

# Techniki mikroprocesorowe, laboratorium 3

## Autorzy: Konrad Sobolowski, Bartłomiej Boczek

**Treść zadania:** Korzystając z modułu mikrokontrolera MSP430 oraz innych modułów systemu SML-3 należy zaprojektować i zrealizować licznik 8-bitowy – ładowanie, zliczanie w górę – kod NKB

### Podłączone moduły

Do mikrokontrolera MSP430 zostały podłączone następujące moduły:

- Port 1 => Moduł z pokrętłami hex
- Port 2 => Moduł z przyciskami monostabilnymi
- Port 3 => Moduł z dwusegmentowym wyświetlaczem liczb hex

### Wydruk kodu programu

```
1 ;-----
2 ; MSP430 Assembler Code Template for use with TI Code Composer Studio
3 ;
4 ;
5 ;-----
6 .cdecls C,LIST,"msp430.h" ; Include device header file
7
8 ;-----
9 .def RESET ; Export program entry-point to
10 ; make it known to linker.
11 ;-----
12 .text ; Assemble into program memory.
13 .retain ; Override ELF conditional linking
14 ; and retain current section.
15 .retainrefs ; And retain any sections that have
16 ; references to current section.
17
18 ;-----
19 RESET mov.w #_STACK_END,SP ; Initialize stackpointer
20 StopWDT mov.w #WDTPW|WDTHOLD,&WDTCTL ; Stop watchdog timer
21
22
23 ;-----
24 ; Main loop here
25 ;-----
26
27 INITIALIZATION:
28 wyświetlaczem BIS.B #0FFh, &P3DIR ;okreslamy kierunek przepływu => port P3 jest podłączony do modułu z
29 CLR.B &P3OUT ;wyzערuj wyjścia
30
31 MOV.B #003h, &P2IE ;włączanie przerwań dla pinów 1 i 2 należących do portu 2
32 MOV.B #003h, &P2IES ;flaga przerwania (xIFG) ustawiana zboczem opadającym
33 BIS.B #000h, &P2DIR ;port P2 ustawiony jako wejściowy => post podłączony do modułu z
34 przyciskami BIS.B #000h, &P1DIR ;pirt P1 ustawiony jako wejściowy => post podłączony do modułu z
35 pokrętłami hex
36
37 chcieli wyświetlić MOV.B &P1IN, R9 ;wczytuje wektor z hexa, rejestr R9 przechowuje liczbę którą będziemy
38 MOV.B R9, &P3OUT ;inicjalizacja wyświetlacza
39
40 MAIN:
41 ładowania MOV.B #000H, R5 ;inicjalizacja flagi mówiącej czy przerwanie nastąpiło poprzez przycisk
42 MOV.B R9, &P3OUT ;wyświetlenie zawartości rejestru R9 na wyświetlaczu
43 MOV.B #003h, &P2IE ;włączanie przerwań dla pinów 1 i 2 należących do portu 2
44 BIS.W #GIE+CPUOFF+OSCOFF+SCG1+SCG0,SR ;wprowadzenie procesora w tryb energooszczędny => LPM4
45 NOP
46
47 MOV.W #00D00h, R15 ;inicjalizacja licznika potrzebnego do niwelacji drgań styków
48
```

```

49      styków      MOV.W    #00800h, R14      ;inicjalizacja drugiego z liczników potrzebnych do niwelacji drgań
50
51      BIT.B      #001h,R5      ;sprawdzenie czy przewanie było spowodowane przyciskiem ładowania
52      JZ MAIN      ;jeżeli tak, to nie musimy unikać drgań styków
53
54  PREVENT_VIBRATIONS:
55      DEC.W      R15      ;zdekrementuj zawartość R15 => licznika
56      JZ MAIN      ;jeżeli R15 == 0 skocz do MAIN
57      BIT.B      #001h, &P2IN      ;sprawdź czy wciśnięty jest przycisk numer 1
58      JNZ PREVENT_VIBRATIONS      ;i jeżeli jest != 0 to skocz do obsługi dragania styków
59      DEC.W      R14      ;zdekrementuj zawartość w rejestrze R14
60      JNZ PREVENT_VIBRATIONS
61      ;;;;;;;;;;;;;;
62  INCREMENT:
63      BIC.W      #GIE, SR      ;nie przyjmuj przerwań
64      BIT.B      #001h, R6      ;czy oba klawisze są wciśnięte na raz
65      MOV.B      #000h, R6      ;wyzeruj flagę
66      JNZ MAIN      ;jeśli 2 guziki są wciśnięte to nie inkrementujemy
67      INC.B      R9      ;inkrementuj R9
68      MOV.B      R9, &P3OUT      ;wyświetl aktualną liczbę (rejestr R9) na wyświetlacz
69      JMP MAIN
70
71
72  INTERRUPT:
73      BIT.B      #002h, &P2IN      ;czy drugi przycik (ładowania) jest wciśnięty
74      JZ INT_LOAD      ;jeżeli tak, to rozpocznik operację ładowania
75      CLR.B      &P2IFG
76      BIC.W      #CPUOFF, 0(SP)      ;jeżeli nie, to powrót z przerwania po śladzie na stosie
77      RETI
78
79  INT_LOAD:
80      MOV.B      #000h, R6      ;wyzeruj R6
81      BIT.B      #001h, &P2IN      ;czy pierwszy przycisk jest wciśnięty
82      JNZ INT_END      ;jeżeli nie to skacz do INT_END
83      MOV.B      #001h, R6      ;jeżeli tak to ustaw flagę że dwa wciśnięte
84
85  INT_END:
86      MOV.B      &P1IN, R9      ;wczytaj wartość z pokręteł do rejestru R9
87      MOV.B      R9, &P3OUT      ;a następnie wyświetl tą wartość na wyświetlacz
88      MOV.B      #001h, R5      ;ustaw flagę mówiącą, że zostało wykonane ładowanie
89      CLR.B      &P2IFG
90      BIC.W      #CPUOFF+GIE, 0(SP);powrót z przerwania po śladzie na stosie
91      RETI
92
93
94
95
96  ;-----
97  ; Stack Pointer definition
98  ;-----
99      .global __STACK_END
100     .sect .stack
101
102  ;-----
103  ; Interrupt Vectors
104  ;-----
105     .sect ".reset"      ; MSP430 RESET Vector
106     .short RESET
107     .sect ".int01"
108     .short INTERRUPT

```

## Podsumowanie

W powyższym projekcie użyty został system oszczędzania energii przez mikrokontroler MSP430. Kontroler po wstępnej inicjalizacji, wyłączeniu watchdoga i ustawieniu odpowiednich rejestrów, przechodzi w stan oszczędzania energii z włączonymi przerwami. Port 2 ustawiony jest jako wejściowy z dwoma pierwszymi pinami ustawionymi jako zgłaszającymi przerwania. Do owych dwóch pinów podpięte są dwa przyciski monostabilne ( zerowy to dekrementacja, pierwszy odpowiedzialny jest za resetowanie )

Przy naciśnięciu jednego z przycisków, a co za tym idzie zgłoszeniu przerwania, procesor wychodzi ze stanu uśpienia i realizowana jest odpowiednia procedura obsługi przerwań INTERRUPT odpowiadającemu przerwaniom zgłaszanym przez port 2 wedle specyfikacji urządzenia. Jeżeli wciśnięty jest przycisk ładowania( 1 ), realizowane jest ładowanie licznika. W przeciwnym przypadku badany jest stan przycisku odpowiedzialnego za inkrementację licznika, co dzieje się już w pętli głównej programu, aby zniwelować możliwość nie zarejestrowania wciśnięcia w tym czasie przycisku reset.

Dodatkowo wprowadzona została flaga, która sprawdza czy w przypadku wciśnięcia przycisku reset, nie został także wciśnięty przycisk dekrementacji co mogłoby spowodować niekorzystny efekt natychmiastowej dekrementacji zaraz po puszczeniu przycisku reset.