

```

1  // Firmware desenvolvido para o hardware da placa McLabII
2  // I2C + comunicação serial
3  #include<pl8f452.h>
4  #include<delays.h>
5  #include<usart.h>
6  #include<i2c.h>
7
8  #pragma config WDT = OFF, LVP = OFF, OSC = XT, PWRT = ON, BOR = ON, BORV = 42
9
10 //Protótipos de funções
11 void ISR_High_Priority(void);
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 volatile unsigned char new_dt, I2Cdt, rx_dt = 0;
18
19 //-----
20 #pragma code vec_int_high_priority = 0x08
21 void vec_int_high_priority(void)
22 { _asm GOTO ISR_High_Priority _endasm }
23
24 #pragma interrupt ISR_High_Priority
25 void ISR_High_Priority(void)
26 {
27     if(PIR1bits.RCIF)
28     {
29         rx_dt = ReadUSART();
30         new_dt = 1;
31         PIR1bits.RCIF = 0;
32     }
33 }
34
35 //-----
36 void main(void)
37 {
38     PORTB = 0x00;
39     PORTD = 0x00;
40
41     ADCON1 = 0x07;          // PORTA como pinos digitais
42
43     TRISB = 0b00001111;
44     TRISC = 0b10111001;    // RC7: Rx e RC6:Tx
45     TRISD = 0x00;
46     TRISE = 0b00000100;
47
48     OpenUSART( USART_TX_INT_OFF &      // Int. de transmissão habilitada
49               USART_RX_INT_ON &        // Int. de recepção habilitada
50               USART_ASYNC_MODE &       // Comunicação assíncrona
51               USART_EIGHT_BIT &
52               USART_CONT_RX &          // Recepção contínua
53               USART_BRGH_HIGH, 25 );   // Alta velocidade 9600bps
54
55     OpenI2C(MASTER, SLEW_OFF);          // I2C Master mode
56     SSPADD = 9;                         // Vel. transmissão I2C: 100KHz
57     SSPSTAT = 0x80;
58
59     RCONbits.IPEN = 1;                  //Habilita prioridade de interrupções
60     IPR1bits.RCIP = 1;                  //Int. USART (RX) é de alta prioridade
61     INTCONbits.GIE = 1;                  //Habilita interrupções
62     INTCONbits.PEIE = 1;                 //Habilita interrupções de periféricos
63
64     PIR1bits.RCIF = 0;                  //Inicializa flag de interrupções
65     inicia_lcd();
66
67     while(1)    //Loop principal
68     {
69         while((PORTB | 0xF0) == 0xFF)
70         {
71             if(new_dt)
72             {
73                 new_dt = 0;
74                 if(rx_dt != '0')
75                 {
76                     escreve_comando(0xC4);

```

```

77         escreve_dado(rx_dt);
78     }
79 }
80
81 Delay1KTCYx(20);    //Delay para debounce das teclas
82 if((PORTB | 0xF0) == 0b11111110)
83     WriteUSART('X');
84 else if((PORTB | 0xF0) == 0b11111101)
85 {
86     IdleI2C();
87     StartI2C();
88     while(SSPCON2bits.SEN){};
89     if(!WriteI2C(0xA0));
90     {
91         if(!WriteI2C(rx_dt))
92         {
93             if(WriteI2C('b'))
94                 WriteUSART('*');
95             else
96                 WriteUSART('b');
97         }
98     }
99     StopI2C();
100 }
101 else if((PORTB | 0xF0) == 0b11111011)
102 {
103     IdleI2C();
104     StartI2C();
105     while(SSPCON2bits.SEN){};
106     if(!WriteI2C(0xA0))
107     {
108         if(!WriteI2C(rx_dt))
109         {
110             RestartI2C();
111             while(SSPCON2bits.SEN){};
112             if(!WriteI2C(0xA1))
113             {
114                 I2Cdt = ReadI2C();
115                 WriteUSART(I2Cdt);
116             }
117             else
118                 WriteUSART('#');
119         }
120     }
121     StopI2C();
122 }
123 else if((PORTB | 0xF0) == 0b11110111)
124     WriteUSART('Z');
125
126 while((PORTB | 0xF0) != 0xFF){}
127 Delay1KTCYx(30);    //Delay para debounce das teclas
128 while((PORTB | 0xF0) != 0xFF){}
129 }
130 }
131 }
132
133 //-----
134 //Envia comando para o LCD
135 void escreve_comando(char c)
136 {
137     PORTEbits.RE0=0;
138     PORTD=c;
139     Delay10TCYx(1);
140     PORTEbits.RE1=1;
141     Delay1TCY();
142     Delay1TCY();
143     Delay1TCY();
144     PORTEbits.RE1=0;
145     Delay1KTCYx(1);
146 }
147
148 //-----
149 //Escrita de uma dado no LCD
150 void escreve_dado(char d)
151 {
152     PORTEbits.RE0=1;

```

```

153 PORTD=d;
154 Delay10TCYx(1);
155 PORTEbits.RE1=1;
156 Delay1TCY();
157 Delay1TCY();
158 Delay1TCY();
159 PORTEbits.RE1=0;
160 Delay1KTCYx(1);
161 PORTEbits.RE0=1;
162 }
163
164 //-----
165 //Função de inicialização do LCD
166 void inicia_lcd(void)
167 {
168     escreve_comando(0x38);
169     Delay1KTCYx(3);
170     escreve_comando(0x38);
171     escreve_comando(0x06);
172     escreve_comando(0x0C);
173     escreve_comando(0x01);
174     Delay1KTCYx(1);
175     escreve_comando(0x80); //Posiciona cursor
176
177     escreve_dado('.');
178     escreve_comando(0xC0);
179     escreve_dado('E');
180     escreve_dado('n');
181     escreve_dado('d');
182     escreve_dado(':');
183 }

```