



POLITECHNIKA
OPOLSKA

Wydział Elektroniki, Automatyki i Informatyki
Instytut Automatyki

Technika mikroprocesorowa

Ćwiczenie 2

opracował:
Sebastian Juraszek

Opole, 2020

1. Wstęp

Głównym celem programu jest realizacja funkcji przerwań. Port P1 i P2 posiadają taką możliwość. Każdy bit PxIE odblokowuje możliwość modyfikacji powiązanego bitu z rejestru flag PxIFG a więc załącza korespondujące przerwanie zewnętrzne:

- Bit = 0: Przerwanie wyłączone,
- Bit = 1: Przerwanie włączone.

Główny włącznik przerwań maskowanych do których zaliczają się przerwania zewnętrzne z portów P1 i P2 to bit GIE. Załączenie bitu GIE w języku C:

```
_BIS_SR(GIE);
```

Każdy bit rejestrów PxIES ustawia zbocze, które powoduje wygenerowanie flagi przerwania, korespondujący z pozycją w rejestrze PxIFG oraz odpowiednią linią portu wejścia wyjścia P1 lub P2:

- Bit = 0: Flaga PxIFG ustawiana z pojawieniem się zbocza Lo-Hi ,
- Bit = 1: : Flaga PxIFG ustawiana z pojawieniem się zbocza Hi-Lo.

2. Opis działania układu

Po uruchomieniu programu powinna zaświecić się dioda czerwona, następnie naciśnięcie przycisku S1 zaświeca diodę zieloną a czerwoną wyłącza. Kolejne kliknięcie S1 wywołuje reakcję wyłączenia diody zielonej.

3. Kod programu

```
#include
<msp430.h>

#define SW    BIT3           // przycisk S1 -> P1.3
#define GREEN BIT6           // zielona LED -> P1.6
#define RED   BIT0           // Czerwona LED -> P1.0

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

    P1DIR |= GREEN;                      // Set LED pin -> Output
    P1DIR |= RED;                        // Set LED pin -> Output
    P1OUT ^= RED;                        // Zaszwiecenie czerwonej diody
    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
```

```

}

#pragma vector=PORT1_VECTOR
__interrupt void Port_1(void)
{
    P1OUT ^= GREEN;           // Toggle Green LED
    P1OUT ^= ~RED;            // Toggle Green LED
    P1IFG &= ~SW;             // Clear SW interrupt flag
}

```

4. Wnioski.

Ćwiczenie zostało wykonane bez płytki MSP430G22553. W teorii kod powinien spełniać założoną funkcję realizacji przerwań. W programie zostały użyte diody jak i przyciski znajdujące się na mikrokontrolerze. Mimo braku urządzenia myślę, że program będzie działał poprawnie.