



Politechnika Opolska

LABORATORIUM

Technika Mikroprocesorowa

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

Temat ćwiczenia:

Projekt wykonali:

<i>Nazwisko i imię:</i>		<i>Nazwisko i imię:</i>	
1.	Marco Tiszbierak	2.	Klaudiusz Tacica
3.	Marek Szczekała	4.	

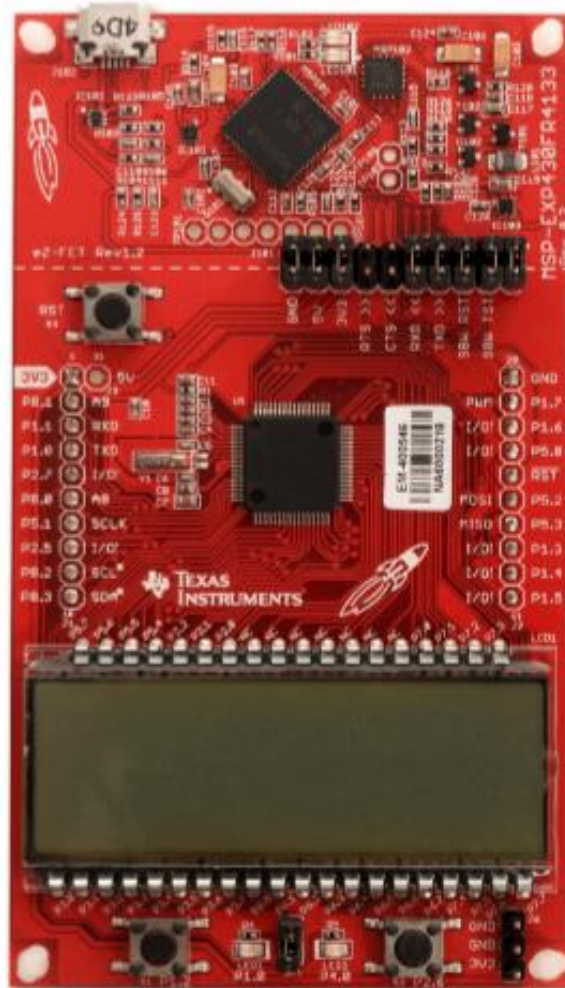
<i>Ocena:</i>	<i>Data:</i>	<i>Uwagi:</i>

1. Cel ćwiczenia:

Głównym celem ćwiczenia było napisanie programu w języku C według ustalonych przez prowadzącego Laboratorium wytycznych, wgranie oraz wytestowanie poprawności działania programu. Program zostało napisany oraz wgrany poprzez aplikację Code Composer Studio.

2. Opis urządzenia wykorzystanego w zadaniu:

Zestaw deweloperski MSP-EXP430FR4133 LaunchPad™ to łatwy w użyciu moduł (EVM) dla mikrokontrolera MSP430FR4133 (patrz rysunek 1). Zawiera wszystko, co jest potrzebne do rozpoczęcia prac nad platformą mikrokontrolera MSP430™ FRAM opartą na pamięci RAM (ULU), w tym emulację na pokładzie do programowania, debugowania i pomiarów energii. Płytkę zawiera wbudowane przyciski i diody LED do szybkiej integracji prostego interfejsu użytkownika i wyświetlacza ciekłokrystalicznego (LCD).



Rys.1 Płytkę firmy Texas Instruments MSP-EXP430FR4133 na której zostało wykonane zadanie.

3. Założenia Programu:

Napisany program miał na celu wykonywać proste zadanie, mianowicie po naciśnięciu przycisku miał wyzwalać lampkę LED o określonym kolorze. W naszym wypadku jako pierwsza po wgraniu programu oraz wciśnięciu przycisku włączała się LED czerwona, która miała migać z określoną częstotliwością do momentu ponownego wciśnięcia przycisku, który tym razem miał za zadanie wyłączyć lampkę czerwona po czymysterować działanie lampki LED drugiej zielonej, która również miała pulsować z określoną swoją częstotliwością do momentu ponownego wciśnięcia przycisku, który znów wyłączał LED zieloną oraz ponownie włączał czerwoną i tak w kółko.

4. Program

```
1 |
2 | #include <msp430.h>
3 |
4 | #define SW      BIT3           // Przycisk -> P1.3
5 | #define RED     BIT0           // LED czerwona -> P1.0
6 | #define GREEN   BIT6           // LED zielona -> P1.6
7 |
8 | void main(void) {
9 |     WDTCTL = WDTPW | WDTHOLD; // Sprawdzanie na błędy w kodzie
10 |
11 |     P1DIR |= RED+GREEN;         // Ustalenie pinów LED -> Wyjście
12 |     P1DIR &= ~SW;               // Ustawianie przycisku SW -> wejście
13 |     P1REN |= SW;                // Włącz rezystor dla pinu SW
14 |     P1OUT |= SW;                // Select Pull Up for SW pin
15 |
16 |     volatile unsigned int flag = 0;
17 |     while(1)
18 |     {
19 |         if(!(P1IN & SW))         // Gdy przycisk jest wciśnięty
20 |         {
21 |             _delay_cycles(20000); // Czekaj 20ms - czekamy 20ms na ustabilizowanie się przycisku
22 |             if(!(P1IN & SW))      // Sprawdzanie czy jest nadal przycisk wciśnięty
23 |             {                     // Ignoruj gdy sygnał wciśnięcia przycisku krótszy niż 20ms
24 |                 while(!(P1IN & SW)); // Poczekaj na puszczenie przycisku
25 |                 flag = !flag;       // Zmiana wartości flagi
26 |             }
27 |         }
28 |         if(flag)                 // Sprawdź wartość flagi
29 |         {
30 |             P1OUT &= ~GREEN;      // Zielona LED -> Wyłącz
31 |
32 |             volatile unsigned int i; // Zadeklarowanie zmiennej i
33 |
34 |             P1OUT ^= 0x01;         // Przełączenie czerwonej LED P1.0
35 |
36 |             i = 20000;              // Określenie czasu częstotliwości mrugania
37 |             do i--;
38 |             while(i != 0);
39 |
40 |
41 |         }
42 |         else
43 |         {
44 |             P1OUT &= ~RED;         // Czerwona LED -> Wyłącz
45 |
46 |             volatile unsigned int i; // Zadeklarowanie zmiennej i
47 |
48 |             P1OUT ^= BIT6;         // Czerwona LED -> Wyłącz
49 |
50 |             i = 10000;              // Określenie czasu częstotliwości mrugania
51 |             do i--;
52 |             while(i != 0);
53 |
54 |         }
55 |     }
```

