Politechnika Opolska

LABORATORIUM

Technika Mikroprocesorowa

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

Temat ćwiczenia:
Program wykorzystujący przerwania

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. Ćwiczenie polegało na stworzeniu programu operującego na tzw. instrukcjach przerwania.

Opis programu

Po uruchomieniu programu zaświeca się dioda, która miga z określoną częstotliwością. Po wciśnięciu przycisku miga ona dalej ale zaświeca się również druga, która gaśnie po puszczeniu przycisku.

Na początku programu widzimy definiowanie diod i przycisku, później przypisywanie ich do wejść i wyjść. Po tym następuje przypisanie przycisku jako elementu wykonawczego przerwań. Przerwania nie są domyślnie obsługiwane w języku C, najpierw więc trzeba uruchomić ich wykonywanie. Do tego służ funkcja __bis_SR_register(GIE). Następująca po niej pętla odpowiada za miganie diody na początku programu z określoną częstotliwością. Po wciśnięciu przycisku zostaje wykonana funkcja przerwania i zaświeca się druga dioda. Po puszczeniu przycisku następuje zresetowanie flagi przerwania i przerwanie tej funkcji.

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 |= GREEN_LED + RED_LED;          // Klasyczne definiowanie

   P1OUT |= RED_LED;
   P1DIR &= ~SW;
   P1REN |= SW;
   P1OUT |= SW;

   P1IES &= ~SW;                                  // Zerowanie flagi przerwań
   P1IE |= SW;                                    // Włączanie obsługi przerwań

   __bis_SR_register( GIE);                       // Odblokowanie obsługi przerwań

   while(1)
    {
        P1OUT ^= RED_LED;                         // Przełączanie czerwonej diody
     __delay_cycles(32000);                       // Opóźnienie przełączania
    };

}

// Funkcja przerwania
#pragma vector=PORT1_VECTOR
__interrupt void Port_1(void)
{

   P1OUT ^= GREEN_LED;                            // Przełączanie zielonej diody
   P1IFG &= ~SW;                                  // Zerowanie flagi przerwania
   __bic_SR_register_on_exit( LPM4_bits );                  // Zakończenie funkcji przerwania

}
```