



WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI I INFORMATYKI
INSTYTUT AUTOMATYKI
KIERUNEK AUTOMATYKA I ROBOTYKA
STUDIA NIESTACJONARNE I STOPNIA

LABORATORIUM - GRUPA L1

TECHNIKA MIKROPROCESOROWA

ĆWICZENIE 1.

Mruganie lampek sygnalizacyjnych mikrokontrolera MSP430

Wykonali:

95687 – Dawid Geschlecht
94911 – Łukasz Hanusiak

Prowadzący:

Mgr. Inż. Andreas Kowol

1. CEL I ZAKRES ĆWICZENIA

Celem ćwiczenia było napisanie kodu, który miał sterować mruganiem diod kontrolnych.

2. WSTĘP TEORETYCZNY

Jednym z języków którymi można programować mikroprocesory jest język C. Klasyfikuje się go jako język wysokiego poziomu do programowania systemów operacyjnych i innych zadań niskiego poziomu. Jego zaletą jest szybkość wczytywania się kodu i realizacji zadania. Kolejnym plusem jest stosunkowo małowymagającym językiem jeżeli chodzi o pamięć.

- Zapis skrótowy

```
int a = 1;
a += 5;    /* to samo, co a = a + 5;    */
a /= a + 2; /* to samo, co a = a / (a + 2); */
a %= 2;    /* to samo, co a = a % 2;    */
```

W języku C stosuje się zapis skrótowy, który początkującym może niewiele mówić, lecz gdy się do niego przyzwyczaimy staje się ten zapis bardzo sympatyczny.

- Operatory bitowe

- negacja bitowa (NOT) (" ~ "),
- koniunkcja bitowa (AND) (" & "),
- alternatywa bitowa (OR) (" | "),
- alternatywa rozłączna (XOR) (" ^ ").

"~" a	"&" a b	" " a b	"^" a b
0 1	0 0 0	0 0 0	0 0 0
1 0	1 1 1	1 1 1	0 1 1
	0 0 1	1 0 1	1 0 1
	0 1 0	1 1 0	1 1 0

- Operacje na 4 bitach

a		0101	=	5
b		0011	=	3
-----+				
~a		1010	=	10
~b		1100	=	12
a & b		0001	=	1
a b		0111	=	7
a ^ b		0110	=	6

3. Kod programu

```
1 #include <msp430.h>
2
3 #define SW BIT3 // Switch -> P1.3
4 #define RED BIT0 // Red LED -> P1.0
5 #define GREEN BIT6 // Red LED -> P1.0
6 #define Time 1000
7 void main(void) {
8     WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
9
10    P1DIR |= RED+GREEN; // Set LED pin -> Output
11    P1DIR &= ~SW; // Set SW pin -> Input
12    P1REN |= SW; // Enable Resistor for SW pin
13    P1OUT |= SW;
14    P1OUT &= ~GREEN;
15    P1OUT &= ~RED; // Select Pull Up for SW pin
16    unsigned int flag = 0;
17    while(1)
18    {
19        if(!(P1IN & SW)) // If SW is Pressed
20        {
21
22            flag ++;
23            __delay_cycles(320000);
24        }
25        switch (flag)
26        case 1:
27        {
28            {
29                P1OUT |= RED;
30                P1OUT &= ~GREEN;
31                __delay_cycles(Time);
32                break;
33            };
34            case 2:
35            {
36                P1OUT &= ~RED;
37                P1OUT |= GREEN;
38                __delay_cycles(Time);
39                break;
40            };
41            case 3:
42            {
43                P1OUT |= RED;
44                P1OUT |= GREEN;
45                __delay_cycles(Time);
46                break;
47            };
48            case 4:
49            {
50                P1OUT &= ~RED;
51                P1OUT &= ~GREEN;
52                __delay_cycles(Time);
53                break;
54            };
55            default :
56            {
57                flag=0;
58                __delay_cycles(Time);
59            }
60        }
61    }
62 }
```

4. Wnioski i spostrzeżenia

Z powodu zaistniałej sytuacji, kod powstał przy współpracy z grupą, ponieważ nie posiadamy mikrokontrolera MSP430 w domu. Kod działa tak, że po pierwszym przyciśnięciu przycisku zapali się pierwsza dioda (zielona), następnie zgaśnie zielona i zapali się sama czerwona. Kolejne czyli trzecie przyciśnięcie zapali wszystkie diody naraz (zielona, czerwona). Czwarte przyciśnięcie zgasi obie diody.