

(at)MEGArewolucja

„Lepsze jest wrogiem dobrego“ - to stara życiowa prawda. Przekonujemy się o tym czasami, gdy kupiwszy najnowszą wersję jakiegoś programu stwierdzamy, że zastosowane w nim skróty klawiaturowe są zupełnie inne niż te, do których przyzwyczailiśmy się przez lata. Walimy pięścią w stół, wszystkiego musimy się uczyć od nowa. Nie ma rady. Software jest szczególnie podatny na tego rodzaju działania. Okazuje się jednak, że nie tylko.

Ostatnio, w sposób dość przypadkowy, dowiedziałem się, że firma Analog Devices wycofała z dokumentacji najnowszej wersji mikrokontrolera ADuC812 opis interfejsu I²C. Mikrokontroler ten jest dostępny na rynku już kilka lat i z pewnością powstało do tej pory wiele aplikacji, w których wykorzystuje się ten interfejs do komunikowania z otoczeniem. Gdyby Analog zrezygnował z I²C, to naraziłby na kłopoty niejednego producenta sprzętu wykorzystującego ów mikrokontroler. Jak się okazało, całe zamieszanie spowodowane jest brakiem umowy licencyjnej między AD i Philipsem - właścicielem praw do I²C. Konstruktorów i użytkowników ADuC mogą uspokoić: według zapewnień przedstawiciela Analog Devices, stosowany obecnie interfejs jest i nadal będzie zgodny z I²C, pomimo tego, że nazywa się inaczej.

Znacznie poważniej wyglądają zmiany wprowadzone do rodziny atmelowskich mikrokontrolerów ATmega. Można je określić jako rewolucyjne. Widocznie liczba i jakość błędów w mikrokontrolerach dotychczasowych wersji była na tyle poważna, że firma zdecydowała się na bardzo radykalne posunięcia. Wpro-

Nowe mikrokontrolery Atmela

Tab. 1. Zmiany w nazwach bitów stosowanych w mikrokontrolerach ATmega163 i ATmega323

Nazwy bitów w mikrokontrolerach ATmega163 i ATmega323	Nazwy bitów w mikrokontrolerach ATmega16 i ATmega32	Rejestry I/O w mikrokontrolerach ATmega163 i ATmega323	Komentarz
PWMn(0)	WGMn0	TCCRn(A)	„A” i „0” tylko w 16-bitowym timerze
PWMn1	WGMn1	TCCRnA	
CTCn	WGMn2	TCCRn(B)	„B” tylko w 16-bitowym timerze
CHR9	UCSZ2	USCRB	dotyczy tylko ATmega16
OR	DOR	UCSRA	dotyczy tylko ATmega16
ASB	RWWSB	SPMCR	
ASRE	RWWSRE	SPMCR	
ADFR	ADATE	ADSCR	dotyczy tylko ATmega16

wadzano więc trzy nowe układy serii ATmega, które będą zastępowały produkowane do tej pory.

I tak: mikrokontroler ATmega32 zastępuje dotychczasowy ATmega232, ATmega16 zastępuje ATmega163 i ATmega8 zastępuje (uwaga!) AT90S4433. Taki sposób naprawiania błędów nie byłaby może nawet bardzo dokuczliwy, gdyby nie to, że nowe układy nie zachowują pełnej kompatybilności ze starymi. Zmiany dotyczą m.in. parametrów czasowych, zasad funkcjonowania niektórych bloków mikrokontrolera, znaczenia i nazw wybranych rejestrów, ustawień bezpieczników, poprawienia interfejsu JTAG, a także procedur programowania. Część nowości wynika ze zmian w technologii produkcji tych mikrokontrolerów. Szczegółowy opis poprawek można znaleźć w dokumentach AVR081, AVR083 i AVR084 oraz w notach katalogowych dostępnych na stronach firmowych Atmela (<http://www.atmel.com>).

Wiele punktów opisywanych dotychczas w erratach dla starych układów staje się nieaktualnych dla układów nowych. Trzeba również pamiętać, że niektóre błędy opisywane w erratach zostały poprawione w ostatnich seriach starych układów. Dotyczy to głównie mikrokontrolera ATmega163.

Poniżej zostaną omówione zmiany wprowadzone dla mikrokontrolerów ATmega16, ATmega32 i wybranych przypadków dotyczących mikrokontrolera ATmega8. Dalsza część artykułu będzie dotyczyła tylko ATmega8.



Rozszerzenie możliwości przerywania instrukcji mikrokontrolera

Przerwania w sposób całkowicie pewny mogą zmieniać sterowanie

bez względu na wykonywaną instrukcję mikrokontrolera ATmega16. Wyeliminowano możliwość powstawania martwych pętli, co zdarzało się w układzie starej wersji.



Zakończenie przerwania TWI w stanie power-down

Operacje interfejsu 2-wire w stanie power-down nie będą przerywane przez inne przerwania. Przerwania tego interfejsu nie spowodują powrotu do stanu Idle, jeśli nastąpią w stanie power-down.



TWI Master uodporniony na zakłócenia

Zastosowano filtr cyfrowy w celu wyeliminowania zakłóceń impulsowych powodujących błędną interpretację początku transmisji. Ponadto, jeśli warunek startu zostanie źle rozpoznany, zostanie wygenerowany status błędu (Bus Error) i ustawiony bit TWINT, gdy linia SDA przejdzie w stan idle. Poprzednio w takich sytuacjach powstawała martwa pętla.



Operacja zapisu TWCR jest ignorowana, jeśli jest powtarzana (bezpośrednio jedna po drugiej)

Kolejne zapisy do rejestru TWCR działają obecnie tak, jak byśmy tego oczekiwali. Nie są potrzebne instrukcje NOP pomiędzy poszczególnymi zapisami.



Poprawna faza sygnału PWM

W nowych mikrokontrolerach wprowadzono zmiany w generowaniu sygnału PWM pod kątem poprawności jego fazy.



Wyeliminowanie ograniczenia prędkości TWI w trybie Slave

Ograniczenie prędkości interfejsu TWI nie obowiązuje w nowych mikrokontrolerach. Częstotliwość taktowania CPU musi być w trybie slave co najmniej 16-krotnie większa niż częstotliwość sygnału SCL (patrz nota katalogowa).



Eliminacja problemów z ustawianiem UBRR

Zmiany ustawień prędkości transmisji powodują natychmiastowy efekt zarówno dla nadajnika, jak i odbiornika. Zapisując UBRR, nie trzeba zerować UBRRH.



Poprawki związane z OverRun Flag i Frame Error dla USART-a

Znacznik OverRun w mikrokontrolerze ATmega32 jest zawsze na bieżąco związany z FIFO, nie występują fałszywe błędy ramek sygnalizowane przez OverRun.



Zmiany nazw

W tab. 1 zestawiono zmiany nazw niektórych bitów, a w tab. 2 zmiany nazw rejestrów (cechy funkcjonalne pozostały bez zmian).



Poprawki timerów/liczników

W celu pełniejszego zaznajomienia się z wprowadzonymi w tym zakresie poprawkami należy zajrzeć do not katalogowych. Zmiany dotyczą:

- ustawiania wartości zliczanej w trybie PWM,
- rozszerzenia timera/licznika0 o funkcje compare i PWM,
- wprowadzenia opcji Frequency Correct dla timera/licznika1 pracującego w trybie PWM. Phase Correct pozostaje zachowana.

Tab. 2. Zmiany w nazwach rejestrów mikrokontrolerów ATmega163 i ATmega323

Stare nazwy rejestrów	Nowe nazwy rejestrów	Komentarz
ADCSR	ADCSRA	Dotyczy ATmega32
GIMSK	GICR	Dotyczy ATmega16
MCUSR	MCUCSR	Dotyczy ATmega16
UBRRHI	UBRRH	Dotyczy ATmega16 ta lokacja daje dostęp do dwóch rejestrów (patrz punkt dotyczący ulepszenia UART-u)



Poprawki dotyczące aktualizowania rejestru OCR w trybie PWM

Wartość zapisana do rejestru *Output Compare Register* w trybie PWM nie jest używana jako wartość porównywana, dopóki timer/licznik nie osiągnie wartości TOP. W mikrokontrolerach ATmega163 i ATmega323 nowa wartość OCR jest używana już w tym cyklu, w którym timer/licznik osiąga wartość TOP. W ATmega32 osiągnięcie wartości TOP powoduje dopiero aktualizację wartości porównywanej.



Udoskonalenie przetwornika A/C

Obecnie przetwornikiem analogowo-cyfrowym można wykonywać pomiary różnicowe ze wzmocnieniem.



Usprawnienia UART-a

Zastosowany w AT90S4433 i ATmega163 UART został zastąpiony przez USART w ATmega8 i ATmega16. Jest on kompatybilny z wcześniejszym z jednym wyjątkiem: zastosowano dwupoziomowy rejestr odbiorczy działający jak pamięć kolejkowa FIFO (*First In First Out*). Muszą być przestrzegane poniższe zasady:

- Funkcjonujące de facto dwa rejestry odbiorcze działają jak jeden rejestr cykliczny FIFO. Z tego powodu UDR musi być czytany tylko raz dla każdej przychodzącej danej. Ważne jest, że flagi błędów (FE i DOR - uwaga nowe nazwy!), a także dziewiąty bit danej (RXB8) są buforowane razem z daną w rejestrze odbiorczym. Dlatego bity statusowe muszą być sprawdzane zawsze przed odczytem rejestru UDR, w przeciwnym razie informacja o błędzie zostanie utracona w chwili, gdy zostanie utracona dana w buforze.
- Odbiorczy rejestr przesuwały działają teraz jak 3-poziomowy bufor. Jest tak, gdyż umożliwiono pozostawienie danej odbieranej w rejestrze przesuwałącym do czasu wykrycia nowego bitu startu, jeśli rejestry buforowe są pełne. Dzięki takim rozwiązaniom nowy USART jest bardziej odporny na błędy naczYTania danych (*Data OverRun* - DOR).

Rejestr UBRRHI jest umieszczony pod tym samym adresem, co dwa urządzenia. W ATmega8 i ATmega16 adres ten jest współdzielony z rejestrem UCSRC. Dostęp do UBRRH bądź

do UCSRC ustala bit URSEL. Po zerowaniu URSEL zostaje wybierany rejestr UBRRH - zachowana jest tym samym kompatybilność z AT90S4433 (ATmega163).

Jest jeszcze jedna, mniej istotna różnica w działaniu interfejsów. Po zainicjowaniu procesora, RXB8 przybierał stan „1” w AT90S4433 i ATmega163, a w USART-cie ATmega8 i ATmega16 jest on zerowany.



Zmiany parametrów elektrycznych

W nowych mikrokontrolerach zastosowano odmienną w stosunku do poprzednich technologii wytwarzania. Wskutek tego nastąpiły zmiany niektórych parametrów elektrycznych. Na przykład prąd Icc w stanie uśpienia (*Power Down Sleep Mode*) zwiększył się z ok. 4 µA dla do ok. 15 µA dla nowych układów. Szczegóły w notach katalogowych.



Zmiany timingu dla operacji zapisu pamięci EEPROM

Operacje zapisu pamięci EEPROM wymagają w nowych mikro-

kontrolerach 8448 cykli oscylatora (2048 w układach poprzednich). Oscylator RC jest kalibrowany na częstotliwość 1 MHz w obydwu układach. Na czas zapisu pamięci EEPROM można wpływać, wpisując odpowiednią wartość do zmiennej OSCCAL.



Zmiany dotyczące programowania

Zastosowano nowy algorytm programowania równoległego obsługujący stronicowanie pamięci EEPROM. Zmianie uległ przy tym timing tej operacji.



Nowe ustawienia bezpieczników

W tab. 3 przedstawiono porównanie ustawień bezpieczników dla układów starej i nowej wersji.



Oscylatory i ustawienie opóźnienia startu

Bezpieczniki CKSEL określają, który oscylator ma być aktywny, a także jak długi ma być opóźnienie startu. Szczegółów działania tych

Tab. 3. Porównanie ustawienia bezpieczników w układach ATmega163-ATmega16 i ATmega323-ATmega32⁽¹⁾

Bezpiecznik	Domyślne ustawienie w mikrokontrolerach ATmega163 i ATmega323	Domyślne ustawienie w mikrokontrolerach ATmega16 i ATmega32	Ustawienie kompatybilne w mikrokontrolerach ATmega163 i ATmega323
OCDEN	-	1	1
JTAGEN	- (ATmega163)		
0 (ATmega323)	0	1 ^(2a) (ATmega163)	
0 (ATmega323)			
SPIEN	0	0	0
CKOPT	-	1	0 ⁽³⁾ (ATmega163)
0 ^(2b) (ATmega323)			
EESAVE	1	1	1
BOOTSZ1	1	0	1
BOOTSZ0	1	0	1
BOOTRST	1	1	1
BODLEVEL	1	1	1
BODEN	1	1	1
SUT1	-	1	patrz uwaga (4)
SUT0	-	0	patrz uwaga (4)
CKSEL3	0	0	patrz uwaga (4)
CKSEL2	0	0	patrz uwaga (4)
CKSEL1	1	0	patrz uwaga (4)
CKSELO	0	1	patrz uwaga (4)

Objaśnienia:
 1. Kreska oznacza, że bezpiecznik nie występuje
 2a. Patrz punkt "Interfejs JTAG"
 2b. Patrz punkt "Oscylatory i ustawienie opóźnienia startu"
 3. Patrz punkt "Oscylatory i ustawienie opóźnienia startu"
 4. Bezpiecznik CKSEL jest dostępny zarówno w ATmega323, jak i w ATmega32, jakkolwiek działanie SUT i CKSEL powinno być przeanalizowane przy przejściu na mikrokontroler ATmega32. Patrz punkt *Oscylatory i ustawienie opóźnienia startu*.

Tab. 4. Zmiany nazw bitów w mikrokontrolerze ATmega8

Nazwa bitu w mikrokontrolerze AT90S4433	Nazwa bitu w mikrokontrolerze Atmega8	Rejestr I/O (AT90S4433)	Komentarz
PWM10	WGM1n0	TCCR1A	
PWM11	WGM1n1	TCCR1A	
CTC1	WGM12	TCCR1B	
CHR9	UCSZ2	USCRB	
OR	DOR	USCRA	
OCIE1	OCIE1A	TIMSK	zmieniono pozycję bitu
OCF1	OCF1A	TIFR	zmieniono pozycję bitu
WDTOE	WDCE	WDTCR	nie zachowano bezpośredniej kompatybilności
AINBG	ACBG	ACSR	

bezpieczników w nowych mikrokontrolerach należy szukać w notach katalogowych, w rozdziale *System Clock and Clock Option*.

Wykorzystanie oscylatora jako źródła przebiegów zegarowych poprzez wyjście XTAL2 jest w starych mikrokontrolerach możliwe bez żadnych ograniczeń. W nowych należy w tym celu zaprogramować bezpiecznik CKOPT. Trzeba pamiętać o zwiększonym poborze mocy w takiej konfiguracji. Wyjście pracuje w trybie *rail-to-rail* (w pełnym zakresie napięcia zasilającego).



Zmiany w timerze-watchdogu

Częstotliwość pracy watchdogu w nowych mikrokontrolerach dochodzi do 1 MHz bez względu na wartość napięcia zasilającego (poprzednio było to możliwe tylko dla napięcia równego 5 V).



Interfejs JTAG

Zastosowany w ATmega16 interfejs JTAG może być wykorzystywany do programowania, testowania krawędziowego (*boundary-scan*) i debugowania. W celu umożliwienia programowania mikrokontrolera poprzez interfejs JTAG, bezpiecznik JTAGEN powinien być ustawiony. Aby zachować kompatybilność z ATmega163, bezpiecznik ten powinien być wyzerowany. Włączenie interfejsu powoduje zajęcie czterech linii portów I/O na jego potrzeby.



Ulepszenie interfejsu JTAG

Mikrokontrolery ATmega323 i ATmega32 wykorzystują interfejs JTAG do programowania, skanowania krawędziowego i debugowania. W po-

równaniu z układem ATmega323, w ATmega32 zwiększono liczbę skanowanych komórek, dzięki czemu znacznie poprawiono m.in. obsługę funkcji analogowych. Wprowadzono ponadto nową instrukcję BREAK, która jest niezwykle przydatna podczas debugowania programu. Dzięki niej liczba ustawianych pułapek nie jest niczym limitowana.



Zmiany w operacji samoprogramowania

Samoprogramowanie było dostępne w mikrokontrolerach starych wersji i zachowano je oczywiście w nowych. Wprowadzono jednak pewne zmiany w zachowaniu się CPU podczas tej operacji. Obecnie CPU jest zatrzymywane jedynie podczas programowania (*No-Read-While-Write* - NRWW) pamięci Flash. Bit SPMEN w rejestrze SPMCR jest automatycznie zerowany w obu wersjach mikrokontrolera.



Inne

Nowe mikrokontrolery wyposażono w odmienne w porównaniu ze starymi sygnatury elektroniczne.

ATmega8

Użytkownicy mikrokontrolera AT90S4433 muszą powoli „przesiadać się” na ATmega8, który jest nowocześniejszą i poprawioną wersją poprzednika. Przesiadka jest możliwa nawet w aplikacjach już istniejących, aczkolwiek z pewnymi zastrzeżeniami, gdyż i w tym przypadku nie zachowano pełnej kompatybilności układów. Niewątpliwa przewaga mikrokontrolerów ATmega nad poprzednikami jest związana z większą pojemnością pamięci Flash, SRAM i EEPROM, nowymi

komponentami wewnętrznymi i nowymi cechami użytkowymi. Zachowano tę samą topografię wyprowadzeń. Wymieniając jednak mikrokontroler w działającym urządzeniu, trzeba się upewnić, czy różnice między wersjami układów na to pozwalają. Przeszkodą mogą być choćby nieco inne zasady dostępu do rejestrów, nie mówiąc już o różnicach w wartościach parametrów elektrycznych. Punkty budzące ewentualne wątpliwości w odniesieniu do układu AT90S4433 zostaną omówione pokrótce poniżej (dotyczą rozwiązań wykorzystywanych w ATmega8).

Bezpieczniki i tryby programowania

W mikrokontrolerze ATmega8 umożliwiono programowanie pamięci Flash i EEPROM w trybie programowania szeregowego po zaprogramowaniu bezpieczników.

Niepoprawne zmienianie kanałów w trybie Free Running

W mikrokontrolerze ATmega8 bity MUXn i REFS1/0 rejestru ADMUX są buforowane przez rejestr tymczasowy. Dzięki temu CPU może w sposób swobodny zmieniać kanały w bezpiecznym momencie konwersji (bez zakłóceń). Szczegóły w nocie katalogowej.

Czas stabilizacji źródła napięcia referencyjnego

W mikrokontrolerze ATmega8 czas ustabilizowania się źródła napięcia referencyjnego jest określony parametrami zamieszczonymi w nocie katalogowej niezależnie od tego czy układ *Brown-out Detector* jest włączony, czy nie.

Działanie układu Brown-out

Działanie detektora Brown-out jest obecnie niewrażliwe na aktywność wyprowadzeń I/O.

Programowanie szeregowo przy zasilaniu poniżej 2,9 V

Jeśli układ ATmega8 pracuje w warunkach zgodnych ze specyfikacją techniczną (dotyczy napięcia i częstotliwości pracy), nie powinny występować żadne problemy z programowaniem szeregowym.

Tab. 5. Zmiany nazw rejestrów w mikrokontrolerze ATmega8

Nazwy rejestrów w AT90S4433	Nazwy rejestrów w ATmega8	Komentarz
SP	SPL	
GIMSK	GICR	
MCUSR	MCUCSR	
UBRRHI	UBRRH	zmieniono adres rejestru

Tab. 6. Zmiany lokacji bitów w ATmega8

TIMSK	TIFR	Pozycja bitu w	
		AT90S4433	ATmega8
TOIE1	TOV1	7	2
OC1E1	OCF1	6	4
TCIE1	ICF1	3	5
TOIE0	TOV0	1	8

Utrata synchronizacji przez UART, jeśli linia RxD jest na poziomie niskim, podczas gdy UART jest zablokowany

Podobnie jak w mikrokontrolerach ATmega16 i ATmega32, w ATmega8 również zastąpiono UART przez USART. Wyeliminowano przy tym błąd utraty synchronizacji w opisywanej sytuacji.

Różnice w wyprowadzeniach układów AT90S4433 i ATmega8

W mikrokontrolerze ATmega8 wprowadzono nowe, w stosunku do AT90S4433, funkcje alternatywne niektórych wyprowadzeń I/O. Jeśli pozostaną niewykorzystywane, nie będą stwarzały problemów z kompatybilnością.

PC6 (!Reset)

Układ AT90S4433 posiada dedykowane wyprowadzenie !RESET. Niestety w ATmega8 funkcja zerowania jest przypisana do wyprowadzenia PC6, którego funkcjonowanie zależy od ustawienia bezpiecznika RSTDISBL.

PB7/6(XTAL/OSC)

ATmega8 nie ma dedykowanych wyprowadzeń XTAL1 i XTAL2. Funkcja XTAL2 i TOSC2 jest aktualnie dzielona z wyprowadzeniem PB7, natomiast XTAL1 i TOSC1 z PB6. Konfigurację ustala bezpiecznik CKSEL.

ADC7/6

W układzie ATmega8 w obudowie TQFP (i MLF) wyprowadzenia ADC6 i ADC7 zostały dołączone do wyprowadzeń 19 i 22. W AT90S4433 były one niewykorzystywane.

Vcc/GND

W układzie ATmega8 w obudowie TQFP (i MLF) wprowadzono dodatkową parę wyprowadzeń zasilających: Vcc - nóżka 6 i GND - nóżka 3. W AT90S4433 były one niewykorzystywane. Połączenie tych wyprowadzeń zgodnie z przeznaczeniem zwiększa odporność na szумы, jednak w przypadku zastępowania układu AT90S4433 przez ATmega8, powinny pozostać niepołączone.

Zmiany nazw

W tab. 5 i 6 przedstawiono zmiany w nazewnictwie bitów i rejestrów.

Zmiany w timerach/licznikach i preskalerach

Najważniejsze zmiany dotyczące timerów/liczników wprowadzone w ATmega8 to:

- możliwość zerowania,
- ustawianie wartości zliczanej w trybie PWM,
- wprowadzenie korekcji fazy i częstotliwości w trybie PWM dla timer/licznika1 i 2.
- tryb szybkiej PWM

Położenie bitów w rejestrach TIMSK i TIFR

Wszystkie maski i flagi przerwań mikrokontrolera AT90S4433 zostały uwzględnione w układzie ATmega8, przy czym zmieniono ich lokację. Zmiany przedstawiono w tab. 6.

Zerowanie TCNT1 w trybie PWM

Rozdzielczość modulacji PWM może być równa 8, 9 lub 10 bitów, jednakże wykorzystywane przy tym timer/licznik jest 16-bitowy. Mikrokontroler AT90S4433 umożliwiał wpisanie do rejestru TCNT1 liczby przewyższającej rozdzielczość, co mogło powodować nieprawidłowości w modulacji. W układzie ATmega8 efekt ten został wyeliminowany poprzez zerowanie niewykorzystywanych bitów.

Zerowanie OCR1xH w trybie PWM

W mikrokontrolerze AT90S4433 wykorzystującym PWM, sześć bardziej znaczących bitów OCR1xH było zawsze zerowanych, bez względu na wykorzystywaną rozdzielczość modulacji, co mogło prowadzić do błędnego działania układu. W ATmega8 poprawiono ten błąd - zerowane są tylko te bity, które wynikają z ustawionej rozdzielczości. **Zerowanie timera/licznika1 w trybie porównania z preskalerem**

Timer/licznik1 układu AT90S4433 wykazywał błędy podczas pracy w trybie porównywania z preskalerem. Licznik był w nim zerowany natychmiast po zliczeniu do stanu zapisanego w rejestrze OCR1 bez względu na ustawienia preskalera. W ATmega8 jest on zerowany dopiero w momencie n-tego zliczenia, gdzie n zależy od ustawienia preskalera.

Ustawienia wyprowadzeń i znaczników w trybie porównania z preskalerem (dotyczy wszystkich timerów/liczników)

Przypadek ten jest podobny do poprzedniego, dotyczy tylko momentu ustawiania znaczników i wyjścia wyniku porównania.

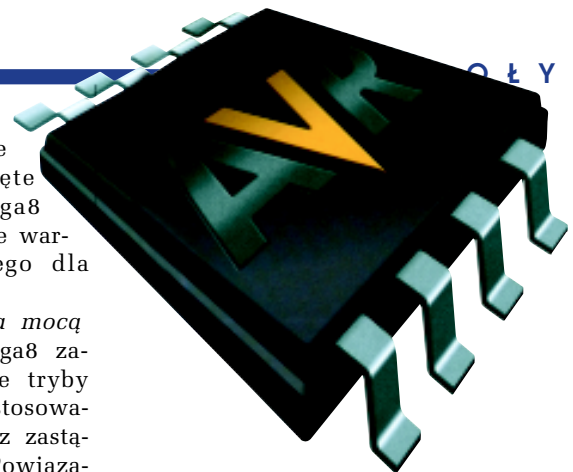
Aktualizacja OCR1x w trybie PWM

Pomiędzy układami AT90S4433 i ATmega8 występują w pewnych przypadkach różnice dotyczące aktu-

Tab. 7. Porównanie bezpieczników w układach AT90S4433 i ATmega8⁽¹⁾

Bezpiecznik	Domyślne ustawienie w AT90S4433	Domyślne ustawienie w ATmega8	Ustawienie kompatybilne w ATmega8
RSTDISBL	-	1	1
WDTON	-	1	1 ⁽²⁾
SPIEN	0	0	0
CKOPT	-	1	0 ⁽³⁾
EESAVE	-	1	1
BOOTSZ1	-	0	0 (N/A) ⁽⁴⁾
BOOTSZ0	-	0	0 (N/A) ⁽⁴⁾
BOOTRST	-	1	1
BODLEVEL	1	1	1
BODEN	1	1	1
SUT1	-	1	patrz uwaga (5)
SUT0	-	0	patrz uwaga (5)
CKSEL3	-	0	patrz uwaga (5)
CKSEL2	0	0	patrz uwaga (5)
CKSEL1	1	0	patrz uwaga (5)
CKSELO	0	1	patrz uwaga (5)

Objaśnienia:
 1. Kreska oznacza, że bezpiecznik nie występuje
 2. Patrz punkt "Zmiany w timerze watchdog"
 3. Patrz punkt "Oscylatory i ustawienie opóźnienia startu"
 4. SPM i samoprogramowanie nie występowało w AT90S4433. Ustawienie domyślne jest dobre, jeśli projekt jest dostosowany do ATmega8.
 5. Działanie SUT i CKSEL powinno być przeanalizowane przy przejściu na mikrokontroler ATmega8. Patrz punkt *Oscylatory i ustawienie opóźnienia startu*.



alizacji rejestru OCR1. Zapisanie nowej wartości do OCR1 odbywa się poprzez rejestr tymczasowy (nie jest natychmiastowe). Jeśli OCR1A jest czytany przed dokonaniem aktualizacji (dotyczy AT90S4433), faktycznie pobierana jest wartość z bufora OCR1A. Może to być przyczyną używania do późniejszych porównań innej wartości niż odczytana. W mikrokontrolerze ATmega8 rejestr OCR1x jest odczytywany przed jego aktualizacją - nie korzysta się z dodatkowego bufora.

Zapamiętywanie stanu wyjścia OCnx

Poziom wyprowadzenia OCnx w mikrokontrolerze ATmega8 jest zapamiętywany przed zablokowaniem trybu Output Compare. Ponowne jego odblokowanie przywraca zapamiętywany stan. Wszystkie wyjścia wyniku porównań są ustawiane na zero po wyzerowaniu mikrokontrolera.

Zmiany dotyczące przetwornika analogowo-cyfrowego

Przetwornik A/C został w mikrokontrolerze ATmega8 ulepszony w porównaniu z AT90S4433 o:

- obsługę 10-bitowych wyników adjustowanych w lewo lub w prawo,
- wyposażenie układu w dwa dodatkowe kanały ADC (tylko dla obudów MLF i TQFP).

Projektując urządzenia wykorzystujące przetworniki A/C w oparciu o mikrokontroler ATmega8, trzeba pamiętać, że kanały ADC4 i ADC5 charakteryzują się w nim mniejszą dokładnością niż pozostałe. Niedogodność ta wynika ze współdzielenia wyprowadzeń przypisanych do tych kanałów z interfejsem 2-wire. Konsekwencją takiego rozwiązania jest konieczność zasilania ich z napięcia Vcc, nie AVCC. Parametry elektryczne zamieszczono w notach katalogowych.

Ustawienie bitu ADCBG w rejestrze ADMUX powodowało w mikrokontrolerze AT90S4433 dołączenie źródła referencyjnego (1,22 V) do wybranego wejścia przetwornika A/C. Ten sam efekt jest osiągany w ATmega8 przez ustawienie bitów

MUX3...0 na 1110b. Miejsce bitu ADCBG zostało zajęte w mikrokontrolerze ATmega8 przez bity REFS1/0 ustalające wartość napięcia referencyjnego dla przetwornika A/C.

Zmiany systemu zarządzania mocą

W mikrokontrolerze ATmega8 zaimplementowano dodatkowe tryby uśpienia. Tak więc bit SM stosowany w AT90S4433 został teraz zastąpiony przez bity SM2...0. Powiązania między starymi i nowymi ustawieniami są następujące:

- SM=0 (AT90S4433) odpowiada SM2...0=000b (ATmega8)
- SM=1 (AT90S4433) odpowiada SM2...0=010b (ATmega8)

Ważne jest ponadto, że zmieniono położenie bitu SE. Szczegóły w notach katalogowych.

Zmiany w SPI i USART

Obydwa interfejsy mogą obecnie pracować z podwójną prędkością. UART stosowany w AT90S4433 jest zastąpiony USART-em w ATmega8. Zmiany są identyczne jak w mikrokontrolerach ATmega163 i ATmega323.

Zmiany w interfejsie programowania

W mikrokontrolerze ATmega8 wprowadzono kilka udoskonaleń interfejsu programowania. Szczególnie ważne są te, które dotyczą interfejsu *In-System Programming*. Zmiany są związane z wprowadzeniem dodatkowych bezpieczników w układzie ATmega8. Szczegóły w notach katalogowych.

Zmieniono również algorytm programowania równoległego. Obecnie wykorzystuje się dodatkowe wyprowadzenia PAGEL (PD7) i BS2 (PA0) niezbędne do prawidłowego programowania pamięci Flash, EEPROM i dodatkowych bezpieczników. Niestety uległ zmianie również timing operacji programowania. Szczegóły w notach katalogowych.

Ustawienia bezpieczników

W **tab. 7** opisano wszystkie zmiany związane z bezpiecznikami stosowanymi w mikrokontrolerze ATmega8.

Oscylatory i ustawienie opóźnienia startu

Mikrokontroler ATmega8 wykorzystuje więcej oscylatorów i ustawień czasu startu niż AT90S4433. Podczas oczekiwania na wyjście z trybu Power-down może korzystać z częstotliwości CPU, podczas gdy AT90S4433 wykorzystywał do tego celu jedynie oscylator watchdoga.

Więcej informacji związanych z fazą *Start-up* mikrokontrolera jest zawartych w rozdziale *System Clock and Clock Options*.

Szczególną uwagę należy zwrócić na zmiany ustawień bezpieczników w trybie ISP, który jest zależny od zegara systemowego. Złe ustawienie oscylatora może uniemożliwić ponowne zastosowanie programowania szeregowego. Wówczas będzie możliwe jedynie zastosowanie programowania równoległego. Wykorzystanie oscylatora jako źródła przebiegów zegarowych poprzez wyjście XTAL2 jest w AT90S4433 możliwe bez żadnych ograniczeń. W ATmega8 należy w tym celu zaprogramować bezpiecznik CKOPT. Trzeba pamiętać o zwiększonym poborze mocy w takiej konfiguracji. Wyjście pracuje w trybie *rail-to-rail* (w pełnym zakresie napięcia zasilającego).

Zmiany w timerze watchdog

Watchdog mikrokontrolera ATmega8 jest udoskonaloną wersją stosowanego w AT90S4433. Jego włączenie lub wyłączenie nie jest tak łatwe jak w pierwotnym układzie. Obecnie zastosowano dwupoziomowe zabezpieczenie, ustawiane bezpiecznikiem WDTON (patrz **tab. 8**). Częstotliwość pracy watchdoga w ATmega8 dochodzi do 1MHz bez względu na wartość napięcia zasilającego (poprzednio obowiązywało to tylko dla napięcia równego 5 V). Przed zastosowaniem watchdoga koniecznie trzeba zapoznać się z notą katalogową.

Na podstawie materiałów firmy Atmel opracował

Jarosław Doliński, AVT
jaroslaw.dolinski@ep.com.pl

Tab. 8. Porównanie ustawienia watchdoga w AT90S4433 i ATmega8					
Układ	WDTON	Poziom zabezpieczenia	Stan początkowy WDT	Jak zablokować WDT?	Jak zmienić time-out?
AT90S4433	N/A	N/A	zablokowany	sekwencja czasowa	bez restrykcji
ATmega8	niezaprogramowany	1	zablokowany	sekwencja czasowa	sekwencja czasowa
ATmega8	zaprogramowany	2	odblokowany	zawsze odblokowany	sekwencja czasowa