



POLITECHNIKA
OPOLSKA

WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI I INFORMATYKI

INSTYTUT AUTOMATYKI

Technika mikroprocesorowa

Projekt nr.2 Program z funkcją przerwania na układzie MSP430G2

Sprawozdanie wykonali i opracowali:

Kordian Kluzowski

Marcin Paś

Przemysław Jeziorny

kierunek: Automatyka i robotyka

studia niestacjonarne I stopnia

Opole 2020

1. Cel ćwiczenia

Celem ćwiczenia było napisanie lub zmodyfikowanie poprzedniego programu i użycie funkcji przerywania na układzie MSP430G2 w programie Code Composer.

2. Kod programu

```
1 #include <msp430.h>
2
3 #define SW      BIT3           // Zdefiniowanie przycisku
4 #define GREEN   BIT6           // Zdefiniowanie zielonej diody
5 #define RED     BIT0           // Zdefiniowanie czerwonej diody
6 int a;                       // Zdefiniowanie zmiennej a
7
8 void main(void) {
9     WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
10
11     P1DIR |= GREEN+RED;        // Wybór kierunku rejestru(wyjście)
12     P1DIR &= ~SW;              // Wybór kierunku rejestru(wyjście)
13
14     P1REN |= SW;               // Enable Resistor for SW pin
15     P1OUT |= SW;               // Stan wyjściowy przycisku - stan wysoki
16
17     P1IES &= ~SW;              // Wybór zbocza sygnału wyzwalającego przerwanie(zbocze opadające)
18     P1IE |= SW;                // Włączenie przerwania
19
20     __bis_SR_register(LPM4_bits + GIE); // Enter LPM4 and Enable CPU Interrupt
21 }
22
23 #pragma vector=PORT1_VECTOR
24 __interrupt void Port_1(void)
25 {
26     while(1)
27     {
28         if(!(P1IN & SW)) {
29             a=a+1;
30         }
31         if (a==1) {
32             P1OUT |= BIT0;
33             P1OUT ^= BIT0;
34             P1OUT |= BIT6;
35             __delay_cycles(1000000);
36         }
37         if (a==2) {
38             P1OUT |= BIT6;
39             P1OUT ^= BIT6;
40             P1OUT |= BIT0;
41             __delay_cycles(1000000);
42         }
43         if (a==3) {
44             P1OUT |= BIT0;
45             P1OUT |= BIT6;
46             __delay_cycles(1000000);
47         }
48         if (a==4) {
49             P1OUT &=~BIT0+BIT6;
50             __delay_cycles(1000000);
51         }
52         if(a==5) {
53             a=0;
54         }
55     }
56 }
57 P1IFG &= ~SW;                 // Flaga przerwania (1-'wejście'do procedury obsługi przerwania)
58 }
```

3. Opis działania programu i wnioski

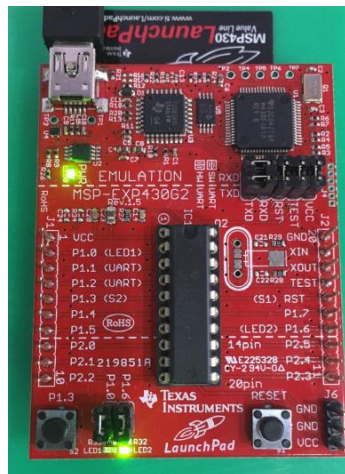
Program który napisaliśmy jest to modyfikacja naszego pierwszego programu.

Dodaliśmy tutaj funkcje przerwania która steruje diodami układu w następujący sposób:

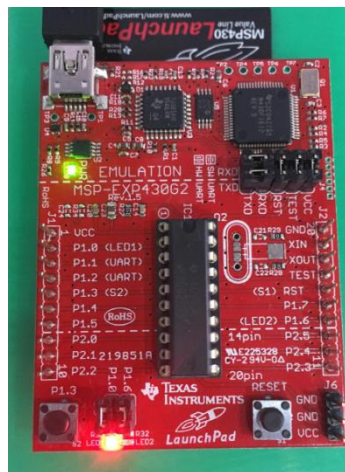
Po załączeniu układu diody są wyłączone i układ czeka na wymuszenie przyciskiem P1.3



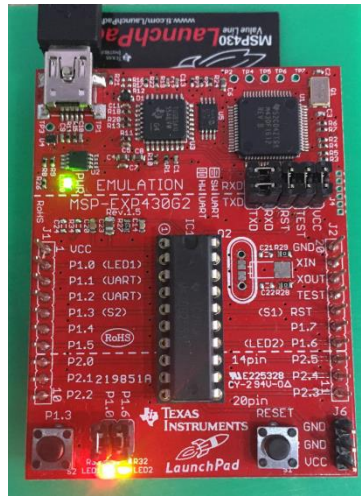
Po wciśnięciu przycisku na układzie zapala się tylko dioda zielona



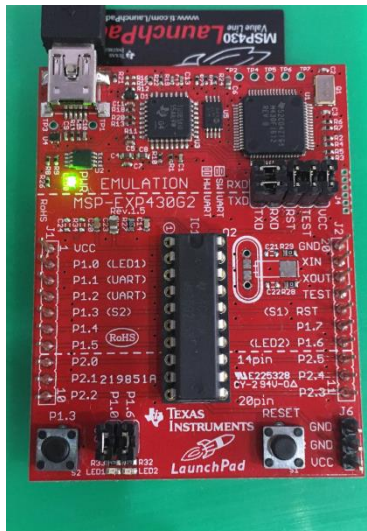
W kolejnym kroku zapali się tylko dioda czerwona. Aby przejść do kolejnego kroku przycisk musi być wciśnięty przez 1 sekundę



Następnym etapem będzie zapalenie się oby dwóch diod, również po wciśnięciu przycisku na 1 sekundę.



Zakończeniem programu będzie zgaszenie diod i zapętlenie.



Po kolejnym wciśnięciu przycisku na 1 sekundę powrócimy na początek programu i zapali się zielona dioda.

Główną instrukcją w tym programie jest IF, która pozwala na przełączanie się poszczególnych własności programu lecz tym razem wykorzystaliśmy ją w poleceniu przerwania.