



# Politechnika Opolska

## LABORATORIUM

### Teoria Mikroprocesorowa

KIERUNEK STUDIÓW:	AiR Ns		ROK STUDIÓW:	III
SEMESTR:	VI	ROK AKADEMICKI:	2019/2020	

### Temat ćwiczenia:

Program wykorzystujący programowalne diody LED oraz przycisk

### Projekt wykonali:

Nazwisko i imię:		Nazwisko i imię:	
1.	Szymon Słaboń	2.	Rychel Konrad
3.	Syguła Dariusz	4.	

Ocena:	Data:	Uwagi:

# Wstęp

Postawione przed nami zadanie polegało a napisaniu kodu do mikrokontrolera MSP-EXP430G2 firmy Texas Instruments. Płytką ta jest wyposażona w dwie programowalne diody LED i jeden programowalny przycisk. Warunkiem ćwiczenia było napisanie programu obsługującego w dowolny sposób te elementy.

## Opis programu

Nasz program jest dosyć prosty w działaniu. Po jego skompilowaniu i załadowaniu Obydwie diody pozostają wyłączone. Po pierwszym wciśnięciu włącza się czerwona dioda, która miga z określoną częstotliwością. Następne kliknięcie włącza również zieloną diodę i obie migają jednocześnie. Jeszcze jedno kliknięcie gasi czerwoną diodę tak, że miga już tylko zielona, a przy kolejnym wciśnięciu przycisku wyłącza się również zielona.

Działanie programu jest oparte na funkcji switch, w której każdy case to kolejne kliknięcia. Aby program wykonywał się cały czas switch został umieszczony w pętli while. Przed switchem umieszczona została instrukcja warunkowa if, która sprawdza czy przycisk został wciśnięty. Jeżeli tak (a uwzględniane są tylko wciśnięcia dłuższe niż czas opóźnienia) to zmienna flag zostaje zwiększona o 1. Właśnie ta zmienna flag będzie argumentem funkcji switch i będzie się on po kliknięciu wyłączającym obydwie diody zerować.

Program działa zgodnie z wytycznymi, jedynym problemem jest to, że program zignoruje wciśnięcie przycisku, jeżeli akurat będzie wykonywana pętla for w którymś z case'ów. Należy „trafić” w moment pomiędzy inkrementacjami zmiennej i.

### Skrypt programu:

```
#include <msp430.h>

#define SW BIT3 // Przycisk -> P1.3
#define RED_LED BIT0 // Czerwony LED-> P1.0
#define GREEN_LED BIT6 // Zielony LED -> P1.0

void main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Zatrzymywanie Watchdog timer'a

    P1DIR |= RED_LED+GREEN_LED; // Ustawianie diod na wyjściu
    P1DIR &= ~SW; // Ustawianie przycisku na wejściu
    P1REN |= SW; // Przyłączanie rezystora sterującego do przycisku
    P1OUT |= SW; // Włączanie przycisku
    P1OUT &= ~GREEN_LED; // Wyłączanie zielonej diody
    P1OUT &= ~RED_LED; // Wyłączanie czerwonej diody

    unsigned int flag = 0; // Definiowanie zmiennej bez znaku
    while(1)
    {
        if(!(P1IN & SW)) // Jeżeli przycisk jest wciśnięty
        {
            flag++; // Inkrementacja wartości zmiennej flag
            __delay_cycles(320000); // Opóźnienie
        }
        switch (flag)
        case 1:
        {
            {
```

```

P1OUT &= ~GREEN_LED; // Wyłączanie zielonej diody
P1OUT ^= RED_LED;    // Przełączanie czerwonej diody

volatile unsigned long i;
for(i = 0; i < 20000; i++); // Opóźnienie
break;

};
case 2:
{
    P1OUT ^= GREEN_LED; // Przełączanie zielonej diody
    P1OUT ^= RED_LED;   // Przełączanie czerwonej diody

    volatile unsigned long i;
    for(i = 0; i < 20000; i++); // Opóźnienie
    break;

};
case 3:
{
    P1OUT &= ~RED_LED; // Wyłączanie czerwonej diody
    P1OUT ^= GREEN_LED; // Przełączanie zielonej diody

    volatile unsigned long i;
    for(i = 0; i < 20000; i++); // Opóźnienie
    break;

};
case 4:
{
    P1OUT &=~RED_LED; // Wyłączanie czerwonej diody
    P1OUT &=~GREEN_LED; // Wyłączanie zielonej diody

    volatile unsigned long i;
    for(i = 0; i < 20000; i++); // Opóźnienie
    break;

};
default :
{
    flag=0;
    __delay_cycles(1000);
}
}
}
}

```