

```

1: /* EXEMPLOS DE APLICACAO EM LINGUAGEM C PARA MICROCONTROLADORES MICROCHIP
2: MCU: PIC16F877A
3: CRISTAL OSCILADOR: HS - 8,000 MHz
4: COMPILADOR: MIKROC PRO FOR PIC - VER:5.61
5: KIT DESENVOLVIMENTO: PICKIT PIC18F - MICROGENIOS
6: GRAVADOR USB: MICROICD ZIF - MICROGENIOS
7: PROGRAMADOR: PICKIT2 - VER: 2.61.00
8: =====
9: CHAVES DIP USADAS [ON] / [OFF]:
10: 1- GLCD / LCD = [LCD]      2- RX = [OFF]      3- TX = [OFF]      4- REL1 = [OFF]
11: 5- REL2 = [OFF]           6- SCK = [OFF]      7- SDA = [OFF]      8- RTC [OFF]
12: 9- LED1 = [OFF]           0- LED2 = [OFF]
13: 1- DIS1 = [OFF]           2- DIS2 = [OFF]      3- DIS3 = [OFF]      4- DIS4 = [OFF]
14: 5- INFR = [OFF]           6- RESIS = [OFF]      7- TEMP = [OFF]      8- VENT = [OFF]
15: 9- AN0 = [OFF]            0- AN1 = [OFF]      a- JUMPER BUZZER FECHADO
16: =====
17: DESCRICAO: AO PRESSIONAR-SE INT2 GRAVA UMA LETRA QUALQUER NA MEMORIA INTERNA
18: EEPROM, AO PRESSIONAR-SE INT2 [RB2] REALIZA-SE A LEITURA DO CONTEUDO GRAVADO.
19: ALEM DISSO APRESENTA MENSAGENS DE INSTRUCAO NA TELA DO LCD. APOS A GRAVACAO
20: [RB1] DESLIGAR O MICROCONTROLADOR DA ENERGIA ELETRICA, AGUARDAR PELO MENOS
21: 1 MINUTO, REENERGIZA-LO E VERIFICAR SE A INFORMACAO FOI RETIDA APENAS USANDO
22: A OPCAO DE LEITURA. MODIFICAR O CONTEUDO E REAPLICAR O PROCEDIMENTO.
23: ===== */
24: // CONEXOES PARA LCD PIC16F - MICROGENIOS
25: sbit LCD_RS at RE2_bit;      sbit LCD_RS_Direction at TRISE2_bit;
26: sbit LCD_EN at RE1_bit;      sbit LCD_EN_Direction at TRISE1_bit;
27: sbit LCD_D4 at RD4_bit;      sbit LCD_D4_Direction at TRISD4_bit;
28: sbit LCD_D5 at RD5_bit;      sbit LCD_D5_Direction at TRISD5_bit;
29: sbit LCD_D6 at RD6_bit;      sbit LCD_D6_Direction at TRISD6_bit;
30: sbit LCD_D7 at RD7_bit;      sbit LCD_D7_Direction at TRISD7_bit;
31: // DEFINICAO DAS VARIAVEIS
32: unsigned char INFO;          // INFORMACAO A SER GUARDADA NA MEMORIA
33: unsigned char INFO_s[6];     // INFORMACAO A SER GUARDADA NA MEMORIA 'STRING'
34: // VARIAVEIS DE MENSAGEM COM SEUS VALORES ATRIBUIDOS
35: unsigned char msg_1[] = "E E P R O M";
36: unsigned char msg_2[] = "TESTE D LE/GRAVA";
37: unsigned char msg_3[] = "G R A V A D A !";
38: unsigned char msg_4[] = "L E I T U R A : . .";
39: unsigned char msg_5[] = "PRESSIONAR: INT1";
40: unsigned char msg_6[] = "LE ou INT2.GRAVA";
41: // DIRETIVAS DO COMPILADOR
42: # define BUZZER PORTC.RC1      // BUZZER = SAIDA
43: # define INT2 PORTB.RB2        // INTERRUPTOR 2 / RB2 [SW1]
44: # define INT1 PORTB.RB1        // INTERRUPTOR 1 / RB1 [SW2]
45: void main() {                  // FUNCAO PRINCIPAL: MAIN
46: ADCON1 = 0x07;                // CONFIGURA ENTRADAS AN COMO E/S DIGITAIS
47: CMCON = 7;                    // DESLIGA COMPARADORES
48: // CONFIGURACOES E/S E ESTADOS INICIAIS [BINARIO]
49: TRISA = 0b00000000;          PORTA = 0b00000000;
50: TRISB = 0b00000110;          // PORTB = 0b00000000;
51: TRISC = 0b00000000;          PORTC = 0b00000000;
52: // TRISD = 0b00000000;        PORTD = 0b00000000;
53: // TRISE = 0b00000000;        PORTE = 0b00000000;
54: // ESTADOS INICIAIS DOS ACESSORIOS
55: BUZZER = 1;
56: // MENSAGEM INICIAL NO LCD
57: Lcd_Init();                    // INICIA O LCD
58: Lcd_Cmd(_LCD_CLEAR);          // LIMPA A TELA DO LCD
59: Lcd_Cmd(_LCD_CURSOR_OFF);    // DESLIGA O CURSOR
60: Lcd_Out(1, 1, msg_2);
61: Lcd_Out(2, 1, msg_1);
62: Delay_ms(1500);

```

```
63: Lcd_Cmd(_LCD_CLEAR); // LIMPA A TELA DO LCD
64: while (1) {
65: if (!INT2) {
66: BUZZER = 0; Delay_ms(15); BUZZER = 1; Delay_ms(50);
67: BUZZER = 0; Delay_ms(15); BUZZER = 1; // SIRENE SOA DUAS VEZES CURTAS
68: // EEPROM_Write(unsigned short address, unsigned short data);
69: EEPROM_Write(0x08, 'K'); //ESCREVE "K" NA POSICAO 0x08 DA EEPROM
70: Lcd_Cmd(_LCD_CLEAR); // LIMPA A TELA DO LCD
71: Lcd_Out(1, 1, msg_1);
72: Lcd_Out(2, 1, msg_3);
73: Delay_ms(1000);
74: Lcd_Cmd(_LCD_CLEAR); // LIMPA A TELA DO LCD
75: } else if (!INT1) {
76: BUZZER = 0; Delay_ms(15); BUZZER = 1; // SIRENE SOA UMA VEZ CURTA
77: // EEPROM_Read(unsigned short address);
78: INFO = EEPROM_Read(0x08); // LE A POSICAO 0x08 DA EEPROM
79: ShortToStr (INFO, INFO_s);
80: Lcd_Cmd(_LCD_CLEAR); // LIMPA A TELA DO LCD
81: Lcd_Out(1, 1, msg_1);
82: Lcd_Out(2, 1, msg_4);
83: Delay_ms(1000);
84: Lcd_Cmd(_LCD_CLEAR); // LIMPA A TELA DO LCD
85: Lcd_Out(1,1,INFO_s);
86: Lcd_Out_Cp(" = DECIMAL");
87: Lcd_Cmd(_LCD_SECOND_ROW);
88: Lcd_Chr_Cp(0+INFO);
89: Lcd_Out_Cp(" = CHARACTER");
90: Delay_ms(2500);
91: Lcd_Cmd(_LCD_CLEAR); // LIMPA A TELA DO LCD
92: else {
93: Lcd_Out(1, 1, msg_5);
94: Lcd_Out(2, 1, msg_6);
95: } // FIM ELSE
96: } // FIM WHILE
97: } // FIM MAIN
```