



**Sprawozdanie z laboratorium
Architektury Komputerów**

Laboratorium numer: 2

Temat: Watchdog Timer, Przerwania

Wykonujący ćwiczenie:

- Patryk Wójtowicz

Studia dzienne I stopnia

Kierunek: Informatyka

Semestr: III

Grupa zajęciowa: Lab 15

Prowadzący ćwiczenie:

Dr inż. Mirosław

Omieljanowicz

.....

OCENA

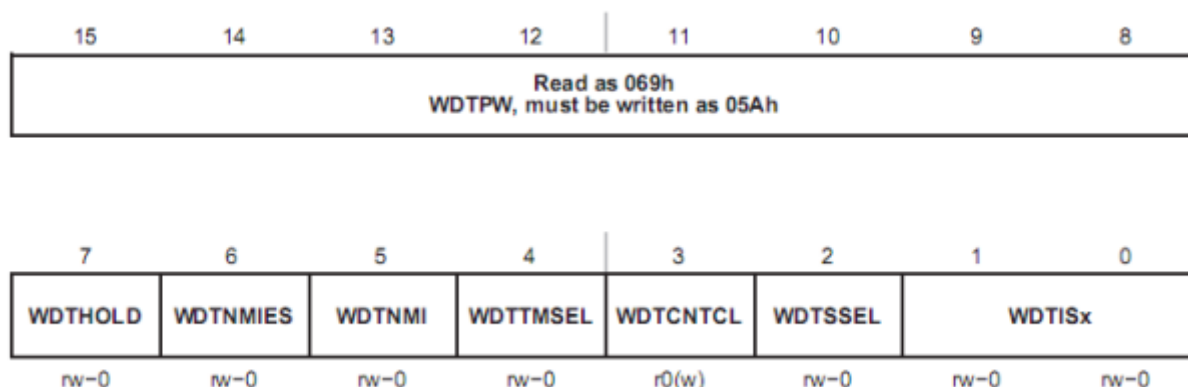
Data wykonania ćwiczenia

18.10.2021r.

.....

Data i podpis prowadzącego

Dodatkowo, moduł MSP430 jest zaopatrzony w 16 bitowy rejestr Watchdog Timer Control (WDTCTL), służy on do konfiguracji Watchdog Timera. Producenci zadbali także o kontrolę nad dostępem do tego rejestru poprzez zmienną Watchdog Timer Password (WDTPW), której wartość należy ustawić na 0x5ah, w przeciwnym wypadku będzie generowane przerwanie. Natomiast podczas odczytywania rejestru zawsze jest odczytywana wartość 069h.



Rysunek 2 Rejestr Watchdog Timer Control (WDTCTL)

Do zarządzania Watchdog Timerem służą następujące wpisy w rejestrze:

1. **WDTHOLD** - Watchdog Timer Hold. Steruje układem WDT. Bit 0 odpowiada za włączenie układu a bit 1 zatrzymuje jego działanie.
2. **WDTNMIES** – Watchdog Timer NMI Edge Select. Służy do ustawienia program w stan inkrementacji zmiennej WDT (bit 0) lub deinkrementacji (bit 1).
3. **WDTNMI** – Watchdog Timer NMI Select. Definiuje to pracę układu dla bitu 0 przyjmie funkcję resetu, dla bitu 1 przyjmie funkcję przerwania niemaskowanego.
4. **WDTTMSEL** – Watchdog Timer Mode Select. Konfiguruje tryb pracy WDT. Bit 0 włącza WDT, dla bitu 1 włączany jest tryb interval, który odmierza czas
5. **WDTCNTCL** – Watchdog Timer Counter Clear. Zerowanie rejestru dla bitu 1, dla bitu 0 nic się nie dzieje.
6. **WDTSSEL** – Watchdog Timer Clock Source Select. Odpowiada za wybór zegara na którym będzie pracować WDT. Bit 1 spowoduje pracę na zegarze ACLK, bit 0 wybierze SMCLK.
7. **WDTISx** – Watchdog Timer Interval Select. Bity ustawiają czas po którym flaga WDTIFG generuje przerwanie.

W trybie pracy Watchdog, za pomocą WDTCNTCL zerujemy licznik. Gdy minie ustalony czas i licznik nie zostanie wyzerowany, rejestr IFG1 zostanie ustawiony z flagą przerwania (WDTIFG) i wykona się reset mikrokontrolera. [2]

Założenia zadania

Założeniem zadania jest demonstracja obsługi przerwania przez Watchdog Timer'a. W przypadku resetu zasilania dioda P.2.1 (status) jest włączona, natomiast podczas przerwania, wywołanego wciśnięciem przycisku P.4.4, zapalenie diód P.1.5 oraz P.1.6 informuje nas o zaistniałym przerwaniu.

Realizacja zadania

```

1 // Program demonstruje działanie układu watchdog timer w trybie pracy watchdog
2 // Podczas pracy po resecie zasilania dioda Status jest zapalona
3 // W przypadku naciśnięcia klawisza następuje zawieszenie programu
4 // Układ watchdog restartuje program a użytkownikowi jest to oznajmiane po przez
5 // Zapalenie diód sygnalizujących P.1.5 oraz P.1.6
6
7 #include <msp430x14x.h>
8 #define KL1 BIT4&P4IN // Klawisz - P4.4
9 #define D1 BIT5&P1OUT // Dioda - P1.5
10 #define D2 BIT6&P1OUT // Dioda - P1.6
11
12 void main( void )
13 {
14     WDTCTL = WDTPW + WDTHOLD; // Wyłączenie układu WDT
15     P1DIR |= BIT5 + BIT6; // Ustawienie bitu P.1.5 i P.1.6 jako wyjście
16     P1OUT &= ~BIT5; // Zgaszenie diody P1.5
17     P1OUT &= ~BIT6; // Zgaszenie diody P1.6
18     P2DIR |= BIT1; // Ustawienie bitu P.2.1 jako wyjście
19     P2OUT &= ~BIT1; // Zapalenie diody P.2.1 (Status)
20     BCSCCTL1 |= XTS + DIVA1 + DIVA0; // (ACLK = LFXT1 = HF XTAL)/8
21
22     do
23     {
24         IFG1 &= ~OFIFG; // Czyszczenie flgi OSCFault
25     }
26     while ((IFG1 & OFIFG)); // Dopóki OSCFault jest ciągle ustawiona
27
28     WDTCTL = WDTPW + WDTCNTCL + WDTSSSEL ; // Tryb pracy watchdog ACLK
29
30     if(WDTIFG & IFG1) // Demonsrcja po przez zgaszenie diody P.2.1 (STATUS)
31     // oraz zapalenie diody P.1.5, P.1.6 że WDT przechwycił przerwanie
32     {
33         IFG1 &= ~WDTIFG; // Czyszczenie Flagi WDTIFG
34         P2OUT |= BIT1;
35         P1OUT |= BIT5;
36         P1OUT |= BIT6;
37     }
38

```

Rysunek 1 Kod programu cz.1

```

38
39     while(1)
40     {
41         if((KL1) == 0) // Sprawdzenie czy klawiesz jest wciśnięty
42         {
43             while(1); // Pętla nieskończona, zawieszenie progrmu
44         }
45         else
46         {
47             WDTCTL=WDTPW +WDTCNTCL; // Zerowanie WDT
48         }
49     }
50 }

```

Rysunek 3 Kod program cz.2

Wnioski

Przy włączonym układzie, Watchdog zapewnia ciągłość pracy mikrokontrolera. Realizuje to poprzez reset mikrokontrolera w przypadku wykrycia zawieszenia programu. Ma to nie bagatelny wpływ na pracę urządzenia które musi być w ciągłym działaniu. Rozwiązuje to problem kiedy to człowiek nie jest w stanie zareagować na daną „usterkę”. Ważnym aspektem konfiguracji MSP430 jest to aby wyłączyć WDT, ma to na celu ułatwienie emulowania programu który w niektórych przypadkach może się zawiesić i zostać zresetowany. Dlatego implementację Watchdoga zaleca się

dopiero w ukończonym programie.

Bibliografia

[1] Informator Laboratoryjny

[2] <https://pl.wikipedia.org/wiki/Watchdog>