

```

1  // Firmware desenvolvido para o hardware da placa McLabII - comunicação serial
2  #include<p18f452.h>
3  #include<delays.h>
4  #include<adc.h>
5  #include<usart.h>
6  #include<timers.h>
7  #include<stdlib.h>
8
9  #pragma config LVP = OFF, OSC = XT, PWRT = ON, BOR = ON, BORV = 42
10 #pragma config CCP2MUX = ON
11 #pragma config WDT = OFF
12
13 //Protótipos de funções
14 void ISR_High_Priority(void);
15 void inicia_lcd(void); //Função de inicialização do LCD
16 void escreve_comando(char c); //Envia comando para o LCD
17 void escreve_dado(char d); //Escrita de uma dado no LCD
18
19 //Definição de variáveis globais
20 unsigned char temp, rx_dt, new_rx = 0;
21 unsigned char new_tx = 0;
22 unsigned char tx_dt[] = {'0', '0', '0'};
23 const rom char convtemp[] = {
24     0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
25     0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
26     2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9,
27     10, 10, 11, 11, 12, 12, 13, 13, 14, 14, 15, 15, 16, 16, 17, 17,
28     18, 18, 19, 19, 20, 20, 21, 21, 22, 22, 23, 23, 24, 24, 25,
29     25, 26, 26, 27, 27, 28, 28, 29, 29, 30, 30, 31, 31, 32, 32, 33,
30     33, 34, 34, 35, 35, 36, 36, 37, 37, 38, 38, 39, 39, 40, 40, 41,
31     41, 42, 42, 43, 43, 44, 44, 45, 45, 46, 46, 47, 47, 48, 48, 49,
32     49, 50, 50, 51, 51, 52, 52, 53, 53, 54, 54, 55, 55, 56, 56, 57,
33     57, 58, 58, 59, 59, 60, 60, 61, 61, 62, 62, 63, 63, 64, 64, 65,
34     65, 66, 66, 67, 67, 68, 68, 69, 69, 70, 70, 71, 71, 72, 72, 73,
35     73, 74, 74, 75, 75, 76, 76, 77, 77, 78, 78, 79, 79, 80, 80, 81,
36     81, 82, 82, 83, 83, 84, 84, 85, 85, 86, 86, 87, 87, 88, 88, 89,
37     89, 90, 90, 91, 91, 92, 92, 93, 93, 94, 94, 95, 95, 96, 96, 97,
38     97, 98, 98, 99, 99, 100, 100, 101, 101, 102, 102, 103, 103, 104, 104, 104,
39     105, 105, 106, 106, 107, 107, 108, 108, 109, 109, 110, 110, 111, 111, 112, 112};
40
41 //-----
42 #pragma code vec_int_high_priority = 0x08
43 void vec_int_high_priority(void)
44 { _asm GOTO ISR_High_Priority _endasm }
45
46 #pragma interrupt ISR_High_Priority
47 void ISR_High_Priority(void)
48 {
49     if(PIR1bits.RCIF)
50     {
51         rx_dt = ReadUSART();
52         new_rx = 1;
53         PIR1bits.RCIF = 0;
54     }
55     if(INTCONbits.TMR0IF)
56     {
57         ConvertADC(); //Inicia conversão
58         WriteTimer0(65536-31250);
59         INTCONbits.TMR0IF = 0;
60         while(BusyADC()); //Aguarda conversão A/D
61         temp = convtemp[ADRESH];
62         new_tx = 1;
63     }
64 }
65
66 //-----
67 void main(void)
68 {
69     ADCON1 = 0x07; // PORTA como pinos digitais
70
71     TRISA = 0b00100001; // RA0 e RA5 como entrada
72     TRISB = 0b00001111;
73     TRISC = 0b10000001; // RC7: Rx e RC6:Tx
74     TRISD = 0x00;
75     TRISE = 0b00000100;
76

```

```

77     OpenADC( ADC_FOSC_32 & ADC_LEFT_JUST &
78             ADC_1ANA_0REF, ADC_CH0 &
79             ADC_INT_OFF );
80
81     OpenUSART( USART_TX_INT_OFF &           // Int. de transmissão habilitada
82               USART_RX_INT_ON &             // Int. de recepção habilitada
83               USART_ASYNC_MODE &           // Comunicação assíncrona
84               USART_EIGHT_BIT &
85               USART_CONT_RX &              // Recepção contínua
86               USART_BRGH_HIGH, 12 );       // Alta velocidade 19200bps
87     IPR1bits.RCIP = 1;                     //Int. USART (RX) é de alta prioridade
88     PIR1bits.RCIF = 0;                     //Inicializa flag de interrupções
89
90     //Timer0
91     OpenTimer0(TIMER_INT_ON & T0_16BIT & T0_SOURCE_INT & T0_PS_1_32);
92     INTCON2bits.TMR0IP = 1; //Int. de timer0 como de alta prioridade
93     WriteTimer0(65536-31250);
94     INTCONbits.TMR0IE = 1;
95     INTCONbits.TMR0IF = 0;
96
97     //Timer2 (PWM)
98     OpenTimer2(TIMER_INT_OFF & T2_PS_1_1);
99
100    RCONbits.IPEN = 1;                      //Habilita prioridade de interrupções
101    INTCONbits.GIE = 1;                     //Habilita interrupções
102    INTCONbits.PEIE = 1;                    //Habilita interrupções de periféricos
103
104    inicia_lcd();
105    PORTC = 0b00000100;
106
107    while(1) //Loop principal
108    {
109        while((PORTB | 0xF0) == 0xFF)
110        {
111            if(new_rx)
112            {
113                new_rx = 0;
114                if(rx_dt != '0')
115                    escreve_dado(rx_dt);
116                else
117                {
118                    escreve_comando(0xC3);
119                    //for...
120                    escreve_dado('X');
121                    escreve_dado('X');
122                    escreve_dado('X');
123                }
124            }
125            if(new_tx)
126            {
127                itoa(temp,tx_dt);
128
129                if(temp > 99)
130                {
131                    WriteUSART(tx_dt[0]);
132                    Delay10TCYx(10);
133                    WriteUSART(tx_dt[1]);
134                    Delay100TCYx(10);
135                    WriteUSART(tx_dt[2]);
136                }
137                else if(temp > 9)
138                {
139                    WriteUSART('0');
140                    Delay100TCYx(10);
141                    WriteUSART(tx_dt[0]);
142                    Delay100TCYx(10);
143                    WriteUSART(tx_dt[1]);
144                }
145                else
146                {
147                    WriteUSART('0');
148                    Delay100TCYx(10);
149                    WriteUSART('0');
150                    Delay100TCYx(10);
151                    WriteUSART(tx_dt[0]);
152                }

```

```

153         new_tx = 0;
154     }
155
156     if(temp > 85)
157         PORTCbits.RC2 = 0;
158 }
159 Delay1KTCYx(20); //Delay para debounce das teclas
160 if((PORTB | 0xF0) == 0b11111110)
161     WriteUSART('0');
162 else if((PORTB | 0xF0) == 0b11111101)
163     WriteUSART('1');
164 else if((PORTB | 0xF0) == 0b11110111)
165     WriteUSART('2');
166 else if((PORTB | 0xF0) == 0b11110111)
167     WriteUSART('3');
168
169 while((PORTB | 0xF0) != 0xFF){}
170 Delay1KTCYx(30); //Delay para debounce das teclas
171 while((PORTB | 0xF0) != 0xFF){}
172 }
173
174 }
175
176 //-----
177 //Envia comando para o LCD
178 void escreve_comando(char c)
179 {
180     PORTEbits.RE0=0;
181     PORTD=c;
182     Delay10TCYx(1);
183     PORTEbits.RE1=1;
184     Delay1TCY();
185     Delay1TCY();
186     Delay1TCY();
187     PORTEbits.RE1=0;
188     Delay1KTCYx(1);
189 }
190
191 //-----
192 //Escrita de uma dado no LCD
193 void escreve_dado(char d)
194 {
195     PORTEbits.RE0=1;
196     PORTD=d;
197     Delay10TCYx(1);
198     PORTEbits.RE1=1;
199     Delay1TCY();
200     Delay1TCY();
201     Delay1TCY();
202     PORTEbits.RE1=0;
203     Delay1KTCYx(1);
204     PORTEbits.RE0=1;
205 }
206
207 //-----
208 //Função de inicialização do LCD
209 void inicia_lcd(void)
210 {
211     escreve_comando(0x38);
212     Delay1KTCYx(3);
213     escreve_comando(0x38);
214     escreve_comando(0x06);
215     escreve_comando(0x0C);
216     escreve_comando(0x01);
217     Delay1KTCYx(1);
218     escreve_comando(0x80); //Posiciona cursor
219
220     escreve_dado('.');
221     escreve_comando(0xC0);
222     escreve_dado('R');
223     escreve_dado('X');
224     escreve_dado(':');
225 }

```