

PRZEDMIOT:
Energoelektronika
KIERUNEK STUDIÓW:
AUTOMATYKA I ROBOTYKA
ROK STUDIÓW:
3
SPECJALNOŚĆ:
-
SEMESTR:
6
ROK AKADEMICKI:
2019/2020
Temat ćwiczenia:
Zapoznanie się z układem MSP430G2 – program 1.
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:
Termin zajęć:

Data:	07.03.2020	Dzień tygodnia:	SOBOTA	Godzina:	15:40
--------------	-------------------	------------------------	---------------	-----------------	--------------

1. Cel ćwiczenia

Celem ćwiczenia było zapoznanie się z układem MSP430 G2 2553 oraz środowiskiem programistycznym CodeComposer – służącym do łatwego kompilowania pisania, kompilowania oraz ładowania programów na ów układ.

2. Zakres ćwiczenia

W ćwiczeniu należało użyć podstaw języka C oraz instrukcji ze strony producenta, by wykonać prosty program uruchamiający, który będzie uruchamiać diody typu LED znajdujące się na płytce. Stworzono taki program. Zastosowano instrukcje **switch**, która pozwoliła na dodanie kilku wariantów świecenia diod. Kolejne instrukcje zmieniano wbudowanym w układ przyciskiem. A żeby to zrobić zastosowano prosty algorytm, który zlicza naciśnięcia przycisku. Oczywiście należy jawnie powiedzieć, że takie rozwiązanie ma wady. Bowiem przycisk zamontowany w układzie jest mało precyzyjny i zdarza mu się zinterpretować jedno naciśnięcie dwukrotnie (wynika to z długości trzymywania przycisku). Problem można zniwelować programowo (np. stosując wykrywanie przez ile czasu przycisk został wciśnięty). Prezentowany program nie posiada jeszcze takiej funkcjonalności.

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 |= LED+LED2; // 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
    p=1;

    while(1)
    {
        if(!(P1IN & SW))
        {
            p=p+1;0;
            for(i = 0; i < 10000; i++);
        }

        switch(p) {
        case 1:
            P1OUT &=~BIT0+BIT6;
            P1OUT |=BIT6;
            break;

        case 2:
            P1OUT &=~BIT6;
            P1OUT |=BIT0;
            break;

        case 3:
            P1OUT |=BIT0+BIT6;
            break;

        case 4:
```

```

P1OUT &=~BIT0+BIT6;

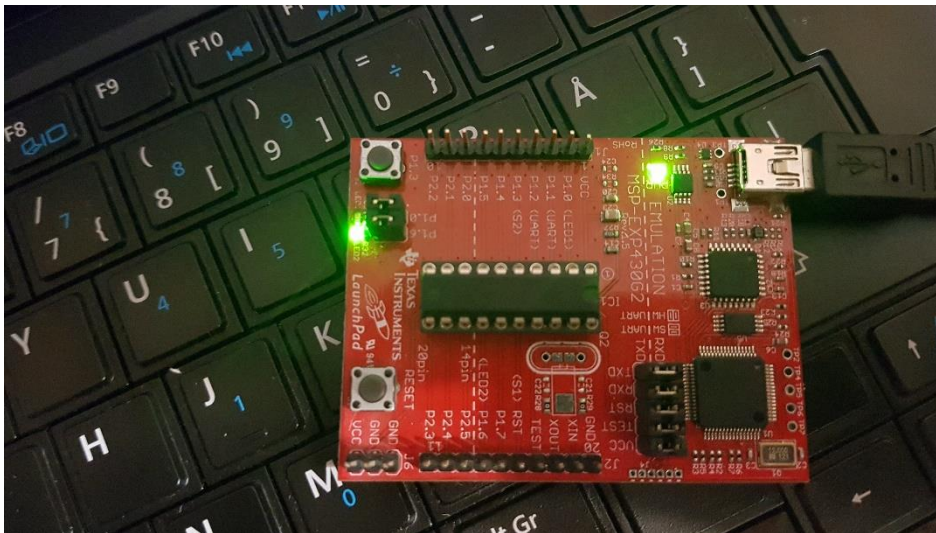
break;

case 5:
    p=0;
    break;

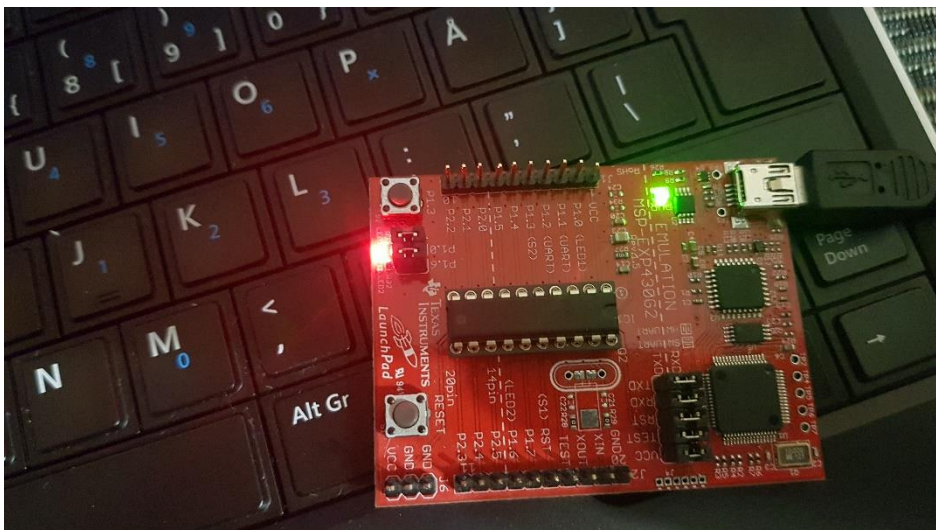
default:
    P1OUT &=~BIT0+BIT6;
    }
    }
}

```

4. Działanie programu na obiekcie rzeczywistym



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



Rysunek 2 – instrukcja druga (naciśnięcie pierwsze) – 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

Prezentowany program prezentuje możliwości układu, za pomocą diod LED i mnogości ich wariantów świecenia można pokazać, że pod kolejne przyciśnięcia przycisku można dołączyć dowolne instrukcje, które będą realizowane. Np. sterowanie wejściami i wyjściami. Układ MSP430 G2 2553 spełnił pokładane w nim nadzieję, a środowisko pozwoliło w przyjaznej formie napisać program i załadować do układu.