

PRZEDMIOT:

Technika Mikroprocesorowa

KIERUNEK STUDIÓW:

AUTOMATYKA I ROBOTYKA

ROK STUDIÓW:

3

SPECJALNOŚĆ:

-

SEMESTR:

6

ROK AKADEMICKI:

2019/2020Temat ćwiczenia:***Funkcje przerwania – program 2***Ćwiczenie wykonali:

<u>Nazwisko:</u>		<u>Imię:</u>		<u>Nazwisko:</u>		<u>Imię:</u>	
1.	Idzi	Dawid	2.	Pawlak	Kamil		

Uwagi:Data:Ocena za sprawozdanie:Zajęcia zdalne.

1. Cel ćwiczenia

Celem ćwiczenia było przy pomocy układu MSP430 G2 2553 oraz środowiska programistycznego CodeComposer użyć funkcji przerwań.

2. Zakres ćwiczenia

Do realizacji ćwiczenia wykorzystany fragmenty poprzedniego kodu. Założono, że w momencie naciśnięcia przycisku realizowane jest przerwanie. A to przerwanie to nic innego jak kod, który zapala poszczególne diody.

3. Kod programu

```
#include <msp430.h>

#define SW BIT3 // Switch -> P1.3
#define LED BIT0 // Red LED -> P1.0
#define LED2 BIT6 // green LED -> P1.0
int p,i;

void main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

    P1DIR |= LED2+ LED; // Set LED pin -> Output
    P1DIR &= ~SW; // Set SW pin -> Input
    P1REN |= SW; // Enable Resistor for SW pin
    P1OUT |= SW; // Select Pull Up for SW pin

    P1IES &= ~SW; // Select Interrupt on Rising Edge
    P1IE |= SW; // Enable Interrupt on SW pin

    __bis_SR_register(LPM4_bits + GIE); // Enter LPM4 and Enable CPU Interrupt
}
p=1;
#pragma vector=PORT1_VECTOR
__interrupt void Port_1(void)
{

    if(!(P1IN & SW)) // jeśli przycisk wcisniety
    {
        p=p+1; //inkrementuj zmienną p
        for(i = 0; i < 10000; i++);
    }

    switch(p) { //otwarcie petli switch
    case 1: // warunek 1
        P1OUT &=~BIT0+BIT6; //świeci tylko zielona dioda
        P1OUT |=BIT6;
        break;

    case 2:
        P1OUT &=~BIT6; //swieci sie tylko czerwona dioda
        P1OUT |=BIT0;
        break;

    case 3:
        P1OUT |=BIT0+BIT6; // świecą się obie diody
        break;
    }
```

```

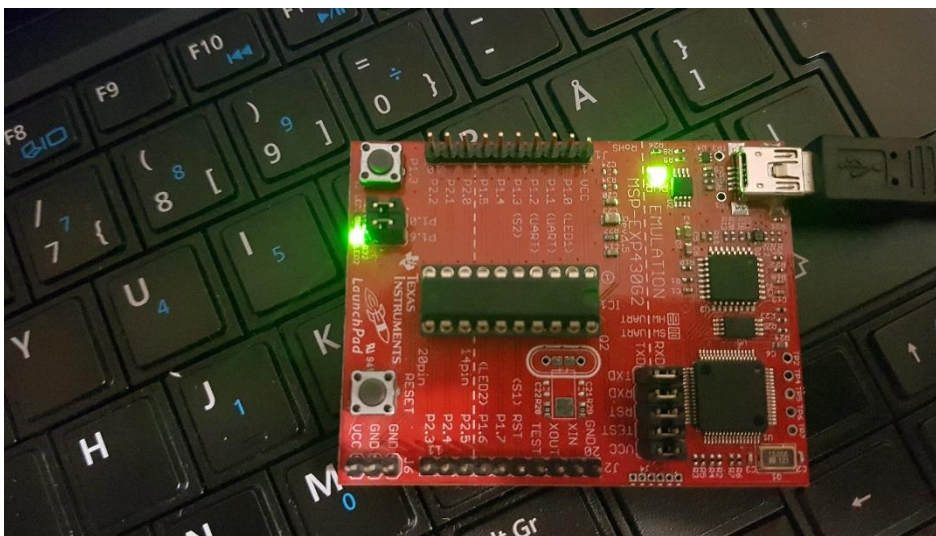
case 4:
    P1OUT &=~BIT0+BIT6; //nie świeci się żadna dioda
    break;

case 5: // zerowanie programu - swego rodzaju reset
    p=0;
    break;

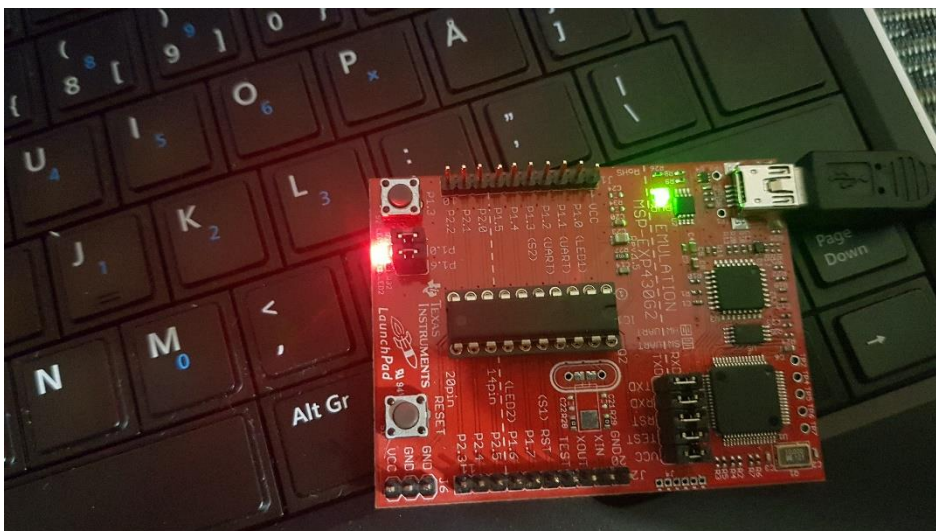
default:
    P1OUT &=~BIT0+BIT6; // opcja defaultowa
}
}

```

4. Działanie programu na obiekcie rzeczywistym



Rysunek 1- Instrukcja pierwsza – świeci tylko zielona dioda – jest to ustawienie startowe.



Rysunek 2 – instrukcja druga (naciśnięcie pierwszej) – świeci tylko czerwona dioda.



Rysunek 3 – instrukcja trzecia (naciśnięcie drugie) – świecą się obie diody (czerwona i zielona)



Rysunek 4 – Instrukcja czwarta (naciśnięcie trzecie) – wyłączenie obu diod.

Instrukcja piąta to kasowanie licznika naciśnięcie – rozpoczęcie cyklu od nowa.

Instrukcja **default** również wyłącza wszystkie diody. W momencie, kiedy nie świecą się diody, a przycisk nie reaguje powinniśmy zresetować program przyciskiem na układzie.

5. Wnioski

Funkcja przerwania została zrealizowana prawidłowo. Na pierwszy rzut oka można łatwo zauważyć, że funkcjonalność tego programu jest identyczna jak poprzedniego. Jednakże program wykorzystujący funkcje przerwania ma tą zaletę, że program wykona się niezależnie od tego co jest aktualnie wykonywane. Logicznie rzecz biorąc funkcja przerwania może być używana wtedy kiedy chcemy niezależnie od fragmentu programu, ani jego działania wymusić inną funkcję. Na przykładzie dwóch diod można to pokazać, również w ten sposób, że w głównym programie będzie się świecić czerwona dioda, a przyciskiem wywołamy przerwanie i zapalimy diodę zieloną.