

```

#include "Timer1.h"

//parametros de configuracion del timer

/*****
Funciones globales pero privadas al archivo TIMER1.c
*****/
void    setOutPB1(){
    DDRB=(1<<PB1); //PB1=salida
}
void    setModeToggle(){
    TCCR1A=(1<<COM1A0); //COM1A=Toggle no funciona en modo PWM
}
void    setModeCTCnoClock(){
    TCCR1B=0; //Modo CTC no prescaler
}
uint8_t isRangoAdmitido(uint16_t frequency){
    return (F_min<=frequency)&&(frequency<=F_max);
}
uint8_t setTipeDelContador(uint16_t frecuenciaDeseada){
    if (isRangoAdmitido(frecuenciaDeseada)){
        resetCounterOn= (F_CPU/(2*1*frecuenciaDeseada))-1;
        return 1;
    }
    return 0;
}
void    setModeCTCnoPrescaler(){
    TCCR1B=(1<<WGM12)|(1<<CS10); //Modo CTC no prescaler
}

/*****
funciones de acceso publicas
*****/
void    TIMER1_set_module(){
    setTipeDelContador(6435);
    setOutPB1();
    setModeToggle();
}
uint8_t TIMER1_set_frequency(char frequencyStr[] ){
    return setTipeDelContador(atoi(frequencyStr));
}
void    TIMER1_set_on(){
    setModeCTCnoPrescaler();
}
void    TIMER1_set_off(){
    setModeCTCnoClock();
}
void    TIMER1_reset_module(){
    TIMER1_set_off();
    TIMER1_set_module();
}

```