

```

1  #include<p18f452.h>
2  #include<delays.h>
3  #include<timers.h>
4  #include<spi.h>
5
6  #pragma config WDT = OFF, LVP = OFF, OSC = XT, PWRT = ON, BOR = ON, BORV = 42
7
8  //Definição de labels
9  #define SS PORTEbits.RE2
10
11 //Protótipos de funções
12 void inicia_lcd(void);           //Função de inicialização do LCD
13 void escreve_comando(char c);    //Envia comando para o LCD
14 void escreve_dado(char d);       //Escrita de uma dado no LCD
15
16 //Definição de variáveis globais
17 unsigned char decod[] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};
18 unsigned int SPIdt, ao_data = 0;
19
20 //-----
21 void main(void)
22 {
23     ADCON1 = 0x07;           // PORTA como pinos digitais
24
25     TRISB = 0b00001100;
26     //TRISC = 0b11010111;    //RC3:SCK, RC4:SDI, RC5:SDO
27     TRISD = 0x00;
28     TRISE = 0b00000000;
29
30     SS = 1;                  //Slave select do conversor D/A
31     PORTD = 0x00;
32
33     INTCON = 0;              //Desabilita interrupções
34
35     OpenSPI(SPI_FOSC_4, MODE_00, SMPMID);
36     inicia_lcd();
37
38     while(1)                //Loop principal
39     {
40         while((PORTBbits.RB2 == 1) && (PORTBbits.RB3 == 1))
41         {
42         }
43
44         Delay1KTCYx(20);     //Delay para debounce das teclas
45
46         if((PORTBbits.RB2 == 0) && (PORTBbits.RB3 == 1))
47         { //RB2 pressionado
48             if(ao_data < 4000) {ao_data += 250;}
49         }
50         else if((PORTBbits.RB2 == 1) && (PORTBbits.RB3 == 0))
51         { //RB3 pressionado
52             if(ao_data >= 250) {ao_data -= 250;}
53         }
54
55         SS = 0;
56         SPIdt = 0x3000 | ao_data;
57         WriteSPI(SPIdt>>8);
58         WriteSPI(0x00FF & SPIdt);
59         SS = 1;
60
61         while((PORTBbits.RB2 == 0) || (PORTBbits.RB3 == 0))
62         {
63             Delay1KTCYx(20);
64             while((PORTBbits.RB2 == 0) || (PORTBbits.RB3 == 0)){}
65         }
66     }
67 }
68
69 //-----
70 //Função de inicialização do LCD
71 void inicia_lcd(void)
72 {
73     escreve_comando(0x38);
74     Delay1KTCYx(3);
75     escreve_comando(0x38);

```

```

77     escreve_comando(0x06);
78     escreve_comando(0x0C);
79     escreve_comando(0x01);
80     Delay1KTCYx(1);
81     escreve_comando(0x80); //Posiciona cursor
82
83     escreve_dado('X');
84 }
85
86 //-----
87 //Envia comando para o LCD
88 void escreve_comando(char c)
89 {
90     PORTEbits.RE0=0;
91     PORTD=c;
92     Delay10TCYx(1);
93     PORTEbits.RE1=1;
94     Delay1TCY();
95     Delay1TCY();
96     Delay1TCY();
97     PORTEbits.RE1=0;
98     Delay1KTCYx(1);
99 }
100
101 //-----
102 //Escrita de uma dado no LCD
103 void escreve_dado(char d)
104 {
105     PORTEbits.RE0=1;
106     PORTD=d;
107     Delay10TCYx(1);
108     PORTEbits.RE1=1;
109     Delay1TCY();
110     Delay1TCY();
111     Delay1TCY();
112     PORTEbits.RE1=0;
113     Delay1KTCYx(1);
114     PORTEbits.RE0=1;
115 }

```