

```

/*****
* File: timer0.c
* Author: Malakabot
* Description: Salta la interrupción del timer0 cada 500ms
* y parpadea el LED (PIN_C6)
* Hardware: PIC18F2550, Cristal 20Mhz, MKBot v.3
* Created: 21 agosto 2021
*****/

#include <xc.h>
#pragma config LVP = OFF
#pragma config FOSC = HSPLL_HS
#pragma config CPUDIV = OSC1_PLL2
#pragma config PLLDIV = 5
#define _XTAL_FREQ 4800000

__interrupt(high_priority) void ISR() // Subrutina de interrupción
{
    if (INTCONbits.TMR0IF==1) // Comprueba el flag del timer0
    {
        LATCbits.LATC6=!LATCbits.LATC6; // Invierte PIN_C6
        TMR0=18661; // Se actualiza para que no empiece contando desde CERO
    }
    INTCONbits.TMR0IF=0;
}

/*
tiempo = (cont_máx - cont_mín)*Prescaler*4*Tosc
tiempo = (65535+1 - 0)*128*4*(1/48MHz)=699ms
Si quisiéramos ajustar ese valor tendríamos que actualizar en ISR
los valores de TMR0L primero y después TMR0H (por lo del registro intermedio)
TMR0=18661, entonces...
tiempo= (65535+1 - 18661)*128*4*(1/48MHz)=500ms
*/
void conf_timer0()
{
    INTCONbits.GIE=1; // Habilita interrupciones
    INTCONbits.PEIE=1; // Habilita interrupciones periféricas
    INTCONbits.TMR0IE=1; // Habilita interrupción timer0
    INTCONbits.TMR0IF=0; // Flag de interrupción iniciado a 0

    T0CONbits.TMR0ON=1; // Habilita timer0
    T0CONbits.T08BIT=0; // Contador de 16 bits, desborda a 65535, interrupción salta al reiniciar
    T0CONbits.T0CS=0; // Fuente reloj interno. Cuenta los pulsos de 48MHz
    T0CONbits.T0SE=1; // Cuenta los flancos de bajada
    T0CONbits.PSA=0; // Se le asigna un valor al prescaler
    T0CONbits.T0PS=0b110; // Prescaler 1:128. Cuenta una unidad cada 128 flancos
}

void main()
{
    conf_timer0(); // Configura el timer0
    TRISCbits.TRISC6=0; // LED en PIN_C6
    LATCbits.LATC6=0; // LED empieza a 0

    while(1)
    {
    }
}

```