

```

/*
---- EDUQ - Cabeçalho para programa em linguagem C - Compilador CCS - Rev.2 ----
Microcontroladores PIC - PIC16F628A -- Interrupção Timer 0 -- Tratamento Manual
----- Prof. Valdir Dugo Zaragoza ----- 28/07/2015
*/
//
//----- Inicialização -----
//-----
#include <16F628A.h>

//Osc. cristal XT 4 Mhz, Watch Dog Timer Off, Power Up Timer On e Code Protect Off.

//#FUSES XT,NOWDT,PUT,NOPROTECT // Usar para PIC16F84A
#FUSES XT,NOWDT,MCLR,PUT,NOPROTECT,NOCPD,NOBROWNOUT,NOLVP // Usar para PIC16F628A

#use delay(clock=4M)

//----- Definições de SFR's -----
//-----
//Bytes:
//*
#byte TIMER_0 = 0x01
#byte STATUS = 0x03
#byte FSR = 0x04
#byte PORT_A = 0x05
#byte PORT_B = 0x06
#byte PCLATH = 0x0A
#byte INTCON = 0x0B

//Bit's:
#bit T0IF = INTCON.2
//*/
//----- Definições de Flag's -----
//-----
//Bytes:

//int flag_1=0;

//Bit's:
//#bit flag_250u = flag_1.0

//----- Declarações de entradas -----
//-----
#bit botao1 = PORT_A.0
//#bit botao2 = PORT_A.1
//#bit botao3 = PORT_A.2
//#bit botao4 = PORT_A.3
//#bit botao5 = PORT_A.4

//----- Declarações de saídas -----
//-----
#bit Led1 = PORT_B.0
//#bit Led2 = PORT_B.1
//#bit Led3 = PORT_B.2
//#bit Led4 = PORT_B.3
//#bit Led5 = PORT_B.4
//#bit Led6 = PORT_B.5
//#bit Led7 = PORT_B.6
//#bit Led8 = PORT_B.7

//----- Declarações de Variáveis -----
//-----
//utilizadas p/ salvar contexto:
//static int W_TEMP, STATUS_TEMP, FSR_TEMP, PCLATH_TEMP;

// Uso no programa:

//int x;

//----- Inicialização das Portas -----
//-----
#use fast_io(a)
#use fast_io(b)

```

```

//                                     Sub-rotinas
//-----
//                                     Interrupção do Timer_0 -----
//
// #int_timer0           // Usada no tratamento automático da interrupção por Timer 0
// #inline
void int_timer_0(){
T0IF = 0;
TIMER_0=TIMER_0+6;           //GERA OVERFLOW DO TIMER_0 A CADA 250uS

//                                     Adicionar o tratamento da int timer0 abaixo:
// flag_250u=1;

}
*/
//----- Salva Contexto -----
// void salva_contexto() {
// #asm
//     MOVWF W_TEMP
//     SWAPF STATUS,W
//     MOVWF STATUS_TEMP
//     MOVF FSR,W
//     MOVWF FSR_TEMP
//     MOVF PCLATH,W
//     MOVWF PCLATH_TEMP
//     CLRF PCLATH
//     CLRF STATUS
// #endasm
// }
//----- Restaura Contexto -----
// #asm
//     MOVF PCLATH_TEMP,W
//     MOVWF PCLATH
//     MOVF FSR_TEMP,W
//     MOVWF FSR
//     SWAPF STATUS_TEMP,W
//     MOVWF STATUS
//     SWAPF W_TEMP,F
//     SWAPF W_TEMP,W
// #endasm
// }
//----- Programa Principal -----

void main(){           // Programa principal

set_tris_a(0b11111111); // Definição das entradas e saídas (PORT_A = entrada e
PORT_B = saída)
set_tris_b(0b00000000);

//setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);//setup_wdt(WDT_18MS); // overflow 256us e WDT
não utilizado

PORT_A = 0;           // Inicialização das variáveis
PORT_B = 0;

//enable_interrupts(GLOBAL|INT_RTCC); // Habilita interrupções - Global (tratamento
manual) e Timer 0

//----- Laço de repetição -----

while(true){
}
}
//----- Tratamento Global das Interrupções -----
// #int_global // Interrupção Global (tratamento manual)
void trata_int(void){

    salva_contexto();
    if (T0IF) {
        int_timer_0();
    }
    restaura_contexto();
}
*/

```