dsp3=0;    //declaração de variável local inicializada
20     unsigned char Dsp4=0;    //declaração de variável local inicializada
21     unsigned char x=0;
22     Inic_Regs ();            //chamada a função: configurar SFRs
```

Main_31.c

Display Multiplexado / Código-fonte

```

23 while(1)
24 {
25     Aciona_DPS_7_seg (Dsp4, Dsp3, Dsp2, Dsp1); //chamada à função: atualizar
26     Delay1KTCYx(8); //delay de 4ms
27     if(! (x==250)) x++; //se x diferente de 250, incrementa
28     else //senão, incrementa contador
29     {
30         x=0; //x=0
31         Dsp1+=1; //incrementa unidade
32         if (Dsp1==10) //unidade estourou?
33         {
34             Dsp1=0; //sim, zera unidade
35             Dsp2+=1; //incrementa dezena
36         }
37         if (Dsp2==10) //dezena estourou?
38         {
39             Dsp2=0; //sim, zera dezena
40             Dsp3+=1; //incrementa centena
41         }
42         if (Dsp3==10) //centena estourou?
43         {
44             Dsp3=0; //sim, zera centena
45             Dsp4+=1; //incrementa unidade de milhar
46         }
47         if (Dsp4==10) //unidade de milhar estourou?
48         {
49             Dsp4=Dsp3=Dsp2=Dsp1=0; //sim, zera contador
50         }
51     }
52 }
53

```

Função para atualização dos displays, que é realizada com passagem dos valores a serem utilizados em cada display;

Verifica se passou um segundo para, então, o programa incrementar o contador;

Display Multiplexado / Código-fonte

```
55  Esta funcao inicializa os resgistradores SFRs.*/
56  void Inic_Regs (void)
57  {
58      TRISA = 0x00;          //PORTA saída
59      TRISB = 0x00;          //PORTB saída
60      TRISC = 0x00;          //PORTC saída
61      TRISD = 0x00;          //PORTD saída
62      TRISE = 0x00;          //PORTE saída
63      ADCON1 = 0x0F;         //configura pinos dos PORTA e PORTE como digitais
64      PORTA = 0;             //limpa PORTA
65      PORTB = 0;             //limpa PORTB
66      PORTC = 0;             //limpa PORTC
67      PORTD = 0x00;          //apaga displays
68      PORTE = 0;             //limpa PORTE
69  //*****
70  }
```

Display Multiplexado

- Na Aula 14 (Timers) será mostrado um programa mais otimizado para displays multiplexados utilizando o TIMER;

Próxima Aula

Aula 12

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