

Rotina:memória_EEPROM_externa.c

Prof.: Alberto Willian Mascarenhas

```
#include <16F877A.h>
#device adc=8

#FUSES NOWDT           //No Watch Dog Timer
#FUSES HS              //High speed Osc (> 4mhz for PCM/PCH) (>10mhz for PCD)
#FUSES NOPUT          //No Power Up Timer
#FUSES NOPROTECT       //Code not protected from reading
#FUSES NODEBUG         //No Debug mode for ICD
#FUSES NOBROWNOUT     //No brownout reset
#FUSES NOLVP           //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
#FUSES NOCPD           //No EE protection
#FUSES NOWRT           //Program memory not write protected

#use delay(clock=20000000)

#ifndef lcd_enable
#define lcd_enable      pin_E1      // pino enable do LCD
#define lcd_rs          pin_E2      // pino rs do LCD
//#define lcd_rw        pin_e2      // pino rw do LCD
#define lcd_d4          pin_d4      // pino de dados d4 do LCD
#define lcd_d5          pin_d5      // pino de dados d5 do LCD
#define lcd_d6          pin_d6      // pino de dados d6 do LCD
#define lcd_d7          pin_d7      // pino de dados d7 do LCD
#endif
#include "C:\Alberto\IFMT 2023-II\Microcontroladores\Driver\mod_lcd.c"

#include "2404.C"

long int End_HL(byte H, byte L){
    return (H<<8)+ L;
}

char* IntToHex(int value) {
    int a;
    int b;
    char* buffer;
    a = value && 16;
    b = (value>>4)&16;
    buffer[0] = (a<10)?'0'+a:'A'-(a-10);
    buffer[1] = (b<10)?'0'+b:'A'-(b-10);
    return buffer;
}
```

```

WRITE_FLOAT_EXT_EEPROM(long int n, float data) {
    int i;

    for (i = 0; i < 4; i++)
        write_ext_eeprom(i + n, *((int8*)&data + i) );
}

float READ_FLOAT_EXT_EEPROM(long int n) {
    int i;
    float data;

    for (i = 0; i < 4; i++)
        *((int8*)&data + i) = read_ext_eeprom(i + n);

    return((float)data);
}

void eeprom_int16(unsigned int endereco, int16 valor_il6)
{
    write_ext_eeprom(endereco, make8(valor_il6,0));
    write_ext_eeprom(endereco+1, make8(valor_il6,1));
}

void main() {

    unsigned int8 a,b,c,d;
    unsigned int endereco;
    unsigned int16 Temperatura, tempNovo;
    float pressao = 42.37f;
    float result_f;

    setup_adc_ports(NO_ANALOGS);
    setup_adc(ADC_OFF);
    setup_psp(PSP_DISABLED);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);

    lcd_ini(); // Inicializa LCD
    delay_ms(10);
    init_ext_eeprom();
    delay_ms(10);
}

```

```

printf (lcd_escreve, "\f iniciando...");
delay_ms(1000);

while (true)
{
    // Escreve vários valores inteiros (um byte) a partir do endereço 01
    write_ext_eeprom(0x1, 6);
    write_ext_eeprom(0x2, 9);
    write_ext_eeprom(3, 12);
    write_ext_eeprom(4, 40);
    write_ext_eeprom(5, 50);
    write_ext_eeprom(6, 60);
    write_ext_eeprom(7, 70);

    printf(lcd_escreve, "\fValor: %d", read_ext_eeprom(0x2 )); // Lê o valor no endereço 02
    printf(lcd_escreve, "\r\nValor: %d", read_ext_eeprom(4)); // Lê o valor no endereço 04
    delay_ms(3000);

    // Escrita e leitura de números Longos
    Temperatura = 910;
    endereco = 8; // define o novo endereço -> 08
    eeprom_int16(endereco, Temperatura); // Escreve um valor de dois byte (longo) na memória externa

    a = read_ext_eeprom(8); // Lê um byte no endereço 08
    b = read_ext_eeprom(9); // Lê um byte no endereço 09
    tempNovo = make16(b,a); // Monta um longo (dois byte)
    printf(lcd_escreve, "\fTemp.: %lu", tempNovo);
    delay_ms(3000);

    // Escrita e leitura de float
    pressao = 42.37f;
    endereco = 10; // define o novo endereço
    WRITE_FLOAT_EXT_EEPROM(endereco, pressao); // Escreve um valor de quatro byte (float) na memória externa

    result_f = READ_FLOAT_EXT_EEPROM(10); // Lê um valor de quatro byte (float) na memória externa
    printf(lcd_escreve, "\fPressao: %4.1f", result_f);
    delay_ms(3000);
}
}

```