TIMER VS DELAY

```
#pragma config FOSC = EXTRC      // Oscillator Selection bits (RC oscillator)
#pragma config WDTE = OFF        // Watchdog Timer Enable bit (WDT disabled)
#pragma config PWRTE = OFF       // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOREN = OFF       // Brown-out Reset Enable bit (BOR disabled)
#pragma config LVP = OFF         // Low-Voltage (Single-Supply) In-Circuit Serial Programming Enable bit (RB3 is d
#pragma config CPD = OFF         // Data EEPROM Memory Code Protection bit (Data EEPROM code protection off)
#pragma config WRT = OFF         // Flash Program Memory Write Enable bits (Write protection off; all program memo
#pragma config CP = OFF          // Flash Program Memory Code Protection bit (Code protection off)

#include <xc.h>
#define _XTAL_FREQ 6000000 // Define system clock frequency for delay functions
#define LED1 0x01
#define LED2 0x02
#define LED3 0x04

unsigned int a,b,c,count = 0;
void main(void) {
    TRISC = 0x00;
    PORTC = 0x00;


    T1CON = 0x01;

    INTCON |= 0xC0;// GIE AND PIE ENABLED
    PIE1 |= 0x01;
    PIR1 &= 0xFE;//
    while(1);
    return;
}

void __interrupt() ISR()
{
    if (PIR1 & 0x01)
    {
        a++;
        b++;
        c++;

        if(c == 15) //0.69
        if(c == 15) //0.69
        {
          PORTC^= (1 << 2); //0000 0000 ^ 0000 0100 = 0000 0100
            c=0;
        }
        if(a == 30) //1.308
        {
            PORTC^= (1 << 0); // 0000 0100 ^ 0000 0101 = 0000 0001
            a=0;
        }
         if(b == 45)//1.962
        {
             PORTC^= (1 << 1); // 0000 0001 ^ 0000 0011 = 0000 0010
            b=0;
        }
        PIR1 &= (~0x01);// flag reset
}
}

```