

```

1 // Básico para o LCD funcionar
2 // P3.0 = SDA e P3.1=SCL
3
4 #define BR100K 10 //SCL operando em 100 kHz (SMCLK)
5 #define BR10K 21 //SCL operando em 50 kHz (SMCLK)
6 #define BR10K 105 //SCL operando em 10 kHz (SMCLK)
7
8 // Dois possíveis endereços do PCF8574
9 #define PCF_ADR 0x27
10 // #define PCF_ADR 0x3F
11
12
13 // Protótipo das funções
14 void config_I2C(void);
15 void LCD_inic(void);
16 void LCD_aux(char dado);
17 void PCF_STT_STP(void);
18 int PCF_read(void);
19 void PCF_write(char dado);
20 void delay(long limite);
21
22
23 void main(void){
24     config_I2C();
25     LCD_inic();
26     //LCD pronto para usar
27     ...
28 }
29
30
31
32 // Configurar Pinos I2C - UCSB0
33 // P3.0 = SDA e P3.1=SCL
34 void config_I2C(void){
35     P3SEL |= BIT1 | BIT0; // Use dedicated module
36     //P3REN |= BIT1 | BIT0; // Resistor enable
37     //P3OUT |= BIT1 | BIT0; // Pull-up
38
39     UCB0CTL1 |= UCSWRST; // UCSI B0 em ressete
40
41     UCB0CTL0 = UCSYNC | //Síncrono
42               UCMODE_3 | //Modo I2C
43               UCMST; //Mestre
44
45     UCB0BRW = BR100K; //100 kbps
46     //UCB0BRW = BR50K; // 20 kbps
47     //UCB0BRW = BR10K; // 10 kbps
48     UCB0CTL1 = UCSSEL_2; //SMCLK e remove ressete
49 }
50
51
52
53 // Inicializar LCD modo 4 bits
54 void LCD_inic(void){
55
56     PCF_STT_STP(); //Colocar PCF em estado conhecido
57
58     // Preparar I2C para operar
59     UCB0I2CSA = PCF_ADR; //Endereço Escravo
60     UCB0CTL1 |= UCTR | //Mestre TX
61               UCTXSTT; //Gerar START
62     while ( (UCB0IFG & UCTXIFG) == 0); //Esperar TXIFG=1
63     UCB0TXBUF = 0; //Saída PCF = 0;
64     while ( (UCB0CTL1 & UCTXSTT) == UCTXSTT); //Esperar STT=0
65     if ( (UCB0IFG & UCNACKIFG) == UCNACKIFG) //NACK?
66         while(1);
67
68     // Começar inicialização
69     LCD_aux(0); //RS=RW=0, BL=1
70     delay(20000);
71     LCD_aux(3); //3
72     delay(10000);
73     LCD_aux(3); //3

```

```

74     delay(10000);
75     LCD_aux(3);        //3
76     delay(10000);
77     LCD_aux(2);        //2
78
79     // Entrou em modo 4 bits
80     LCD_aux(2);        LCD_aux(8);        //0x28
81     LCD_aux(0);        LCD_aux(8);        //0x08
82     LCD_aux(0);        LCD_aux(1);        //0x01
83     LCD_aux(0);        LCD_aux(6);        //0x06
84     LCD_aux(0);        LCD_aux(0xF);      //0x0F
85
86     while ( (UCB0IFG & UCTXIFG) == 0)    ;        //Esperar TXIFG=1
87     UCB0CTL1 |= UCTXSTP;                    //Gerar STOP
88     while ( (UCB0CTL1 & UCTXSTP) == UCTXSTP) ;    //Esperar STOP
89     delay(50);
90 }
91
92
93 // Auxiliar inicialização do LCD (RS=RW=0)
94 // *** Só serve para a inicialização ***
95 void LCD_aux(char dado){
96     while ( (UCB0IFG & UCTXIFG) == 0);        //Esperar TXIFG=1
97     UCB0TXBUF = ((dado<<4)&0xF0) | BIT_BL;    //PCF7:4 = dado;
98     delay(50);
99     while ( (UCB0IFG & UCTXIFG) == 0);        //Esperar TXIFG=1
100    UCB0TXBUF = ((dado<<4)&0xF0) | BIT_BL | BIT_E; //E=1
101    delay(50);
102    while ( (UCB0IFG & UCTXIFG) == 0);        //Esperar TXIFG=1
103    UCB0TXBUF = ((dado<<4)&0xF0) | BIT_BL;    //E=0;
104 }
105
106
107 // Gerar START e STOP para colocar PCF em estado conhecido
108 void PCF_STT_STP(void){
109     int x=0;
110     UCB0I2CSA = PCF_ADR;                    //Endereço Escravo
111
112     while (x<5){
113         UCB0CTL1 |= UCTR          |    //Mestre TX
114         UCTXSTT;                //Gerar START
115         while ( (UCB0IFG & UCTXIFG) == 0); //Esperar TXIFG=1
116         UCB0CTL1 |= UCTXSTP;        //Gerar STOP
117         delay(200);
118         if ( (UCB0CTL1 & UCTXSTP) == 0) break; //Esperar STOP
119         x++;
120     }
121     while ( (UCB0CTL1 & UCTXSTP) == UCTXSTP); //I2C Travado (Desligar / Ligar)
122 }
123
124
125 // Ler a porta do PCF
126 int PCF_read(void){
127     int dado;
128     UCB0I2CSA = PCF_ADR;                    //Endereço Escravo
129     UCB0CTL1 &= ~UCTR;                      //Mestre RX
130     UCB0CTL1 |= UCTXSTT;                    //Gerar START
131     while ( (UCB0CTL1 & UCTXSTT) == UCTXSTT);
132     UCB0CTL1 |= UCTXSTP;                    //Gerar STOP + NACK
133     while ( (UCB0IFG & UCRXIFG) == 0);      //Esperar RX
134     dado=UCB0RXBUF;
135     delay(50);
136     return dado;
137 }
138
139
140 // Escrever dado na porta
141 void PCF_write(char dado){
142     UCB0I2CSA = PCF_ADR;                    //Endereço Escravo
143     UCB0CTL1 |= UCTR          |    //Mestre TX
144     UCTXSTT;                //Gerar START
145     while ( (UCB0IFG & UCTXIFG) == 0)    ;        //Esperar TXIFG=1
146     UCB0TXBUF = dado;                    //Escrever dado

```

```
147     while ( (UCB0CTL1 & UCTXSTT) == UCTXSTT) ; //Esperar STT=0
148     if ( (UCB0IFG & UCNACKIFG) == UCNACKIFG) //NACK?
149         while(1); //Escravo gerou NACK
150     UCB0CTL1 |= UCTXSTP; //Gerar STOP
151     while ( (UCB0CTL1 & UCTXSTP) == UCTXSTP) ; //Esperar STOP
152     delay(50);
153 }
154
155
156
157 void delay(long limite){
158     volatile long cont=0;
159     while (cont++ < limite) ;
160 }
161
```