

```

1  // Firmware desenvolvido para o hardware da placa McLabII - comunicação serial
2  #include<p18f452.h>
3  #include<delays.h>
4  #include<usart.h>
5
6  #pragma config WDT = OFF, LVP = OFF, OSC = XT, PWRT = ON, BOR = ON, BORV = 42
7
8  //Protótipos de funções
9  void ISR_High_Priority(void);
10 void inicia_lcd(void); //Função de inicialização do LCD
11 void escreve_comando(char c); //Envia comando para o LCD
12 void escreve_dado(char d); //Escrita de uma dado no LCD
13
14 //Definição de variáveis globais
15 volatile unsigned char new_dt, rx_dt = 0;
16
17 //-----
18 #pragma code vec_int_high_priority = 0x08
19 void vec_int_high_priority(void)
20 { _asm GOTO ISR_High_Priority _endasm }
21
22 #pragma interrupt ISR_High_Priority
23 void ISR_High_Priority(void)
24 {
25     if(PIR1bits.RCIF)
26     {
27         rx_dt = ReadUSART();
28         new_dt = 1;
29         PIR1bits.RCIF = 0;
30     }
31 }
32
33 //-----
34 void main(void)
35 {
36     PORTB = 0x00;
37     PORTD = 0x00;
38
39     ADCON1 = 0x07; // PORTA como pinos digitais
40
41     TRISB = 0b00001111;
42     TRISC = 0b10111001; // RC7: Rx e RC6:Tx
43     TRISD = 0x00;
44     TRISE = 0b00000100;
45
46     OpenUSART( USART_TX_INT_OFF & // Int. de transmissão habilitada
47               USART_RX_INT_ON & // Int. de recepção habilitada
48               USART_ASYNC_MODE & // Comunicação assíncrona
49               USART_EIGHT_BIT &
50               USART_CONT_RX & // Recepção contínua
51               USART_BRGH_HIGH, 25 ); // Alta velocidade 9600bps
52
53     RCONbits.IPEN = 1; //Habilita prioridade de interrupções
54     IPR1bits.RCIP = 1; //Int. USART (RX) é de alta prioridade
55     INTCONbits.GIE = 1; //Habilita interrupções
56     INTCONbits.PEIE = 1; //Habilita interrupções de periféricos
57
58     PIR1bits.RCIF = 0; //Inicializa flag de interrupções
59     inicia_lcd();
60
61     while(1) //Loop principal
62     {
63         while((PORTB | 0xF0) == 0xFF)
64         {
65             if(new_dt)
66             {
67                 new_dt = 0;
68                 if(rx_dt != '0')
69                     escreve_dado(rx_dt);
70                 else
71                 {
72                     escreve_comando(0xC3);
73                     //for...
74                     escreve_dado('X');
75                     escreve_dado('X');
76                     escreve_dado('X');

```

```

77         }
78     }
79 }
80 Delay1KTCYx(20); //Delay para debounce das teclas
81 if((PORTB | 0xF0) == 0b1111110)
82     WriteUSART('0');
83 else if((PORTB | 0xF0) == 0b1111101)
84     WriteUSART('1');
85 else if((PORTB | 0xF0) == 0b1111011)
86     WriteUSART('2');
87 else if((PORTB | 0xF0) == 0b1110111)
88     WriteUSART('3');
89
90 while((PORTB | 0xF0) != 0xFF){}
91 Delay1KTCYx(30); //Delay para debounce das teclas
92 while((PORTB | 0xF0) != 0xFF){}
93 }
94
95 }
96
97 //-----
98 //Envia comando para o LCD
99 void escreve_comando(char c)
100 {
101     PORTEbits.RE0=0;
102     PORTD=c;
103     Delay10TCYx(1);
104     PORTEbits.RE1=1;
105     Delay1TCY();
106     Delay1TCY();
107     Delay1TCY();
108     PORTEbits.RE1=0;
109     Delay1KTCYx(1);
110 }
111
112 //-----
113 //Escrita de uma dado no LCD
114 void escreve_dado(char d)
115 {
116     PORTEbits.RE0=1;
117     PORTD=d;
118     Delay10TCYx(1);
119     PORTEbits.RE1=1;
120     Delay1TCY();
121     Delay1TCY();
122     Delay1TCY();
123     PORTEbits.RE1=0;
124     Delay1KTCYx(1);
125     PORTEbits.RE0=1;
126 }
127
128 //-----
129 //Função de inicialização do LCD
130 void inicia_lcd(void)
131 {
132     escreve_comando(0x38);
133     Delay1KTCYx(3);
134     escreve_comando(0x38);
135     escreve_comando(0x06);
136     escreve_comando(0x0C);
137     escreve_comando(0x01);
138     Delay1KTCYx(1);
139     escreve_comando(0x80); //Posiciona cursor
140
141     escreve_dado('.');
142     escreve_comando(0xC0);
143     escreve_dado('R');
144     escreve_dado('X');
145     escreve_dado(':');
146 }

```