

MSP430

Moduł Timer_A, układ Watchdog



Kontynuujemy cykl artykułów na temat procesora MSP430 firmy Texas Instruments. Ten będzie poświęcony tytułowemu modułowi Timer_A, oraz układowi Watchdog. Uzupełnieniem treści artykułu są programy umieszczone na płycie CD. Wszystkie przykłady programów i konfiguracje sprzętowe wykonano w oparciu o dostępną w ofercie AVT płytkę ewaluacyjną eMeSPek.

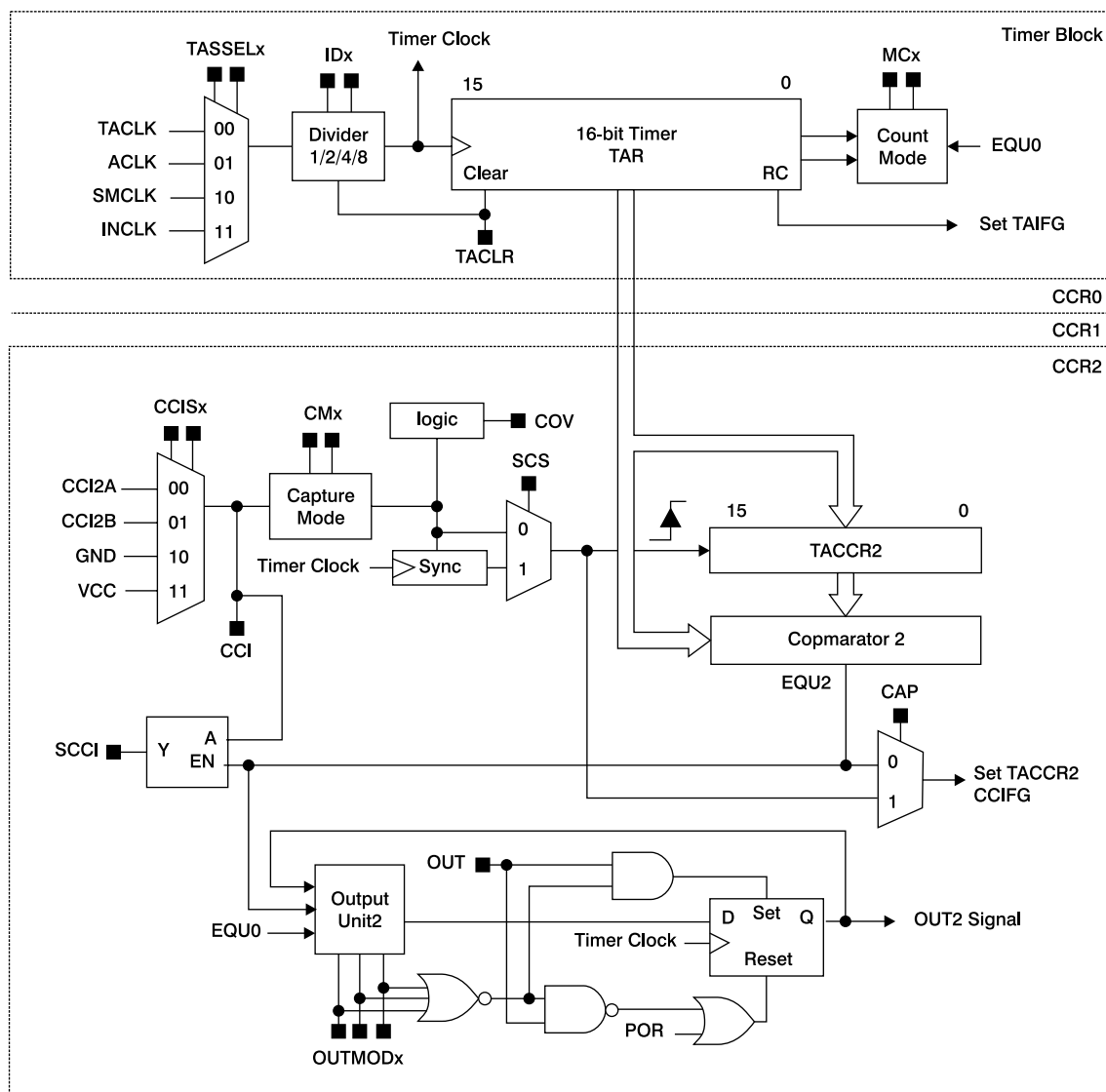
Programy będące uzupełnieniem treści artykułu, prezentują efekty swojej pracy na wyświetlaczu alfanumerycznym LCD (2x16 znaków, sterownik zgodny z HD44780). Płytkę ewaluacyjną eMeSPek, posiada wyprowadzone złącze wyświetlacza, jednak aby móc

sterować wyświetlaczem, należy odpowiednio skonfigurować płytke. Zworki JP1 i JP2 powinny należy połączyć w pozycji 1-2. Wówczas to linie EN i RS wyświetlacza zostaną doprowadzone do mikrokontrolera i możliwa będzie jego komunikacja z modułem wyświetlacza.

Kontrast ustawia się przy pomocy potencjometru PR1.

Współdzielony JTAG

Mikrokontroler MSP430f1232, nie posiada oddzielnych linii dla interfejsu JTAG i podczas programowania/debugowania wykorzystywane są linie portów I/O mikrokontrolera (P1.4...P1.7). Projektując płytke można postąpić w dwojaki sposób: albo linie interfejsu JTAG podłączyć jedynie do złącza JTAG, albo dodatkowo podłączyć je do wyprowadzeń urządzeń peryferyjnych. Na płytce eMeSPek zastosowano rozwiązanie drugie i linie JTAG podłączono do



Rys. 1.

Tab. 1. Tryby pracy licznika TAR.

Tryb pracy.	Opis.
Zatrzymanie. (Stop Mode MCx = 00)	Licznik TAR jest zatrzymany. Wraz z zatrzymaniem licznika, zachowywana jest wartość TAR, oraz wartości rejestrów konfigurujących moduł Timer_A.
Licznie w górę. (Up Mode MCx = 01)	Licznik TAR cyklicznie zlicza od zera do wartości wpisanej do rejestru TACCR0 (Rys. 2a). Odliczane jest TACCR0 + 1 cykli sygnału taktującego. W momencie, gdy wartość TAR zmienia się z TACCR0 - 1 na TACCR0, to w rejestrze TACCTLO ustawiana jest flaga CCIFG. Z następnym narastającym zboczem sygnału taktującego zerowana jest wartość TAR oraz ustawiana jest flaga TAIFG (Rys. 2b).
Praca ciągła. (Continous Mode MCx = 10).	W trybie tym licznik TAR cyklicznie zlicza od zera do wartości 0xFFFF (Rys. 3a). W chwili, gdy następuje przekroczenie licznika (zmiana wartości z 0xFFFF na 0x0000), ustawiana jest flaga TAIFG (Rys. 3b).
Licznie dwukierunkowe. (Up/Down Mode MCx = 11).	Od pozostałych różni się tym, że wartość TAR poza zwiększaniem jest również zmniejszana. Mianowicie licznik TAR cyklicznie zlicza od zera do wartości wpisanej do rejestru TACCR0, a następnie, gdy osiągnie tę wartość rozpoczyna się zmniejszanie jego wartości do zera (Rys 4a). Cały cykl trybu Up/Down trwa 2*TACCR0 cykli sygnału taktującego. W momencie, gdy wartość TAR zmienia się z TACCR0 - 1 na TACCR0 (połowa cyklu), to w rejestrze TACCTLO ustawiana jest flaga CCIFG. Z kolei flaga TAIFG jest ustawiana przy zmianie wartości TAR z 0x0001 na 0x0000 (Rys. 4b).

Tab. 2. Rejestry TACCRx, tryb PWM, konfiguracja sygnałów dla MSP430f1232 (obudowa DW, oraz PW)

Rejestr TACCRx	Tag CCISx, konfigurujący wyjście OUTx	Wyjście OUTx Nóżka/Pin
TACCR0	CC10A	22/P1.1
	CC10B	26/P1.5
	GND	10/P2.2
TACCR1	Vcc	Wewnętrzne ADC10
	CC11A	23/P1.2
	CC11B	27/P1.6
TACCR2	GND	19/P2.3
	Vcc	Wewnętrzne ADC10
	CC12A	24/P1.3
TACCR2	CC12B	28/P1.7
	GND	20/P2.4
	Vcc	Wewnętrzne ADC10

List. 1. Szablony procedur obsługi przerwania – TACCR0, TAIV, WDT

```

//wektor TACCR0
#pragma vector=TIMER0_VECTOR
__interrupt void Timer_A0 (void)
{
    ... // CCR0
}

//wektor TAIV
#pragma vector=TIMER1_VECTOR
__interrupt void Timer_A1(void)
{
    switch( TAIV )
    {
        case 2: //CCR1
            break;
        case 4: //CCR2
            break;
        case 10: //przepełnienie
            break;
    }
}

// wektor układu Watchdog
#pragma vector=WDT_VECTOR
__interrupt void watchdog_timer(void)
{
    ...
}
    
```

2, 4 lub 8. Decydują o tym nastawy bitów IDx. Każde narastające zbocze sygnału zegarowego powoduje zmianę stanu licznika TAR zależnie od wybranego trybu pracy. W dowolnym momencie programista może odczytać wartość licznika TAR, jak również ją zmodyfikować.

Ustawienie bitu TACLRL powoduje zerowanie licznika. Należy jednak pamiętać o tym, że wraz z licznikiem zerowane są bity IDx nastawy preskalera oraz w przypadku trybu pracy Up/Down, kierunek zliczania impulsów.

Odpowiednia konfiguracja modułu Timer_A daje możliwość generowania zdarzeń czasowych, impulsów o regulowanym czasie trwania (PWM), czy też przechwytywania sygnału taktującego rejestr TACCRx (tryb capture). „Większe”

mikrokontrolery serii 1xx, poza modułem Timer_A, posiadają identyczny funkcjonalnie moduł Timer_B. W module tym licznik, współpracuje z siedmioma rejestrami porównującymi / przechwytyjącymi, co znacząco wzbogaca funkcje mikrokontrolera.

Tryby pracy licznika TAR

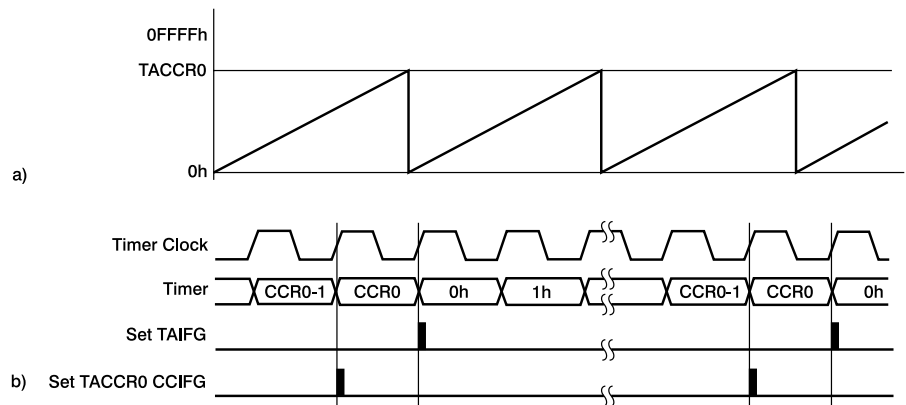
Licznik TAR ma możliwość pracy w jednym z czterech trybów (tab. 1). Wyboru trybu pracy dokonujemy programowo modyfikując wartości bitów MCx w rejestrze TACTL. Specyficznym trybem pracy jest „Stop Mode”, gdyż zatrzymuje on licznik TAR. W trybie „Continous Mode”, po przepełnieniu licznika TAR ustawiana jest flaga TAIFG. Z kolei w trybach „Up Mode” i „Up/Down

linii danych D5...D7 wyświetlacza LCD. Dzięki temu „nie utracono” czterech pinów I/O mikrokontrolera. Nie utracono również możliwości korzystania z wyświetlacza LCD podczas debugowania programu. Umożliwia to opcja kompilatora IAR (Emulator-> Release JTAG on Go), która to (gdy aktywna) powoduje odłączenie linii JTAG i skonfigurowanie ich jako standardowych pinów I/O. Uruchamiając program należy korzystać z tej opcji przed każdym fragmentem kodu, który używa współdzielonych pinów. Sposób użycia tej opcji demonstruje umieszczony na płycie CD program LCDvsJTAG.

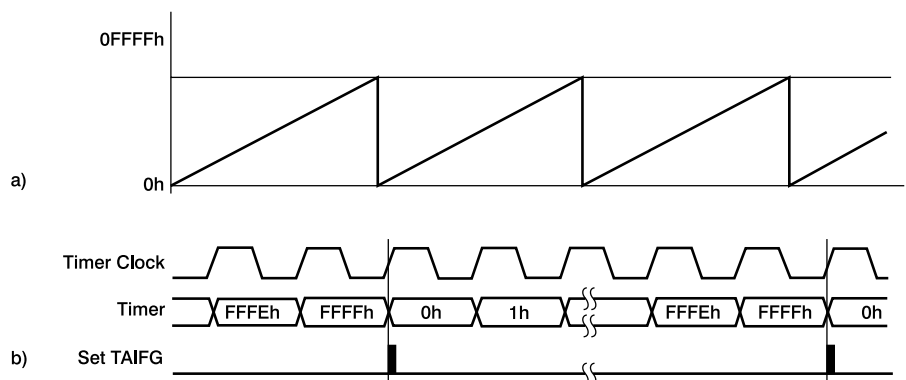
Moduł Timer_A – wstęp

Moduł Timer_A to 16-bitowy asynchroniczny licznik TAR współpracujący z trzema przechwytyjącymi/porównującymi rejestrami -TACCR0, TACCR1, TACCR2 (rys. 1).

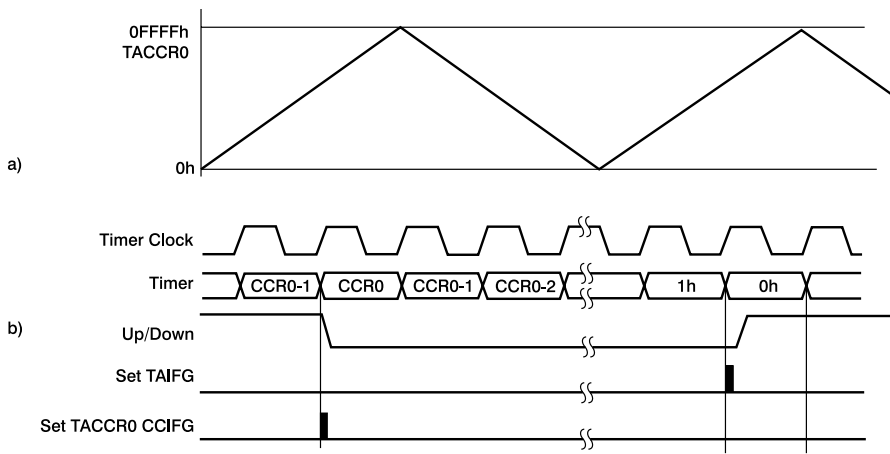
Sygnał taktujący licznik TAR wybierany jest programowo (TASSELx) spośród jednego z czterech dostępnych źródeł. Są to: zewnętrzne INCLK, TACLK (P2.1, P1.0 mikrokontrolera MSP430f1232) oraz wewnętrzne ACLK, SMCLK. Częstotliwość zegarowa może być wewnętrznie dzielona przez



Rys. 3.



Rys. 3.



Rys. 4.

Mode” licznik TAR współpracuje z rejestrem TACCR0. Również i w tych trybach przepełnienie licznika jest powodem ustawienia flagi TAIFG, a dodatkowo przy zmianie wartości TAR z TACCR0-1 na TACCR0 w rejestrze TACTL0 ustawiana jest flaga CCIFG.

Licznik może być wykorzystywany do generowania zdarzeń czasowych. Gdy programista włączy przerwanie od flagi TAIFG (TAIE rejestru TACTL) oraz CCIFG (CCIE rejestru TACCTLx), to w momencie ich ustawienia (odmierzenie zdefiniowanego odcinka czasu) program przejdzie do procedury obsługi przerwania.

Na płycie CD, umieszczono program – RTC demonstrujący wykorzystanie trybu pracy „Up

Mode”, do implementacji zegara czasu rzeczywistego. Program korzysta z bibliotek `Lcd2x16.lib` (obsługa wyświetlacza LCD) i `Convert.lib` (konwersja liczb). Biblioteki przystosowano do obsługi płytki eMeSPek. Program RTC, to korzysta on z trybu pracy energooszczędnej LMP3. Natomiast efekty działania programu, są prezentowane na wyświetlaczu LCD – raz na sekundę jest odświeżany upływający czas.

Wektory przerwania, modułu Timer_A

Z modułem Timer_A, powiązane są dwa wektory procedur obsługi przerwania (list. 1). Są to wektory TACCR0 i TAIV. Ten pierwszy współ-

pracuje z rejestrem TACCR0. Drugi związany jest z pozostałymi rejestrami oraz przepełnieniem licznika TAR. Wyższy priorytet obsługi posiada przerwanie od TACCR0. Przejście do procedury obsługi przerwania odbywa się na skutek ustawienia flagi CCIFG rejestru TACCR0. W chwili wejścia do procedury obsługi przerwania automatycznie zerowana jest flaga CCIFG rejestru. W przypadku TAIV wejście do procedury obsługi przerwania odbywa się na skutek ustawienia flagi CCIFG któregoś z rejestrów TACCR1, TACCR2, bądź flagi TAIFG informującej o przepełnieniu się licznika TAR. Informacja o tym, które z flag są aktualnie ustawione jest przechowywana w rejestrze TAIV. Jakkolwiek dostęp do rejestru (odczyt, bądź zapis), powoduje wyczyszczenie flagi przerwania o najwyższym priorytecie. Najwyższy priorytet wśród flag wektora TAIV posiada CCIFG rejestru TACCR1, następnie CCIFG rejestru TACCR2, oraz TAIFG. Jeśli w rejestrze TAIV będą ustawione dwie flagi np.: CCIFG od rejestru TACCR1, oraz CCIFG od rejestru TACCR2, to zgodnie z tym, co napisałem program wejdzie do procedury obsługi przerwania, dostęp do rejestru TAIV spowoduje wyczyszczenie flagi CCIFG od TACCR1, po czym program opuści procedurę obsługi przerwania. Ponieważ nadal w rejestrze TAIV pozostaje ustawiona flaga CCIFG rejestru TACCR2, to program ponownie wejdzie do procedury obsługi przerwania TAIV i rozpocznie jej wykonywanie.

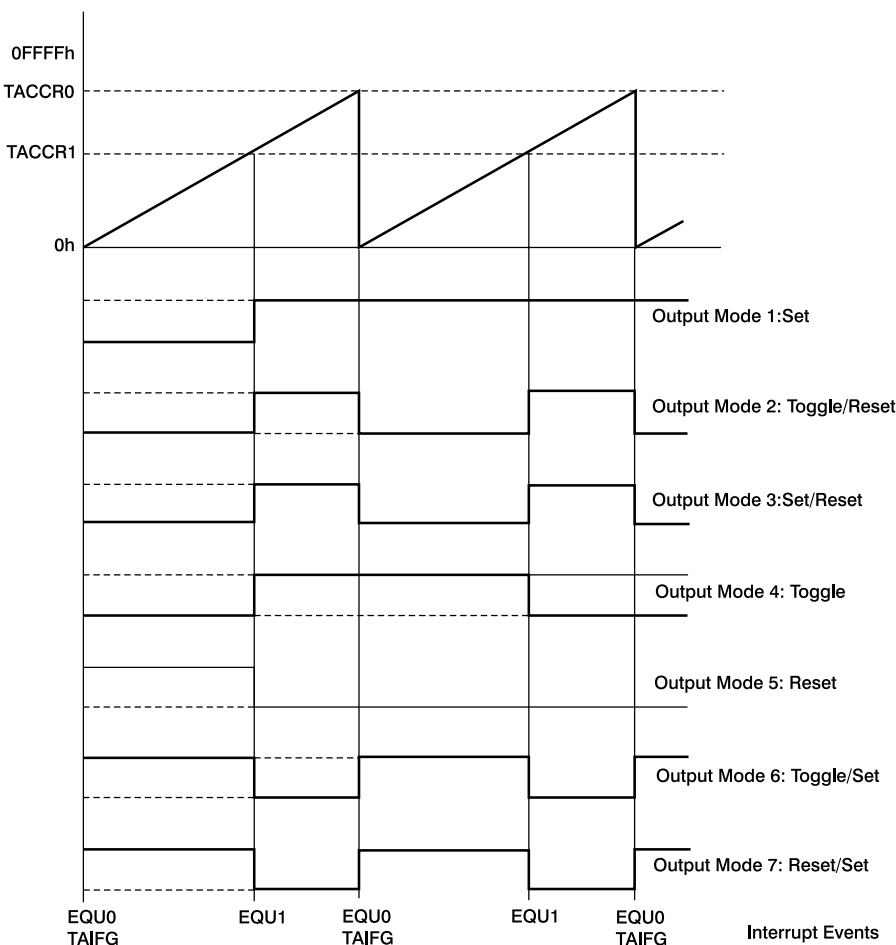
Rejestry TACCRx

Moduł Timer_A posiada trzy rejestry TACCRx, a z każdym z rejestrów jest powiązany konfiguracyjny rejestr TACCTLx. Rejestry mogą pracować w jednym z dwóch trybów pracy. Dostępne tryby pracy to porównujący oraz przechwytyjący (compare/capture). Konfiguracji trybu pracy dokonujemy modyfikując w rejestrze konfiguracyjnym TACCTLx wartość tagu CAP. Jeśli bit CAP posiada wartość zero to rejestr TACCRx pracuje w trybie porównującym (compare), w przeciwnym wypadku rejestr jest skonfigurowany jako rejestr przechwytyjący (capture).

Tryb porównujący – compare

W tym trybie wartość licznika TAR porównywana jest z wartością wpisaną do rejestru TACCRx. W momencie, gdy obie wartości są równe, to ustawiana jest flaga CCIFG rejestru TACCRx, sygnał wejściowy CCI jest zatrzymywany w SCCI (rejestr TACCTLx), a w zależności od konfiguracji OUTMODx (również rejestr TACCTLx), zmienia się stan wyjścia OUTx.

Fakt, że ustawiana jest flaga CCIFG sprawia, iż tryb porównujący może być wykorzystywany do generowania zdarzeń czasowych. Wystarczy, że programista aktywuje przerwanie dla rejestru TACCRx (CCIE rejestru konfiguracyjnego – TACCTLx), a w procedurze obsługi przerwania będzie mógł reagować na odmierzenie zdefiniowanego odcinka czasu. Oczywiście w mikrokontrolerze musi być aktywna obsługa przerwania maskowalnych (ustawiony bit GIE



Rys. 5.

w rejestrze SR). Podczas generowania zdarzeń czasowych nie należy zapominać o trybach pracy licznika TAR. Mianowicie w trybach „Up Mode”, oraz „Up/Down Mode” używany jest rejestr TACCR0. Skutkuje to tym, że nie może on pełnić funkcji rejestru porównującego. Tryb pracy licznika „Continuous Mode” nie posiada takiego ograniczenia. Wszystkie rejestry TACCRx mogą być wykorzystywane, co daje możliwość generowania trzech niezależnych zdarzeń czasowych.

Z każdym z rejestrów TACCRx powiązany jest sygnał wyjściowy OUTx. Sygnał ów jest doprowadzony do wybranych pinów mikrokontrolera, oraz wewnętrznego przetwornika analogowo cyfrowego. Wyboru wyjścia sygnału OUTx możemy dokonać modyfikując wartość tagu CCISx z rejestru TACCLx (tab. 2). Natomiast modyfikując wartość tagu MODOUTx możemy zdefiniować sposób pracy sygnału OUTx. Dostępnych jest osiem trybów pracy sygnału OUTx (tab. 3). Pokróćce pisząc, w trybach pierwszym oraz piątym zatraskiwane jest wyjście OUTx. Tryb czwarty na wyjściu OUTx generuje sygnał, który ma częstotliwość równą połowie częstotliwości sygnału taktującego licznik TAR. Natomiast tryby: drugi, trzeci, szósty oraz siódmy wykorzystane są do generowania przebiegów o regulowanym wypełnieniu(PWM). Na rys. 5 przedstawiono jak tryby pracy wpływają na zmiany sygnału OUTx, w przypadku pracy licznika TAR w trybie „Up Mode”.

Na płycie CD umieszczono program PWM, przy pomocy którego generowane są dwa niezależne sygnały PWM. Każdy z sygnałów posiada częstotliwość 1 Hz. Różne natomiast jest ich wypełnienie. Wyjścia sygnałów PWM zostały doprowadzone do pinów mikrokontrolera odpowiedzialnych za sterowanie diodami. Dzięki temu obserwując zachowanie diod możliwa jest analiza efektów działania programu. Piny sterujące diodami D1, D2 płytki eMeSPek są skonfigurowane jako piny peryferyjne. Nie wspominałem o tym wcześniej, ale aby wyjście OUTx było fizycznie dostępne na powiązany z nim pinie mikrokontrolera, to konieczna jest jego konfiguracja jako pinu peryferyjnego.

Tryb przechwytyjący – capture

W tym trybie pracy przechwytywany jest sygnał skonfigurowany jako wejście rejestru TACCRx. Wybór wejścia odbywa się przy pomocy bitu CCISx z rejestru konfiguracyjnego TACCLx. Dostępne są cztery wejścia sygnału, jednak w trybie przechwytywania przeważnie wykorzystywane są dwa: CCIxA, CCIxB. Do każdego z wejść CCIxA, CCIxB przypisane jest źródło sygnału. Powiązania wejście <-> źródło są różne dla różnych typów mikrokontrolerów MSP430 (tab. 4).

Modyfikując tag CMx, możemy wyłączyć tryb przechwytywania. Dodatkowo zmiana wartości CMx pozwala zdefiniować zbocze sygnału wejściowego, po wystąpieniu którego wyzwala się przechwytywana będzie wartość licznika TAR.

Tab. 3. Tryby pracy sygnału OUTx

OUT-MODx	Tryb pracy.	Skrócony opis.
000	Wyjście (Output)	Wartość wyjścia OUTx jest zmieniana w wyniku zmiany wartości bitu OUTx w rejestrze TACCRx.
001	Ustaw (Set)	Wyjście OUTx jest ustawiane, gdy TAR = TACCRx. Pozostaje ustawione do momentu wyczyszczenia licznika TAR.
010	Zmień/Wyczyść (Toggle/Reset)	Wartość wyjścia OUTx jest zmieniana, gdy TAR = TACCRx, a czyszczona, gdy TAR = TACCR0.
011	Ustaw/Wyczyść (Set/Reset)	Wyjście OUTx jest ustawiane, gdy TAR = TACCRx, a czyszczone, gdy TAR = TACCR0.
100	Zmień (Toggle)	Wartość wyjścia OUTx jest zmieniana, gdy TAR = TACCRx.
101	Wyczyść (Reset)	Wyjście OUTx jest czyszczone, gdy TAR = TACCRx.
110	Zmień/Ustaw (Toggle/Set)	Wartość OUTx jest zmieniana, gdy TAR = TACCRx, a ustawiana, gdy TAR = TACCR0.
111	Wyczyść/Ustaw (Reset/Set)	Wyjście OUTx jest czyszczone, gdy TAR = TACCRx, a ustawiane, gdy TAR = TACCR0.

Możliwe nastawy to: zbocze rosnące, opadające, oraz rosnące i opadające. W momencie, gdy sygnał wejściowy spełni warunki zdefiniowane przy pomocy bitów CMx (zbocze sygnału), to w rejestrze TACTLx zostanie ustawiona flaga CCIFG, a do rejestru TACCRx zostanie wpisana wartość licznika TAR. Ustawienie flagi CCIFG umożliwia programiście reakcję na zdarzenie w procedurze obsługi przerwania. Tryb ten można wykorzystać we własnych aplikacjach na wiele sposobów, ale jego podstawowe przeznaczenie to pomiar czasu (z dokładnością do cyklu zegarowego sygnału sterującego pracą TAR) wpływającego pomiędzy zboczami sygnałów.

Jeśli kolejne wyzwolenie nastąpi zanim programista odczyta wartość rejestru TACCRx, to zostanie ustawiony bit COV informujący o przepełnieniu trybu przechwytywania. Dodatkowo programista w dowolnym momencie może odczytywać poziom wejściowego sygnału (bit CCI), oraz wymusić synchronizację sygnału wejściowego z sygnałem taktującym licznik TAR (SCS=1). Bezpieczną praktyką jest synchronizowanie sygnału, gdyż zapobiega sytuacji, w której podczas odczytu/zapisu z/do rejestru TACCRx wystąpi przechwycenie i odczytywana/zapisywana wartość będzie miała nieprawidłową wartość.

Na płycie CD umieszczono program demonstrujący pomiar częstotliwości sygnału taktującego licznik TAR. Stało się tak, gdyż program ów w przejrzysty i efektowny sposób demonstruje działanie trybu przechwytywania. Oczywiście można zmodyfikować program w taki sposób, aby mierzył częstotliwości sygnałów doprowadzonych do pinów mikrokontrolera. Nie jest to trudne zadanie. Wystarczy zmienić źródło sygnału taktującego licznik TAR oraz przekonfigurować wejście przechwytywanego sygnału na sygnał doprowadzany do pinu mikrokontrolera a wybrany pin ustawić w tryb pracy pinu peryferyjnego.

W programie znajdującym się na płycie CD (CaptureMode) źródłem sygnału taktującego licznik TAR jest DCO, natomiast przechwytywanym sygnałem jest wewnętrzny zegar ACLK (LFX1CLK w trybie LF). Program, przez więk-

szość czasu przebywa w trybie energooszczędnym LPM0, a raz na sekundę wykonuje pomiar częstotliwości sygnału DCO, którego wynik wyświetla na LCD.

Pomiar częstotliwości DCO umożliwia przeprowadzenie interesującego i kształcącego doświadczenia. Mianowicie w poprzednim artykule wspomniano, że częstotliwość DCO jest zależna od temperatury otoczenia. Dzisiaj, posługując się programem CaptureMode można potwierdzić prawdziwość tej tezy. Wystarczy zmienić temperaturę otoczenia płytki eMeSPek (np.: podgrzać płytkę suszarką do włosów) i obserwować na wyświetlaczu LCD jak zmienia się częstotliwość DCO.

Układ Watchdog – wprowadzenie

Watchdog to swego rodzaju strażnik ciągłości pracy mikrokontrolera. Ma za zadanie wykrywać nieprawidłowości (zbyt długie przebywanie mikrokontrolera w określonym punkcie programu, np.: w skutek jego zawieszenia się) i przeciwdziałać im (wykonać restart mikrokontrolera). Zasada działania układu jest prosta: zlicza on impulsy sygnału zegarowego do momentu, gdy osiągnie zdefiniowaną przez programistę wartość. Wówczas to układ Watchdog „wystawia” sygnał restartujący mikrokontroler. W gestii obowiązków programisty leży okresowa aktualizacja licznika układu Watchdog. Taki schemat postępowania, sprawia, że w prawidłowo działającym programie układ Watchdog nigdy nie powinien zliczyć impulsów powodujących restart mikrokontrolera. Niestety, jak pokazuje praktyka, zdarzają się przypadki, że mikrokontroler „zawisa” i taki restart ma miejsce. Można by wówczas rzec, że urządzenie nie działa prawidłowo, gdyż miał miejsce niekontrolowany restart mikrokontrolera. W zasadzie tak jest. Jednak patrząc na problem z drugiej strony, to ów restart sprawia, że mikrokontroler powraca do normalnej pracy oraz urządzenie zaczyna funkcjonować. Jest to szczególnie istotne w urządzeniach, których nieprzerwana praca jest bardzo ważna, a nieprawidłowe funkcjonowanie mikrokontrolera może być wręcz niebezpieczne. Nikomu nie trzeba uświadamiać jak opłakane

w skutkach mogłoby być zawieszenie się urządzenia pomiarowego i utrata danych, bądź co gorsza urządzenia, które jest czujnikiem dymu, wycieku gazu, lub stanowi część systemu antywłamaniowego. Mogłoby dojść do tragedii. Mając na uwadze fakt, że mikrokontrolery MSP430 są głównie stosowane w urządzeniach, o których wcześniej wspominałem, postanowiłem, że obsługa układu Watchdog zostanie poświęcona spora część dzisiejszego artykułu.

Układ Watchdog – konfiguracja

Mikrokontrolery MSP430 serii 1xx posiadają wbudowany układ Watchdog (rys. 6). Konfiguracja układu odbywa się przy pomocy rejestru WDTCTL. Rejestr konfiguracyjny chroniony hasłem jest. W przypadku dostępu do rejestru bez hasła (odczyt 0x6900, zapis 0x5A00 - bit WDTPW), bądź z błędnym hasłem następuje restart (PUC) mikrokontrolera.

Układ Watchdog, poza trybem pracy, w którym restartuje mikrokontroler, może pracować jako licznik czasu (WDTTMSSEL). Jeśli programista włączy przerwanie od układu Watchdog (WDTIE rejestru IE1) i upłynie zdefiniowany przez programistę czas pracy układu, to w rejestrze IFG1 zostanie ustawiona flaga WDTIFG a program przejdzie do procedury obsługi przerwania. W procedurze obsługi przerwania (list. 1) pro-

gramista będzie mógł zareagować na zaistniałe zdarzenie czasowe. Osobiście odradzam stosowanie układu Watchdog w trybie licznika. Nie powinno się rezygnować z funkcjonalności zapewniającej ciągłość pracy mikrokontrolera (tryb restartu), kosztem dodatkowego licznika w systemie (tryb licznika). Stosowanie układu Watchdog w trybie licznika ma rację bytu jedynie w sytuacji, gdy w systemie jest wykorzystany zewnętrzny układ Watchdog podłączony do nóżki RST/NMI mikrokontrolera.

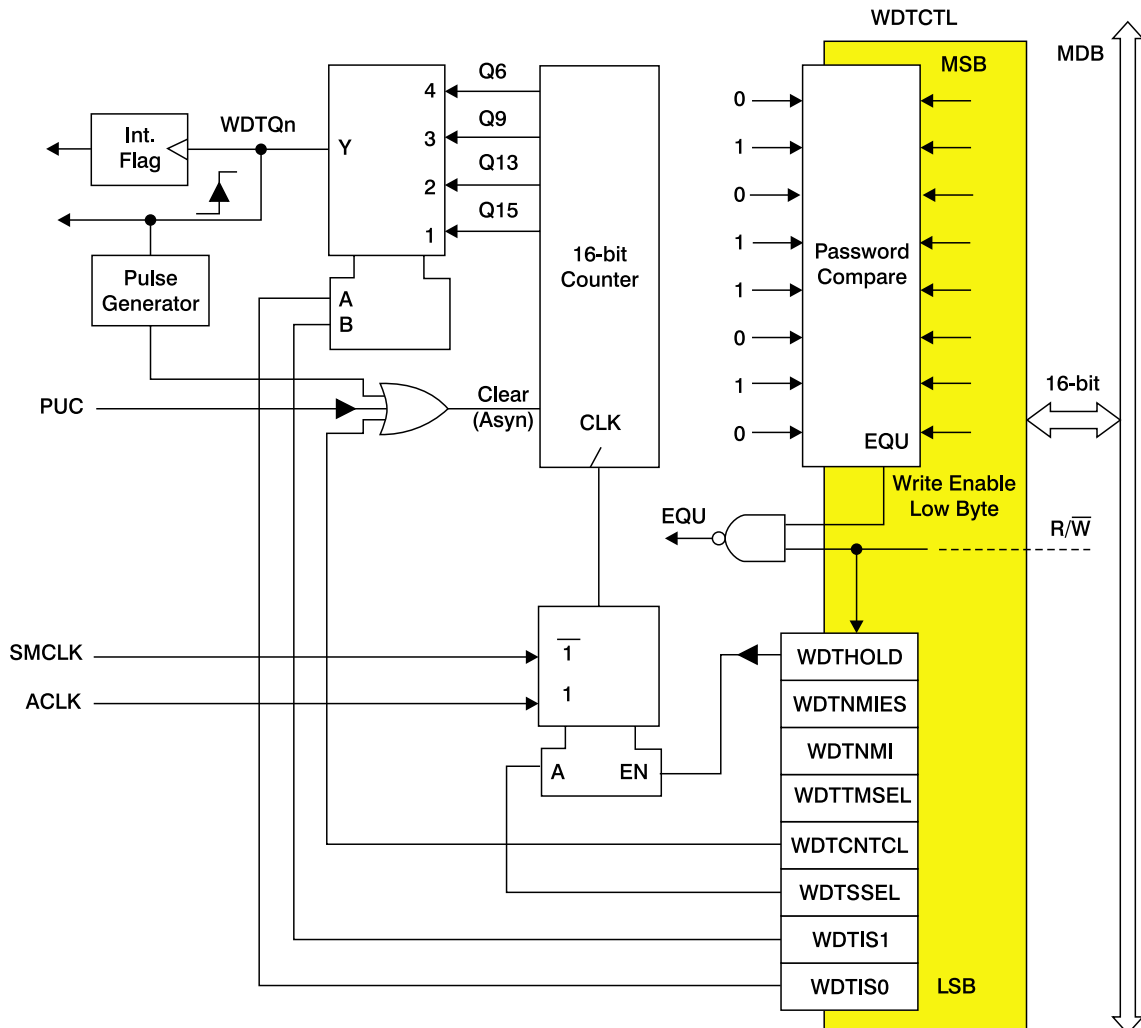
Układ Watchdog może być taktowany jednym z dwóch wewnętrznych sygnałów zegarowych, ACLK, bądź SMCLK (tag TASSELx). Czas pracy układu Watchdog; czyli czas po którym następuje restart mikrokontrolera, bądź w trybie pracy licznika ustawienie flagi WDTIFG i przejście do procedury obsługi przerwania; obliczany jest zgodnie z zależnością: $WDTt = DIVx / SRCf$ (s). We wzorze tym $DIVx$ to dzielnik sygnału taktującego układ Watchdog, natomiast $SRCf$ to jego częstotliwość podawana w Hertzach. Dzielnik częstotliwości sygnału jest konfigurowana przy pomocy bitu WDTISx. Dostępne wartości to 32768, 8192, 512, oraz 8.

Po starcie mikrokontrolera układ Watchdog jest włączony i pracuje w trybie restartu. Źródłem sygnału taktującego układ jest $SMCLK = DCO$. Częstotliwość pracy DCO wynosi około

Tab. 4. Rejestry TACCRx, tryb capture konfiguracja sygnałów dla MSP430f1232 (obudowa DW, oraz PW).

Rejestr TACCRx	Tag CCISx, konfigurujący wejście	Źródło sygnału Nóżka/Pin
TACCR0	CCI0A	22/P1.1
	CCI0B	10/P2.2
TACCR1	CCI1A	23/P1.2
	CCI1B	19/P2.3
TACCR2	CCI2A	24/P1.3
	CCI2B	Wewnętrzny ACLK

720 kHz, natomiast tag WDTISx konfiguruje dzielnik na wartość 32768. Jeśli programista nie zatrzyma układu Watchdog – WDTHOLD, nie przekonfiguruje trybu pracy na licznikowy, albo nie rozpocznie poganiania układu (WDTCNTCL), to zgodnie ze wzorem na czas pracy układu ($WDTt = 32768 / 720\text{kHz}$) po około 45 milisekundach układ Watchdog wystawi sygnał PUC i nastąpi restart mikrokontrolera. Sygnał PUC (Power-Up Clear), w odróżnieniu od sygnału POR (Power-On Reset) nie czyści pamięci RAM mikrokontrolera. Dlatego też, po restarcie PUC, obiekty umieszczone w pamięci RAM pozostają niezmiennione.



Rys. 6.

Przed startem programu kompilator języka C, inicjuje RAMowe obiekty globalne (wartość zero). Programista ma jednak możliwość wyłączenia inicjacji obiektów (dyrektywa `__no_init` poprzedzająca deklarację obiektu). Wówczas, może wykozystać fakt, że po restarcie na skutek zadziałania Watchdoga PUC nie czyści pamięci RAM i pracować dalej tak, jakby nic się nie stało. Przykład programu stosującego wyżej opisany sposób pracy z układem Watchdog umieszczono na płycie CD – RTCWDT. Program ów jest zmodyfikowaną o obsługę Watchdoga wersją programu RTC. Aktualny czas wyświetlany jest na LCD, a zawieszenie się programu jest symulowane w wyniku naciśnięcia dowolnego z przycisków S1, S2.

Programista monitorując stan flagi WDTIFG z rejestru IFG1 ma możliwość sprawdzenia, czy

program uruchamia się w wyniku restartu spowodowanego przez układ Watchdog. Jeśli flaga ma wartość jeden to tak jest. Wówczas programista powinien, zareagować na zaistniałą sytuację, oraz wyczyścić flagę WDTIFG, gdyż nie jest ona czyszczona automatycznie.

Wspomniałem, że po starcie mikrokontrolera układ Watchdog jest włączony i pracuje w trybie restartu. Może zajść sytuacja, że czas startu mikrokontrolera (inicjalizacja zmiennych, itp.) będzie dłuższy niż czas pracy układu Watchdog (45 ms). Programista w kodzie programu (main) nie zdąży odświeżyć stanu licznika Watchdog i mikrokontroler będzie się stale restartował. Rozwiązaniem problemu jest zmodyfikowanie pliku startowego mikrokontrolera, w taki sposób, aby zatrzymać układ Watchdog.

Na koniec artykułu chciałbym pokazać, w jaki sposób można to zrobić

Należy zmodyfikować plik startowy mikrokontrolera – `cstartup.s43` (Embedded Workbench 5.0 `\src\lib\430\`). Przed wykonaniem zmian w pliku proponuję zrobienie kopii zapasowej (np. do katalogu projektu.). Następnie należy plik dodać do projektu (Projekt -> Add Files...). Odszukać sekcję poświęconą konfiguracji układu Watchdog, oraz wyłączyć układ (komentarz w linii „`#if 0`” oraz „`#endif`”). Układ Watchdog jest wyłączany na początku sekcji inicjującej. Mamy zatem pewność, że nie minie czas pracy układu i mikrokontroler nie będzie cyklicznie restartowany.

Łukasz Krysiwicz.

R E K L A M A M A

ZAJRZYJ NA TE STRONY **ZAJRZYJ NA TE STRONY**

e-sklep **SIGMA**
www.wiazki.pl
 Wiązki kablowe Transformatory
 Elementy indukcyjne

www.dexon.pl
 TECHNIKA NAGŁOŚNIOWA

MER SERWIS aparatura kontrolno pomiarowa,
 elementy automatyki, serwis
 ul. Gen. Wł. Andersa 10
 00-201 Warszawa
 fax/tel: +48 22 831 42 56 **www.merserwis.pl**

aparatura pomiarowa, technika lutownicza
www.biall.com.pl
 końcówki kablowe, narzędzia, oscyloskopy **BIALL**

sklep. **INDUCTORS**.pl
 info@feryster.pl ELEMENTY INDUKCYJNE

Cyfronika **okinoity**
 www.cyfronika.com.pl
 elektronika dla wszystkich
 sklep internetowy
 wszystko dla elektroniki
 www.cyfronika.com.pl

GAMMA
www.gamma.pl
 info@gamma.pl **PODZESPOŁY ELEKTRONICZNE**

• NARZĘDZIA • LUTOWNICE • SPRZĘT POMIAROWY •
www.sklep.avt.com.pl
 • KLEJE • KITY VELLEMAN • i wiele innych...

www.alarmy-gerard.pl
SKLEP INTERNETOWY: www.gerard.pl

TONSIL sklep internetowy
 zestawy hi-fi głośniki **www.e-tonsil.pl**

WIĘCEJ NIŻ PROFESJONALNA DYSTRYBUCJA
M ARTHE **www.marthel.pl**
 UKŁADY SCALONE WINBOND, WARYSTORY
 TERMISTORY, KOMPUTERY PRZEMYSŁOWE

REZYSTORY PRECYZYJNE
 SMD - montaż powierzchniowy
 RWE - montaż przewlekany
 polski producent **www.elpod.com.pl**

MS Elektronik
 Dystrybutor Elementów Elektronicznych
 Tel. (58) 629 24 69
 Faks: (58) 629 32 00
 E-mail: info@mselektronik.com.pl
www.mselektronik.com.pl
 Oferta czynnych i biernych elementów elektronicznych renomowanych producentów

www.maszczyk.pl
ZTS MASZCZYK
 05-071 Sulejówek-Miłosna
 ul. Mickiewicza 10
 tel.: (0 22) 783 45 20
 fax: (0 22) 783 90 85
 maszczyk@maszczyk.pl