

Programowanie mikrokontrolerów

Zerowanie, zarządzanie energią, taktowanie, bity konfiguracyjne

Marcin Engel Marcin Peczarski

Instytut Informatyki Uniwersytetu Warszawskiego

10 stycznia 2012

Rejestr MCUCSR

MCU Control and Status Register

7	6	5	4	3	2	1	0
			JTRF	WDRF	BORF	EXTRF	PORF

- ▶ Zawiera informację o przyczynie wyzerowania (ang. *reset*) mikrokontrolera:
 - ▶ JTRF – interfejs JTAG,
 - ▶ WDRF – nadzorca (ang. *watchdog*),
 - ▶ BORF – spadek napięcia zasilania,
 - ▶ EXTRF – wyprowadzenie RESET,
 - ▶ PORF – włączenie zasilania.
- ▶ Bity JTRF, WDRF, BORF, EXTRF są zerowane po włączeniu zasilania.
- ▶ Bit PORF jest ustawiany po włączeniu zasilania.
- ▶ Każdy z tych bitów można wyzerować przez wpisanie zera.

Watchdog – nadzorca

- ▶ Służy do wyprowadzania mikrokontrolera ze stanu zawieszenia.
- ▶ Nie zwalnia od pisania poprawnego kodu!
- ▶ Posiada swój własny oscylator o częstotliwości ok. 1 MHz.
- ▶ Licznik nadzorcy jest zerowany, gdy:
 - ▶ zostanie wykonana instrukcja [WDR](#),
 - ▶ mikrokontroler zostanie wyzerowany.
- ▶ Gdy licznik nadzorcy osiąga wartość maksymalną, to mikrokontroler jest zerowany.

Rejestr WDTCR

Watchdog Timer Control Register

7	6	5	4	3	2	1	0
			WDTOE	WDE	WDP2	WDP1	WDP0

- ▶ WDTOE – używany w procedurze wyłączenia nadzorcy (patrz następny slajd).
- ▶ WDE – służy do włączania i wyłączania nadzorcy.
- ▶ WDP2, WDP1, WDP0 – determinują czas zliczania. Jeśli t jest wartością zapisaną w tych bitach, to czas ten wynosi w przybliżeniu

$$2^{t+14} \mu s.$$

- ▶ Minimalny możliwy do uzyskania czas wynosi ok. 16 ms.
- ▶ Maksymalny możliwy do uzyskania czas wynosi ok. 2 s.

Wyłączanie układu nadzorcy

- ▶ Aby wyłączyć nadzorcę, należy wykonać następującą procedurę.

```
wdr
in r16, WDTCR
ori r16, 1 << WDTOE | 1 << WDE
out WDTCR, r16
ldi r16, 0
out WDTCR, r16
```

- ▶ Ma to na celu zapobieżenie przypadkowemu wyłączeniu.

Brown-out Detection

- ▶ Układ BOD służący do zerowania mikrokontrolera przy spadku napięcia zasilania.
- ▶ Zalecane jest jego użycie, gdy występują powolne zmiany napięcia zasilania.
- ▶ Do wyboru są dwa poziomy kontroli tego napięcia:
 - ▶ ok. 2,7 V,
 - ▶ ok. 4 V.
- ▶ Układ posiada histerezę: napięcie powodujące ponowne włączenie mikrokontrolera jest większe od napięcia wywołującego zerowanie.
- ▶ Mimo sporych rozrzutów produkcyjnych poziomu zadziałania układu BOD producent gwarantuje poprawne działanie mikrokontrolera przy napięciu zasilania większym od napięcia wywołującego zerowanie.

Konfiguracja Brown-out Detection

- ▶ Konfiguracja układu BOD odbywa się za pomocą bitów konfiguracyjnych mikrokontrolera (ang. *fuse bits*).
- ▶ Włączenie następuje przez zaprogramowanie (wyzerowanie) bitu BODEN.
- ▶ Bit BODLEVEL wybiera poziom zadziałania:
 - ▶ 2,7 V – niezaprogramowany (ustawiony na 1),
 - ▶ 4 V – zaprogramowany (ustawiony na 0).

Program demonstrujący sposoby zerowania (1)

- ▶ Skaczymy do początku programu.

```
.cseg  
.org 0  
        rjmp start
```

- ▶ Do wyprowadzenia INT2 podłączamy klawisz, którym będziemy zerować licznik nadzorcy.

```
.org INT2addr  
        wdr  
        reti
```

- ▶ Inicjujemy stos.

```
start:  
        ldi r16, low(RAMEND)  
        out SPL, r16  
        ldi r16, high(RAMEND)  
        out SPH, r16
```


Program demonstrujący sposoby zerowania (2)

- ▶ Do portu A podłączamy diody świeące (wyprowadzenia PA0 do PA4) i brzęczyk (wyprowadzenie PA7). Posłużą one do sygnalizacji wyzerowania i obserwacji jego przyczyny.

```
ldi r16, $FF  
out DDRA, r16
```

- ▶ Po wyzerowaniu wczytujemy informację o jego przyczynie.

```
in r16, MCUCSR
```

- ▶ Kasujemy informację o przyczynie zerowania.

```
mov r17, r16  
cbr r17, $1F  
out MCUCSR, r17
```

- ▶ Zaświecamy odpowiednie diody i włączamy brzęczyk.

```
andi r16, $1F  
com r16  
cbr r16, $80  
out PORTA, r16
```

Program demonstrujący sposoby zerowania (3)

- ▶ Wyłączamy brzęczyk po 250 ms.

```
ldi r25, high(25000)
ldi r24, low(25000)
rcall wait
sbr r16, $80
out PORTA, r16
```

Program demonstrujący sposoby zerowania (4)

- Konfigurujemy przerwanie INT2, aby było wyzwalane wciśnięciem klawisza (opadające zbocze).

```
sbi PORTB, PB2
ldi r16, 1 << INT2
out GICR, r16
sei
```

- Włączamy nadzorcę.

```
ldi r16, 1 << WDE | 1 << WDP2 |
          1 << WDP1 | 1 << WDP0
wdr
out WDTCR, r16
```

- Kręcimy się aż do wyzerowania.

```
forever:
    rjmp forever
```

Minimalizacja poboru mocy

- ▶ Jest bardzo istotna w aplikacjach zasilanych bateryjnie.
- ▶ Pobór mocy zależy:
 - ▶ liniowo od częstotliwości taktowania,
 - ▶ kwadratowo od napięcia zasilania.
- ▶ Zmniejszenie poboru mocy można uzyskać przez usypianie mikrokontrolera, gdy nie ma nic do roboty.
- ▶ Uśpienie może polegać na:
 - ▶ wyłączeniu taktowania jednostki wykonawczej, pamięci i niektórych układów peryferyjnych;
 - ▶ zmniejszeniu częstotliwości taktowania.
- ▶ Mikrokontrolery rodziny AVR różnią się nieco pod względem dostępnych trybów oszczędzania energii.
- ▶ Należy przeczytać odpowiedni rozdział w dokumentacji.
- ▶ Dalej skupimy się na ATmega16 i ATmega32.

Minimalizacja poboru mocy, cd.

- ▶ Jeśli zależy nam na szczególnie małym poborze prądu, to należy przede wszystkim:
 - ▶ zasilać mikrokontroler niższym napięciem, np. 3 V (patrz np. ATmega16L, ATmega16A, ATmega32L, ATmega32A);
 - ▶ zredukować częstotliwość taktowania;
 - ▶ używać trybu uśpienia;
 - ▶ w praktyce najlepszym rozwiązaniem może okazać się ustalenie jak największej częstotliwości taktowania i wybranie odpowiedniego trybu uśpienia – zgodnie z zasadą *zrobić szybko i jak najdłużej spać*.
- ▶ Zalecane są dodatkowe środki:
 - ▶ nie zostawiać „wiszących” wejść;
 - ▶ nie obciążać wyjść w trybie uśpienia;
 - ▶ wyłączyć w układzie docelowym interfejs JTAG i debugowanie w układzie – bity konfiguracyjne JTAGEN i OCDEN;
 - ▶ wyłączyć komparator – bit ACD w rejestrze ACSR;
 - ▶ nie włączać bez wyraźniej potrzeby: BOD, nadzorcy, przetwornika analogowo-cyfrowego, wewnętrznego napięcia referencyjnego.

Tryby uśpienia

- ▶ Umożliwiają wyłączenie nieużywanych układów i redukcję mocy pobieranej ze źródła zasilania.
- ▶ Wejście w tryb uśpienia następuje za pomocą instrukcji **SLEEP**.
- ▶ Mikrokontroler budzi się z trybu uśpienia, gdy zostanie zgłoszone przerwanie.
- ▶ Po obudzeniu mikrokontroler przechodzi do procedury obsługi przerwania.
- ▶ Po zakończeniu obsługi przerwania wykonywana jest instrukcja następna po instrukcji **SLEEP**, która wprowadziła w tryb uśpienia.
- ▶ Idealnym miejscem do spania jest główna pętla programu.

forever:

sleep

rjmp forever

Tryby uśpienia, cd.

- ▶ W trybach uśpienia zachowywana jest zawartość wszystkich rejestrów i pamięci operacyjnej.
- ▶ W zależności od sposobu taktowania budzenie może trwać od kilku taktów zegara do ok. 65 ms.
- ▶ Wyzerowanie zawsze powoduje obudzenie.
- ▶ Jest 6 trybów uśpienia różniących się peryferiami, które są wyłączane i źródłami przerwań, które mogą służyć do wybudzenia.

Rejestr MCUCR

MCU Control Register

7	6	5	4	3	2	1	0
SM2	SE	SM1	SM0				

- ▶ SE – musi być ustawiony na 1, aby instrukcja **SLEEP** wprowadziła mikrokontroler w tryb uśpienia.
- ▶ SM2, SM1, SM0 – wybierają jeden z sześciu trybów uśpienia.

Idle Mode

- ▶ SM2..0 są ustawione na 000.
- ▶ Zatrzymuje taktowanie CPU i pamięci.
- ▶ Działają: SPI, USART, TWI, komparator, przetwornik analogowo-cyfrowy, liczniki, nadzorca, BOD, system przerwań.
- ▶ Obudzenie może nastąpić dowolnym przerwaniem wewnętrznym lub zewnętrznym.
- ▶ Jeśli aktywny jest przetwornik analogowo-cyfrowy, to konwersja startuje automatycznie po wejściu w ten tryb.

ADC Noise Reduction Mode

- ▶ SM2..0 są ustawione na 001.
- ▶ Zatrzymuje taktowanie CPU, pamięci i portów wejścia-wyjścia.
- ▶ Działają: przetwornik-analogowo cyfrowy, przerwania zewnętrzne, TWI, licznik 2, nadzorca, BOD.
- ▶ Obudzenie umożliwiają przerwania: zerowanie, zakończenie przetwarzania ADC, TWI zgodność adresu, licznika 2, gotowość SPM i EEPROM, INT0 i INT1 wyzwalane poziomem, INT2.
- ▶ Jeśli aktywny jest przetwornik analogowo-cyfrowy, to konwersja startuje automatycznie po wejściu w ten tryb.
- ▶ Tryb służy do zmniejszenia poziomu zakłóceń przetwornika analogowo-cyfrowego.

Power-down Mode

- ▶ SM2..0 są ustawione na 010.
- ▶ Zatrzymuje większość oscylatorów.
- ▶ Działają tylko urządzenia asynchroniczne w stosunku do zegara systemowego.
- ▶ Działają: TWI, przerwania zewnętrzne, nadzorca, BOD.
- ▶ Obudzenie umożliwia przerwania: zerowanie, TWI zgodność adresu, INT0 i INT1 wyzwalane poziomem, INT2.
- ▶ Budzenie może trwać dość długo. Uruchomienie oscylatorów wymaga pewnego czasu. Patrz tabele 5, 6, 8, 10 i 12 w dokumentacji.

Power-save Mode

- ▶ SM2..0 są ustawione na 011.
- ▶ Tryb jest podobny do Power-down Mode.
- ▶ Jedyna różnica polega na umożliwieniu asynchronicznej pracy licznika 2.
- ▶ Obudzenie umożliwiają przerwania jak w Power-down Mode oraz przerwania licznika 2.

Standby Mode

- ▶ SM2..0 są ustawione na 110.
- ▶ Tryb wymaga taktowania zewnętrznym rezonatorem.
- ▶ Tryb jest podobny do Power-down Mode.
- ▶ Jedyna różnica polega na niewyłączaniu oscylatora.
- ▶ Umożliwia to obudzenie w 6 taktów zegara.

Extended Standby Mode

- ▶ SM2..0 są ustawione na 111.
- ▶ Tryb wymaga taktowania zewnętrznym rezonatorem.
- ▶ Tryb jest podobny do Power-save Mode.
- ▶ Jedyna różnica polega na niewyłączaniu oscylatora.
- ▶ Umożliwia to obudzenie w 6 taktów zegara.

Orientacyjny pobór prądu

- ▶ Napięcie zasilania 5 V
- ▶ Temperatura 25°C

Tryb pracy	1 MHz	6 MHz
Aktywny	1,8 mA	9 mA
Idle	0,7 mA	4 mA
Power-down, nadzorca nieaktywny	1 μ A	
Power-down, nadzorca aktywny	14 μ A	
Power-save, nadzorca nieaktywny	13 μ A	
Standby, nadzorca nieaktywny	55 μ A	150 μ A

Wybór źródła taktowania

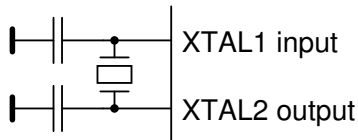
- ▶ Wybór sposobu taktowania mikrokontrolera odbywa się za pomocą bitów konfiguracyjnych (ang. *fuse bits*):
 - ▶ CKSEL3, CKSEL2, CKSEL1, CKSEL0
 - ▶ CKOPT
 - ▶ SUT1, SUT0
- ▶ Bit **zaprogramowany** ma wartość 0.
- ▶ Bit **niezaprogramowany** ma wartość 1.
- ▶ Chcąc zmienić bity konfiguracyjne, należy dokładnie sprawdzić w dokumentacji ustawiane wartości. Niewłaściwe zaprogramowanie bitów konfiguracyjnych związanych z taktowaniem może unieruchomić mikrokontroler i szeregowy interfejs programowania.

Źródła taktowania

Źródło taktowania	CKSEL3..0
Zewnętrzny rezonator kwarcowy lub ceramiczny	1111 – 1010
Zewnętrzny rezonator 32768 Hz	1001
Zewnętrzny układ RC	1000 – 0101
Wewnętrzny kalibrowany oscylator RC	0100 – 0001
Takt zewnętrzny	0000

Rezonator kwarcowy lub ceramiczny

- ▶ Schemat podłączenia



- ▶ Kondensatory powinny być jednakowe o wartości między 12 a 22 pF.
- ▶ Wartość pojemności zależy od zastosowanego rezonatora, pojemności montażowych i poziomu zakłóceń elektromagnetycznych.
- ▶ Należy zajrzeć do noty katalogowej rezonatora.

Rezonator kwarcowy lub ceramiczny, bity konfiguracyjne

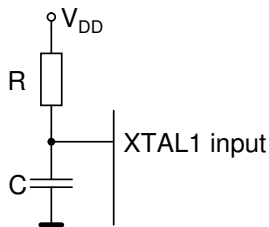
- ▶ Bit CKOPT wybiera tryb pracy oscylatora.
- ▶ Bit CKOPT **zaprogramowany** (wartość 0):
 - ▶ Oscylator działa w pełnym zakresie napięcia zasilania.
 - ▶ Jest zalecany, gdy występują silne zakłócenia.
 - ▶ Jest konieczny, jeśli z wyjścia XTAL2 ma być pobierany takt dla innych układów.
 - ▶ Zakres częstotliwości 1 do 16 MHz.
- ▶ Bit CKOPT **niezaprogramowany** (wartość 1):
 - ▶ Oscylator działa w ograniczonym zakresie napięć.
 - ▶ Redukuje moc pobieraną przez oscylator.
 - ▶ Zakres częstotliwości 0,4 do 8 MHz.
- ▶ Zależnie od częstotliwości i żądanego czasu startu oscylatora należy ustawić bity CKSEL3, CKSEL2, CKSEL1, CKSEL0, SUT1, SUT0 wg tabel 4 i 5 w dokumentacji.

Rezonator kwarcowy 32768 Hz

- ▶ Sposób podłączenia jest taki sam, jak innych rezonatorów kwarcowych.
- ▶ Bit CKOPT wybiera sposób podłączenia kondensatorów.
- ▶ Bit CKOPT **zaprogramowany** (wartość 0):
 - ▶ Podłączone są wewnętrzne kondensatory 36 pF.
 - ▶ Nie trzeba podłączać kondensatorów zewnętrznych.
- ▶ Bit CKOPT **niezaprogramowany** (wartość 1):
 - ▶ Wewnętrzne kondensatory są odłączone.
 - ▶ Trzeba podłączyć kondensatory zewnętrzne wg noty katalogowej.
 - ▶ Czasem mogą wystarczyć pojemności montażowe.
- ▶ Zależnie od żądanego czasu startu oscylatora należy ustawić bity SUT1, SUT0 wg tabeli 6 w dokumentacji.
- ▶ Nie należy mylić tej opcji z opcją taktowania licznika 2 rezonatorem 32768 Hz podłączonym do wyprowadzeń TOSC1 i TOSC2.

Zewnętrzny układ RC

- ▶ Schemat podłączenia



- ▶ Częstotliwość taktowania określa przybliżony wzór $\frac{1}{3RC}$.
- ▶ Pojemność musi mieć wartość co najmniej 22 pF.
- ▶ Zaprogramowanie bitu CKOPT włącza wewnętrzny kondensator 36 pF.
- ▶ Bity CKSEL3, CKSEL2, CKSEL1, CKSEL0, SUT1, SUT0 służą do wyboru zakresu częstotliwości i żądanego czasu startu oscylatora wg tabel 7 i 8 w dokumentacji.

Wewnętrzny kalibrowany oscylator RC

- ▶ Nie wymaga podłączania żadnych zewnętrznych elementów.
- ▶ Bity CKSEL3, CKSEL2, CKSEL1, CKSEL0 umożliwiają wybór jednej z częstotliwości: 1, 2, 4 lub 8 MHz wg tabeli 9 w dokumentacji.
- ▶ Bit CKOPT powinien pozostać niezaprogramowany.
- ▶ Bity SUT1, SUT0 służą do wyboru żądanego czasu startu oscylatora wg tabeli 10 w dokumentacji.
- ▶ Nowy mikrokontroler ma ustawiony wewnętrzny kalibrowany oscylator RC na 1 MHz i najdłuższy czas startu wynoszący 65 ms.

Kalibracja wewnętrznego oscylatora RC

- ▶ Kalibracja odbywa się przez zapisane do rejestru OSCCAL wartości zmieniającej częstotliwość oscylacji, patrz tabela 11 w dokumentacji.
- ▶ W mikrokontrolerze są zapisane cztery wartości kalibrujące dla poszczególnych częstotliwości.
- ▶ Wartości te zapewniają dokładność częstotliwości 3% przy zasilaniu 5 V, w temperaturze 25°C.
- ▶ Po wyzerowaniu do rejestru OSCCAL zapisywana jest wartość kalibrująca dla 1 MHz.
- ▶ Aby skalibrować oscylator dla innej częstotliwości, należy odczytać odpowiednią wartość kalibrującą i zapisać programowo do rejestru OSCCAL.

Odczyt wartości kalibrujących za pomocą PonyProg

- Uruchamiamy skrypt:

```
SELECTDEVICE ATMEGA16
```

```
READ-CALIBRATION 0x100 DATA 0
```

```
READ-CALIBRATION 0x101 DATA 1
```

```
READ-CALIBRATION 0x102 DATA 2
```

```
READ-CALIBRATION 0x103 DATA 3
```


Takt zewnętrzny

- ▶ Umożliwia taktowanie mikrokontrolera z zewnętrznego generatora.
- ▶ Sygnał taktujący należy podać na wyprowadzenie XTAL1.
- ▶ Zaprogramowanie bitu CKOPT włącza wewnętrzny kondensator 36 pF między wyprowadzenie XTAL1 a masę.
- ▶ Bity SUT1, SUT0 służą do wyboru żądanego czasu startu mikrokontrolera wg tabeli 12 w dokumentacji.
- ▶ Okres sygnału taktującego może zmieniać się między kolejnymi cyklami co najwyżej o 2%.

Inne bity konfiguracyjne

- ▶ Bit SPIEN włącza i wyłącza szeregowy interfejs programowania.
 - ▶ Fabrycznie jest zaprogramowany.
 - ▶ PonyProg nie umożliwia jego zmiany :-).
- ▶ Bit EESAVE włącza i wyłącza kasowanie EEPROM przy reprogramowaniu mikrokontrolera.
 - ▶ Fabrycznie jest niezaprogramowany – EEPROM jest kasowany razem z pamięcią FLASH.
 - ▶ Jego zaprogramowanie przydaje się, jeśli często zmieniamy zawartość FLASH, a chcemy zachowywać zawartość EEPROM.

Ochrona przed konkurencją

- ▶ Do ochrony przed odczytaniem i zmianą zawartości FLASH i EEPROM służą bity zabezpieczające LB2 (Lock2), LB1 (Lock1).
- ▶ LB2=1, LB1=1 – zabezpieczenie wyłączone.
- ▶ LB2=1, LB1=0 – zapis jest zablokowany.
- ▶ LB2=0, LB1=0 – zapis, odczyt i weryfikacja są zablokowane.
- ▶ Bity zabezpieczające należy zaprogramować po zaprogramowaniu pamięci i bitów konfiguracyjnych.
- ▶ W nowym mikrokontrolerze bity zabezpieczające są niezaprogramowane (ustawione na 1).
- ▶ Bity zabezpieczające mogą być skasowane (ustawione na 1) tylko przez skasowanie całej pamięci mikrokontrolera (ang. *chip erase*).

Ważne uwagi o bitach konfiguracyjnych

- ▶ Wszystkie zmiany bitów konfiguracyjnych należy wykonywać bardzo ostrożnie, aby nie zablokować możliwości programowania.
- ▶ Przypominamy, że bit konfiguracyjny
 - ▶ **zaprogramowany** ma wartość **0** i w PonyProg jest **zaznaczony**,
 - ▶ **niezaprogramowany** ma wartość **1** i w PonyProg jest **niezaznaczony**.
- ▶ Mikrokontroler można przywrócić do życia za pomocą programatora równoległego.