

```

1:  1: /*****
2:  2: /* Microcontroller: PIC16F877A @ 8 MHz + MICROGENIOS --> OK! */
3:  3: // E:\8.PICMICRO\CODEEX\0.MICROGENIOS\PIC16F877A\A.EEPROM_24C04
4:  4: /* PROGRAMA EXEMPLO: Escrita e Leitura da EEPROM externa 24C04. OBJETIVO:
5:  5: Aprender a utilizar a EEPROM externa 24C04. PIC16F877A
6:  6: MICROCONTROLADOR: PIC18F4520 --> ADAPTADO PARA PIC16F877A --> KIT PICgenios
7:  7: SOFTWARE: MikroC PRO PIC --> MICROGENIOS - CRISTAL: 8MHz. CHAVES DE FUNÇÃO:
8:  8: LCD ( 1) = ON / SCK ( 6) = ON / SDA ( 7) = ON.
9:  9: OBS: - HABILITAR BIBLIOTECA LCD. - HABILITAR BIBLIOTECA I2C. */
10: 10: // CONEXÕES PARA LCD PIC16F - MICROGENIOS
11: sbit LCD_RS at RE2_bit; sbit LCD_RS_Direction at TRISE2_bit;
12: sbit LCD_EN at RE1_bit; sbit LCD_EN_Direction at TRISE1_bit;
13: sbit LCD_D4 at RD4_bit; sbit LCD_D4_Direction at TRISD4_bit;
14: sbit LCD_D5 at RD5_bit; sbit LCD_D5_Direction at TRISD5_bit;
15: sbit LCD_D6 at RD6_bit; sbit LCD_D6_Direction at TRISD6_bit;
16: sbit LCD_D7 at RD7_bit; sbit LCD_D7_Direction at TRISD7_bit;
17: unsigned char ucContador; // AUXILIAR PARA CONTAGEM.
18: // PROTOTIPOS DE FUNÇÃO
19: void Escrita_24C04(unsigned char ucEndereco,unsigned char ucValor);
20: unsigned char Leitura_24C04(unsigned char ucEndereco);
21: void main(){
22:     TRISB = 0; // DEFINE PORTB COMO SAIDA.
23:     TRISD = 0; // DEFINE PORTD COMO SAIDA.
24:     TRISE = 0; // DEFINE PORTE COMO SAIDA.
25:     ADCON1 = 0x0F; // CONFIGURA TODOS CANAIS COMO DIGITAL.
26: // CONFIG. LCD NO MODO 4 BIT
27:     Lcd_Init(); Lcd_Cmd(_LCD_CLEAR); Lcd_Cmd(_LCD_CURSOR_OFF);
28: // ESCRITA NA EEPROM
29:     Lcd_Out(1,1,"Escr. na 24C04"); // MOSTRA O ESTADO NO LCD.
30:     for(ucContador = 0; ucContador < 16; ucContador++){
31: // Escrita_24C04(ucContador, 0x30 + ucContador);
32: // GRAVA NA EEPROM VALORES DE 0 A 10 EM ASCII.
33: // Escrita_24C04(ucContador, 0x41 + ucContador);
34: // GRAVA NA EEPROM VALORES DE 'A' ATÉ: 'P' MAIUSCULAS EM ASCII.
35:         Escrita_24C04(ucContador, 0x61 + ucContador);
36: // GRAVA NA EEPROM VALORES DE 'a' ATÉ: 'p' MINUSCULAS EM ASCII.
37:         Delay_ms(100); }
38: // LEITURA NA EEPROM
39:     Lcd_Out(1,1,"Lido da 24C04"); // MOSTRA O ESTADO NO LCD.
40:     for(ucContador = 0; ucContador < 16; ucContador++){
41:         Lcd_Chr(2, ucContador+1, Leitura_24C04(ucContador));
42: // LE NA EEPROM VALORES
43:         Delay_ms(100); } while(1){}; // END MAIN
44: // FUNÇÃO QUE ESCRIBE NA MEMÓRIA 24C04
45: void Escrita_24C04(unsigned char ucEndereco,unsigned char ucValor){
46:     I2C1_Init(100000); // FREQUENCIA DE CLOCK
47:     I2C1_Start(); // START NO BARRAMENTO I2C
48:     I2C1_Wr(0xA0); // ENDEREÇO DA MEMÓRIA NO HARDWARE DO KIT (W = 0)
49:     I2C1_Wr(ucEndereco); // ENDEREÇO DA EEPROM ONDE SERÁ SALVO O DADO
50:     I2C1_Wr(ucValor); // ESCRIBE VALOR DA VARIÁVEL 'UCVALOR' NA 24C04
51:     I2C1_Stop(); // FINALIZA A COMUNICAÇÃO I2C
52: // FUNÇÃO QUE FAZ A LEITURA DA MEMÓRIA 24C04
53: unsigned char Leitura_24C04(unsigned char ucEndereco){
54:     unsigned char ucLeitura; // TEMPORARIO PARA LEITURA DA MEMÓRIA 24C04.
55:     I2C1_Start(); // START NO BARRAMENTO I2C
56:     I2C1_Wr(0xA0); // ENDEREÇO DA MEMÓRIA NO HARDWARE DO KIT (W = 0)
57:     I2C1_Wr(ucEndereco); // ENDEREÇO DA EEPROM ONDE SERÁ LIDO O BYTE
58:     I2C1_Repeated_Start(); // RE-START NO BARRAMENTO
59:     I2C1_Wr(0xA1); // ENDEREÇO DA MEMÓRIA NO HARDWARE DO KIT (W = 1)
60:     ucLeitura = I2C1_Rd(0u); // Lê O BYTE
61:     I2C1_Stop(); // FINALIZA A COMUNICAÇÃO I2C
62:     return(ucLeitura); // RETORNA VALOR DA LEITURA.

```