



**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI I INFORMATYKI**

**INSTYTUT AUTOMATYKI**

**KIERUNEK AUTOMATYKA I ROBOTYKA**

**STUDIA NIESTACJONARNE I STOPNIA**

**LABORATORIUM - GRUPA L1**

**TECHNIKA MIKROPROCESOROWA**

**ĆWICZENIE 3**

**Wykonali:**

95687 – Dawid Geschlecht

94911 – Łukasz Hanusiak

**Prowadzący:**

Mgr. Inż. Andreas Kowol

## 1. Opis programu

Program utworzono z dwóch timerów które aktywowane są za pomocą przycisku. Cykl pracy sygnalizują diody LED zielona oraz czerwona.

Cykl programu uruchamiany jest za pomocą przycisku microswitch. Przycisk ten aktywuje przerwanie, wewnątrz przerwania program aktywuje timer 0 który załącza diodę zieloną a po odliczeniu zadanego czasu wyłącza diodę zieloną i włącza diodę czerwoną. Po powtórzeniu cyklu dwa razy złączony zostaje przerwanie timera 1 i wykonanie programu w którym dioda czerwona zmienia swój stan.

## 2. Kod programu

```
#include "msp430g2553.h"
#define RED BIT6           //LED CZERWONY 1.6
#define GREEN BIT0         //LED ZIELONY 1.0
#define SW BIT3            //PRZYCISK 1.3
int POM=0;                //ZMIENNA POMOCNICZA
int POM2=0;               //ZMIENNA POMOCNICZA
void main(void){
    WDTCTL = WDTPW + WDTHOLD;           // Stop watchdog timer

    TA0CTL0 |= CCIE;                    //PRZERWANIE TIMER 0
    TA1CTL0 |= CCIE;                    //PRZERWANIE TIMER 1

    TA0CTL |= TASSEL_2 + TACLRL + ID_3; //LICZENIE W GÓRE
    TA1CTL |= TASSEL_2 + TACLRL + ID_3; //LICZENIE W GÓRE

    P1DIR |= RED;                       //LED CZERWONY
    P1OUT &= ~RED;
    P1DIR |= GREEN;                     //LED ZIELONY
    P1OUT &= ~GREEN;
    P1DIR &= ~SW;                       //PRZYCISK
    P1REN |= SW;
    P1OUT |= SW;

    TA0CCR0 = 37500;                    //USTAWIENIE CZASU TIMER 0
    TA1CCR0 = 12500;                    //USTAWIENIE CZASU TIMER 1

    P1IES &= ~SW;                       //USTAWIENIE PRZERWANIA NA PRZYCISK
    P1IE |= SW;

    __bis_SR_register(LPM0_bits + GIE); // aktywacja przerwania CPU
}

#pragma vector=PORT1_VECTOR
__interrupt void Port_1(void){          //PRZERWANIE PO PRZYCISNIĘCIU PRZYCISKU
    POM=0;
    TACTL |= MC_1;                      //URUCHOMIENIE TIMER 0
    P1IFG &= ~SW;                       //CZYSZCZENIE PRZERWANIA
}
```

```

/

#pragma vector = TIMER0_A0_VECTOR //PRZERWANIE TIMER 0
__interrupt void CCR0_ISR(void){

    POM2++;

    if(POM2==10)
    {
        switch (POM)
        {
            case 0:
            {
                P1OUT |= GREEN;
                break;
            }
            case 1:
            case 2:
            case 3:
            {
                P1OUT ^= GREEN;
                P1OUT ^= RED;
                break;
            }
            default:
            {
                POM=0;
                TA0CTL &= ~MC_3;
                P1OUT &= ~GREEN;
                P1OUT &= ~RED;
                POM2=0;
            }
        }

        POM++;
        POM2=0;
    }
}

#pragma vector = TIMER1_A0_VECTOR //PRZERWANIE TIMER 1
__interrupt void CCR1_ISR(void){
    POM2++;
    if (POM2==10)
    {
        for(POM; POM<=8; POM++)
        {

            P1OUT ^= RED;
            delay_cycles(200000);

        }

        TA1CTL &= ~MC_3;
    }
}
:}

```