



Politechnika Opolska

LABORATORIUM

Teoria Mikroprocesorowa

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

Temat ćwiczenia:

Program wykorzystujący programowalne diody LED oraz przycisk

Projekt wykonali:

Nazwisko i imię:		Nazwisko i imię:	
1.	Ryszard Hałapacz	2.	
3.	Leszek Cieśla	4.	

Ocena:	Data:	Uwagi:

Wstęp

Celem ćwiczenia było wykorzystanie mikrokontrolera MSP-EXP430G2 producenta Texas Instruments i zaprogramowanie dwóch programowalnych diod za pomocą programowalnego przycisku. Użyto środowiska (IDE) Code Composer Studio (środowisko dostarczone przez producenta) w systemie operacyjnym Windows. Program został napisany w języku C, do którego załączono bibliotekę obsługującą mikrokontroler.

Opis programu

Zasada działania Programu:

Mikrokontroler, reaguje na przycisk. Reakcja jego polega na odpowiednim włączaniu i wyłączaniu zasilania diod – czerwonej i zielonej. W pierwszym etapie następuje konfiguracja wejść i wyjść, oraz definiowanie zmiennych globalnych oraz deklaracja funkcji użytych w programie. Stan początkowy, to stan w którym obydwie diody zostały wyłączone. Następnie przechodzimy do pętli while - to w niej wykonuje się zasadnicza część programu. Warunek w pętli zawsze zwraca true, efektem czego jest brak możliwości wyłączenia mikrokontrolera (poza odcięciem od źródła zasilania). Każdorazowe wykonanie pętli wywołuje funkcję `case_flag` która zwraca zależną od warunku zmodyfikowaną zmienną globalną `flag`. Nim jednak przejdziemy do opisu, warto skoncentrować się argumentach tej funkcji. Funkcja za argument przyjmuje drugą użytą w programie funkcję. Zadaniem tej drugiej jest sprawdzenie czy przycisk jest wciśnięty i ewentualna późniejsza inkrementacja zmiennej globalnej `flag`. Wartość jest zwracana jako argument do funkcji `case_flag`, przechodzimy wówczas do ciała funkcji. Znajduję się w niej warunek wielokrotnego wyboru `switch` i on determinuje zasilanie diod. Warto przyjąć się opcji `default`, w nie następuje zerowanie zmiennej globalnej.

Skrypt programu:

```
#include <msp430.h>

#define SW BIT3
#define RED BIT0
#define GREEN BIT6
#define Time 1000
unsigned int flag = 0;

int case_flag();
int check();

void main(void)
{
    WDTCTL = WDTPW | WDTHOLD;

    P1DIR |= RED+GREEN;
    P1DIR &= ~SW;
    P1REN |= SW;
    P1OUT |= SW;
    P1OUT &= ~GREEN;
    P1OUT &= ~RED;
```

```

while(1)
{
    case_flag(check());
}

int check ()
{
    if(!(P1IN & SW))
    {
        flag ++;
        delay_cycles(320000);
    }
    return flag;
}

int case_flag()
{
    switch (flag)
    {
        case 1:
        {
            P1OUT |=RED;
            P1OUT &=~GREEN;
            delay_cycles(Time);
            break;
        };
        case 2:
        {
            P1OUT &=~RED;
            P1OUT |=GREEN;
            delay_cycles(Time);
            break;
        };
        case 3:
        {
            P1OUT |=RED;
            P1OUT |=GREEN;
            delay_cycles(Time);
            break;
        };
        case 4:
        {
            P1OUT &=~RED;
            P1OUT &=~GREEN;
            delay_cycles(Time);
            break;
        };
        default :
        {
            flag=0;
            delay_cycles(Time);
        }
    }
}

```