

# MSP430 w przykładach (6)

## Timer w trybie odmierzania czasu



**W artykule omówimy budowę i działanie modułu Timer w mikrokontrolerach MSP430. Pokażemy w jaki sposób odmierzać za jego pomocą czas, a korzystając z przygotowanego na potrzeby kursu układu „Komputerek” zrealizujemy projekty zegarka elektronicznego oraz pozytywki.**

Moduł Timer w MSP430 tworzą licznik oraz grupa rejestrów wspomagających pracę licznika (od 2 do 7 rejestrów). Dostępne są dwa rodzaje układów Timer: moduły typu A, oraz B. Zainstalowany w module „Komputerek” mikrokontroler MSP430f1232 ma jeden moduł Timer typu A. Zaawansowane układy z rodziny MSP430 mogą mieć nawet kilka bloków funkcjonalnych Timer. Przykładowo MSP430FR5739 z serii FRAM ma 5 modułów Timer (2 typu A, 3 typu B).

### Moduł Timer\_A

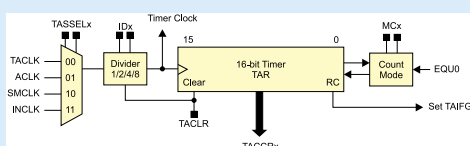
Wbudowany w MSP430f1232 moduł Timer\_A to 16-bitowy, asynchroniczny licznik TAR, który współpracuje z trzema rejestrami TACCR0, TACCR1, TACCR2. Licznik TAR jest konfigurowany za pomocą rejestru TACTL. Pracę rejestrów TACCR0, TACCR1, TACCR2 ustalają rejestry TACCTL0, TACCTL1, TACCTL2.

### Licznik TAR

Schemat blokowy licznika TAR pokazano na **rysunku 1**. Licznik może być taktowany jednym z sygnałów ACLK, SMCLK (wewnętrzne sygnały zegarowe) INCLK, TACLK (sygnały zewnętrzne podawane na nóżkę mikrokontrolera). Źródło taktowania licznika konfiguruje bit TASSELx. Częstotliwość sygnału taktującego licznik można podzielić przez 1, 2, 4 lub 8. Preskaler częstotliwości ustawia bit IDx.

Wartość licznika TAR zmienia się z każdym następującym zboczem sygnału taktującego licznik. W zależności od trybu pracy licznika, jest zwiększana, albo zmniejszana. Ponieważ licznik jest układem 16-bitowym, to minimalna wartość licznika wynosi 0, a maksymalna 65535. Zatem licznik może zliczyć 65536 impulsów, a po osiągnięciu wartości maksymalnej, kolejną wartością licznika jest 0. Zawsze, gdy wartość licznika zmienia się z wartości maksymalnej na 0, to w rejestrze konfiguracyjnym TACTL jest ustawiana flaga przerwania TAIFG (od przepełnienia licznika TAR). Flaga jest obsługiwana w przerwaniu modułu Timer\_A (wektor TAIV). Źródło przerwania konfiguruje bit TAIE.

Wyzerować licznik możemy wpisując do jego rejestrów wartość 0. Ten sam efekt uzyskamy ustawiając bit konfiguracyjny TACLRL. Jednak korzystając



Rysunek 1 Schemat blokowy licznika TAR

### Dodatkowe informacje:

Kody źródłowe programów „Zegarek elektroniczny” oraz „Pozytywka” zamieszczono na serwerze FTP. W programie „Zegarek elektroniczny” nie ma funkcji ustawiania czasu zegara. W dodatku do programu załączono program „Ustaw czas”, w którym wykonano funkcję ustawiania zegara. Można to zrobić używając zamontowanych w module „Komputerek” przycisków SW1 i SW2. Dodatkowo, w materiałach na serwerze FTP publikujemy filmy prezentujące działanie przykładów opisywanych w artykułach oraz dokument opisujący rejestry modułu Timer\_A w mikrokontrolerze MSP430f1232.

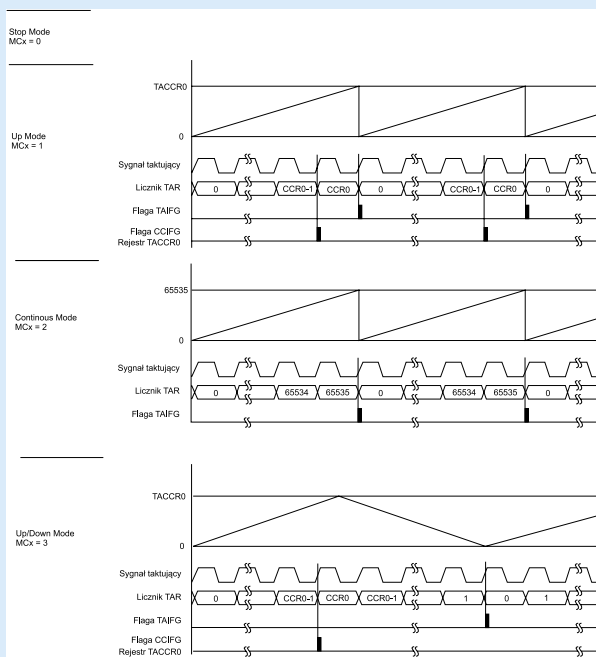
<ftp://ep.com.pl>, user: 32858, pass: 4285avne

„metody bitowej” należy pamiętać, że jego ustawienie nie tylko zeruje licznik TAR, ale także jest zerowany bit IDx (preskaler częstotliwości sygnału taktującego licznik).

Licznik TAR może pracować w jednym z 4 trybów pracy. Tryb pracy licznika konfiguruje bit MCx. Działanie licznika w zależności od trybu pracy pokazano na **rysunku 2**.

W trybie *Stop Mode*, licznik jest wyłączony. Tryb ten jest używany do zatrzymania pracy licznika TAR.

W trybie *Up Mode* (licz w górę), licznik cyklicznie zlicza od 0 do wartości wpisanej do rejestru TACCR0. Odliczane jest TACCR0+1 impulsów. W momencie, gdy wartość licznika TAR zmienia się z wartości TACCR0-1 na TACCR0, to w rejestrze TACCTL0 jest ustawiana flaga CCIFG. Flaga przepełnienia licznika TAIFG jest ustawiana przy zmianie wartości z TACCR0 na 0.



Rysunek 2. Licznik TAR. Tryby pracy. Flagi przerwania

W trybie *Continuous Mode* (praca ciągła), licznik cyklicznie zlicza od 0 do 65535. W momencie przepelnienia licznika ustawiana jest flaga TAIFG.

W trybie *Up/Down Mode* (licz góra/dół), licznik zlicza impulsy od 0 do wartości wpisanej do rejestru TACCR0. Następnie zmniejsza swoją wartość od TACCR0 do 0. Jest odliczane  $2 \times TACCR0$  impulsów. W momencie, gdy wartość licznika TAR zmienia się z wartości TACCR0-1 na TACCR0, to w rejestrze TACCTL0 jest ustawiana flaga CCIFG. Flaga przepelnienia licznika TAIFG jest ustawiana przy zmianie wartości licznika z 1 na 0.

### Rejestry TACCRx

Licznik TAR w MSP430f1232 współpracuje z 3 identycznymi rejestrami TACCRx. Schemat blokowy pojedynczego rejestru TACCRx ilustruje **rysunek 3**.

Rejestry TACCRx mogą pracować albo w trybie porównującym *Compare*, albo w trybie przechwytyjącym *Capture*. Tryb pracy rejestru konfiguruje bit CAP. Tryb porównujący *Compare* jest używany do generowania zdarzeń czasowych. Umożliwia także generowanie impulsów o regulowanym czasie trwania – modulację PWM. Tryb przechwytywania *Capture* jest stosowany do pomiaru czasu trwania impulsu oraz pomiaru czasu upływającego pomiędzy impulsami, ewentualnie do pomiaru częstotliwości sygnału.

W trybie porównującym *Compare* wartość licznika TAR jest porównywana z wartością wpisaną do rejestru TACCRx. W momencie, gdy obie wartości są równe, to automatycznie w rejestrze TACCTLx jest ustawiana flaga przerwania CCIFG oraz wewnętrzny sygnał EQUx. Fakt ustawienia flagi CCIFG można wykorzystać do generowania zdarzeń czasowych. Włączamy przerwania dla rejestru (bit konfiguracyjny CCIE) i w procedurze obsługi przerwania zareagować na odmierzenie zdefiniowanego odcinka czasu.

### Przerwania

Moduł Timer\_A, obsługiwany jest przez dwa wektory przerwań. Są to wektor TACCR0, oraz TAIV. Wektor TACCR0, obsługuje przerwania od rejestru TACCR0. Wektor TAIV obsługuje przerwania od pozostałych rejestrów, oraz przerwanie od przepelnienia licznika TAR. Wyższy priorytet obsługi ma przerwanie od wektora TACCR0.

Szablony procedur obsługi przerwań wektorów TACCR0 oraz TAIV pokazano na **listingu 6.1**. Do obsługi przerwań od wektora TAIV jest używany rejestr TAIV. Jest w nim przechowywana informacja o ustawieniu flagi przerwania, a odczyt rejestru powoduje automatyczne wyzerowanie jednej z nich. W przypadku, gdy w rejestrze jest ustawionych więcej niż jedna flaga, to podczas odczytu rejestru, najpierw zostaje wyzerowana flaga CCIFG rejestru TACCR2, a na koniec flaga TAIFG od przepelnienia licznika TAR.

### Przykłady

Zaprezentujemy dwa przykłady użycia modułu Timer do odmierzania czasu. W przykładzie „Zegarek elektroniczny” będziemy odmierzać jednosekundowe odcinki czasu, a upływający czas (sekundy, minuty, godziny) wyświetlimy na ekranie LCD. Do tego celu użyjemy między innymi licznika TAR pracującego w trybie *Up Mode*.

```

Listing 6.1 Procedury obsługi przerwania. Wektory TACCR0, TAIV
#pragma vector=TIMER_A0_VECTOR // wektor TACCR0
interrupt void Timer_A (void)
{
    // flaga: CCIFG,
    // źródło: rejestr TACCR0
}
#pragma vector=TIMER_A1_VECTOR // wektor TAIV
interrupt void Timer_A1 (void)
{
    switch( TAIV ) // odczyt rejestru TAIV
    {
        case 2: //flaga: CCIFG, //źródło: rejestr TACCR1
        {
            break;
        }
        case 4: //flaga: CCIFG, //źródło: rejestr TACCR2
        {
            break;
        }
        case 10: // flaga: TAIFG // źródło: przepelnienie // licznika TAR
        {
            break;
        }
    }
}
    
```

Wykorzystamy przepelnienie licznika i moment ustawienia flagi przerwania TAIFG (**rysunek 2**). Czas pracy licznika TAR w trybie *Up Mode* określa **wzór 6.1**. Wartość wpisywaną do rejestru TACCR0 obliczymy ze **wzoru 6.2**.

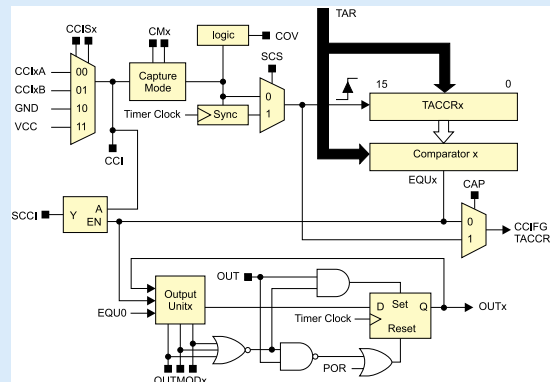
$$tPracy = (wRejestr0 + 1) / fZegara \quad (6.1)$$

$$wRejestr0 = tPracy \times fZegara - 1 \quad (6.2)$$

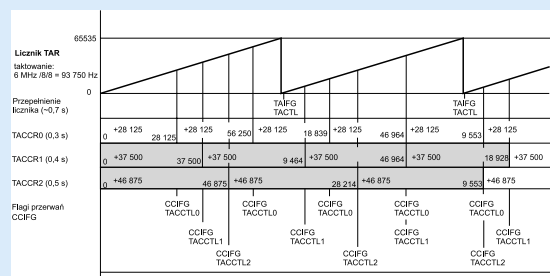
gdzie:

- tPracy: czas pracy licznika TAR w trybie *Up Mode* (s).
- fZegara: częstotliwość sygnału taktującego licznik TAR (Hz).
- wRejestr0: wartość rejestru TACCR0.

W przykładzie „Pozytywka” będziemy odtwarzać utwory muzyczne zapisane w formacie RTTTL (*Ring Tone Transfer Language*) zaprojektowanym przez firmę Nokia. Do odmierzania czasu użyjemy licznika TAR pracującego w trybie *Continuous* (praca ciągła) oraz rejestrów TACCR0, TACCR1, TACCR2 obsługiwanych w trybie *Compare* (porównaj).



**Rysunek 3.** Schemat blokowy rejestru TACCRx



**Rysunek 4.** Cykliczne odmierzanie trzech odcinków czasu (0,3; 0,4; 0,5 s). Przepelnienie licznika i ustawienie flagi TAIFG można użyć do pomiaru czwartego odcinka czasu (około 0,7 s)

**Listing 2. Procedura obsługi przerwania Timera 1**

```
#pragma vector=TIMER1_VECTOR
__interrupt void Timer_A1 (void)
{
    switch( TAIV )
    {
        case 2:
            break;
        case 4:
            break;
        case 10:
            ++sekunda;
            ZegarAktualizuj();
            _bic_SR_register_on_exit(CPUOFF + SCG0 + SCG1);
            break;
    }
}
```

Licznik TAR pracujący w trybie *Continuous* oraz rejestry TACCRx w trybie *Compare* najczęściej używane są do cyklicznego odmierzenia czasu. Wykorzystując wszystkie rejestry TACCRx możemy odmierzyć 3 odcinki czasu, a pomiar polega na cyklicznym dodawaniu (tzw. podbijaniu) wartości rejestru TACCRx (rysunek 4). Czas trwania odmierzanego odcinka czasu definiuje wzór 6.3. Wartość wpisywaną do rejestru TACCRx obliczymy ze wzoru 6.4.

$$oCzas = wRejestrx / fZegara \quad (6.3)$$

$$wRejestrx = oCzas \times fZegara \quad (6.4)$$

gdzie:

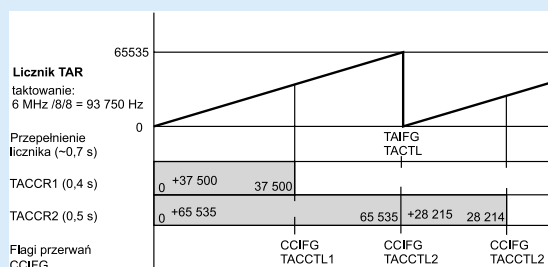
- oCzas: odmierzany odcinek czasu [s].
- Zegara: częstotliwość sygnału taktującego licznik TAR [Hz].
- wRejestrx: wartość rejestru TACCRx

Ponieważ rejestry TACCRx są 16-bitowe, to można do nich wpisać liczbę z zakresu 0...65535. Jeśli wartość obliczona ze wzoru 6.4 jest większa niż 65535, to odmierzanie „długiego” odcinka czasu należy podzielić na etapy i kilkakrotnie odmierzać „krótkie” odcinki czasu aż do momentu uzyskania wymaganego opóźnienia (rysunek 5).

### Zegarek elektroniczny

Program „Zegarek Elektroniczny” uruchamiamy korzystając z modułu „Komputerek”. Zworki JP7 i JP8 dołączające rezonator kwarcowy do źródła zegarowego LFXT1 należy ustawić w pozycji LF. Pozostałe zworki układu należy ustawić w pozycji IO/Off, a w złączu szpilkowym Dis1 zamontować wyświetlacz LCD.

Pliki źródłowe programu zostały zamieszczone na płycie CD oraz na serwerze FTP. Struktura programu jest identyczna, jak programu z ćwiczenia „Sekundnik” (EP01/13). Na jego początku dołączane są pliki nagłówkowe, deklarowane procedury oraz zmienne globalne. W programie głównym jest konfigurowany generator taktujący Basic Clock, moduł Watchdog Timer, porty we/wy, wyświetlacz LCD oraz jest włączana obsługa przerwania



**Rysunek 5. Pomiar dwóch odcinków czasu: 0,4 s oraz 1 s. Odmierzając odcinek 1 s, pomiar podzielono na dwa etapy (65535+28215=93750 taktów licznika TAR, czyli 0,7 s+0,3 s=1 s)**

maskowalnych. W pętli głównej programu, na ekranie LCD jest wyświetlany czas, a następnie mikrokontroler jest wprowadzany w tryb uśpienia LPM3.

Różnica pomiędzy programem „Sekundnik” a programem „Zegarek elektroniczny” jest taka, że w tym pierwszym czas był odmierzany przy użyciu modułu Watchdog (tryb pracy Timer, zegar), a z trybu uśpienia mikrokontroler budziły przerwania od przepełnienia licznika Watchdog. W drugim programie czas odmierza licznik TAR (tryb pracy *Up Mode*), a z uśpienia mikrokontroler budzą przerwania od przepełnienia licznika TAR.

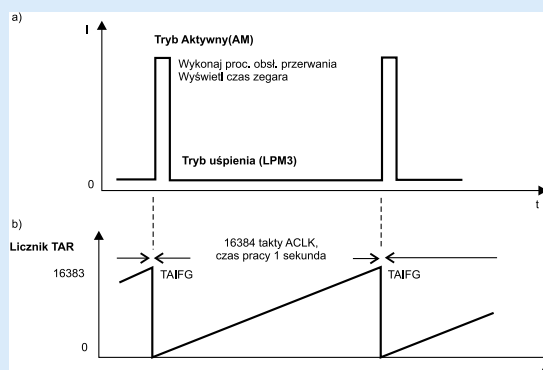
W programie licznik TAR jest taktowany sygnałem zegarowym ACLK (bit TASSEL\_1), jest włączana obsługa przerwania od przepełnienia licznika (bit TAIE) oraz tymczasowo zatrzymywana praca licznika (bit MC\_0). Odpowiada za to następujące podstawienie: TACTL=TASSEL\_1+MC\_0+TAIE;. W kolejnych instrukcjach licznik jest zerowany oraz jest zapisywany rejestr TACCR0:

```
TAR = 0;
TACCR0 = 16383;
```

Ponieważ licznik ma odmierzać jednosekundowe odcinki czasu, to zgodnie ze wzorem 6.1, do rejestru TACCR0 jest wpisywana wartość 16383 (częstotliwość sygnału ACLK wynosi 16384 Hz). Licznik jest uruchamiany po zakończeniu konfigurowania (włączenie trybu pracy *Up Mode*) za pomocą polecenia TACTL |= MC\_1;

Licznik zaczyna pracować i z każdym narastającym zboczem sygnału ACLK zwiększana jest jego wartość. W pętli głównej programu wywoływana jest procedura wypisująca na ekranie wyświetlacza aktualny czas zegara, a następnie mikrokontroler wprowadzany jest w tryb uśpienia LPM3. W stanie uśpienia sygnał zegarowy ACLK jest aktywny i taktuje licznik TAR. W momencie, gdy minie czas pracy licznika (sekunda) to licznik przepełni się (zmiana wartości TAR z TACCR0 na 0) i zostanie ustawiona flaga TAIFG. Ponieważ w programie włączona została obsługa przerwania od przepełnienia licznika, to ustawienie flagi TAIFG obudzi mikrokontroler z uśpienia i aktywuje wykonanie procedury obsługi przerwania, którą umieszczono na listingu 2.

W procedurze sprawdzane jest źródło przerwania i jeśli przerwanie wywołało ustawienie flagi od przepełnienia licznika TAR (wartość 10 w wektorze TAIV), to jest inkrementowany licznik sekund (zmienna sekunda), aktualizowany czas zegara oraz zgłaszany rozkaz opu-



**Rysunek 6. „Zegarek elektroniczny” pętla główna programu: a) praca mikrokontrolera w trybie aktywnym / trybie uśpienia, b) praca licznika TAR w trybie Up Mode**

**Listing 3. Pomiar czasu**

```
TACCRx = 0;
TACCTLx = (CCIE + CCIFG);
TACTL |= (MC_2 + TACLRL);
__bis_SR_register(CPUOFF + SCG0 + SCG1);
```

czenia trybu uśpienia po powrocie z procedury obsługi przerwania. Po wyjściu z procedury obsługi przerwania, ponownie jest wykonywana pętla główna programu. Na ekranie wyświetlacza jest wypisywany aktualny czas zegara, a mikrokontroler jest usypiany. Kolejne przerwanie od przepelnienia licznika TAR budzi mikrokontroler, a cykl pracy programu powtarza się. Pracę pętli głównej programu ilustruje **rysunek 6**.

**Pozytywka**

Program „Pozytywka” uruchamiamy korzystając z modułu „Komputerek”. Zworki JP7 i JP8 dołączające rezonator kwarcowy do źródła zegarowego LFXT1 należy ustawić w pozycji LF. Zworkę JP6 należy ustawić w pozycji Spk (dołączenie brzęczyka). Pozostałe zworki układu należy ustawić w pozycji IO/Off, a w złączu szpilkowym Dis1 zamontować wyświetlacz LCD.

Moduł „Komputerek” zaprogramowany aplikacją „Pozytywka” odtwarza 8 utworów muzycznych zapisanych w formacie RTTTL. Definicje utworów zamieszczono w pliku *Muzyka.h*, opis formatu RTTTL w pliku *Specyfikacja.txt*. Procedury pozwalające „czytać” utwory muzyczne zdefiniowano w pliku *RTTTL.c*.

W efekcie wykonania funkcji Parsuj, dla każdej nuty utworu jest obliczany okres drgania nuty oraz czas trwania. Dodatkowo, procedura sprawdza poprawność utworu. Jeśli analizowany utwór muzyczny nie zawiera błędów, to na ekranie LCD jest wyświetlany jego tytuł i rozpoczyna się odtwarzanie. W przeciwnym wypadku jest wyświetlany komunikat informujący o błędzie. Przykładowo, pierwsze takty 5 symfonii Bethovena zapisane w formacie RTTTL wyglądają następująco:

```
const char utwor[] = "5SymfoniaB:d=4,o=5,b=6
3:8p,8g5,8g5,8g5,2d#5"
```

Wywołanie procedury Parsuj należy wykonać w następujący sposób Parsuj(utwor).

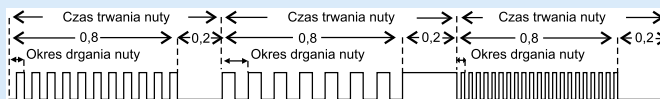
Do odmierzenia czasu trwania oraz okresu drgania nuty używane są rejestry TACCR0, TACCR1. Czas ciszy pomiędzy utworami (1 sekunda) jest odmierzany przy użyciu rejestru TACCR2. Wszystkie rejestry pracują w trybie *Compare*, a licznik TAR jest ustawiony w trybie pracy ciągłej *Continuous*. Nuty w utworze muzycznym grane są w technice „staccato”, co oznacza, że pomiędzy dźwiękami jest wprowadzana pauza (chwila ciszy). Technika odtwarzania nut ilustruje **rysunek 7**.

W sekcji konfiguracyjnej programu definiowane są parametry pracy licznika TAR. Licznik jest zatrzymywany (bit MC\_0) oraz jest dokonywany wybór źródła sygnału taktującego licznik (bit TASSEL\_1). Licznik jest taktowany sygnałem ACLK o częstotliwości 32768 Hz. Za tę funkcjonalność odpowiada następująca linijka programu: TACTL = TASSEL\_1 + MC\_0;

W programie głównym odmierzane są:

- czas trwania nuty (czas drgania nuty + pauza),
- okres drgania nuty,
- pauza pomiędzy utworami.

Do pomiaru czasu trwania nuty (drgania i pauzy) jest używany rejestr TACCR0 oraz procedura obsługi przerwania Timer\_A0 (wektor TACCR0). Obliczona ze



**Rysunek 7. Odtwarzanie nut w technice staccato. Czas drgania nuty to 80% czasu trwania nuty. Czas ciszy nuty to 20% czasu trwania nuty**

wzoru 6.4 wartość jest wpisywana do zmiennej czas0. Ponieważ z obliczeń można uzyskać wartość większą niż 65535 (nuty mogą być grane dłużej niż 2 sekundy), to w procedurze obsługi przerwania odmierzenie czasu zostało podzielone na etapy, którą to technikę opisano we wstępie jako metodę pomiaru „długiego” odcinka czasu. Do pomiaru okresu drgań nuty jest używany rejestr TACCR1 oraz procedura obsługi przerwania Timer\_A1 (wektor TAIV bit 2). Czas jest odmierzany w trybie pomiarów cyklicznych, a w procedurze obsługi przerwania stan linii sterującej brzęczykiem piezoelektrycznym jest zmieniany na przeciwny (pobudzony do drgań brzęczyk generuje dźwięk). Przed pomiarem jest odczytywany obliczony w procedurze Parsuj okres drgania nuty i jest ustawiana wartość zmiennej czas1. Pomiar rozpoczyna się, gdy odczytana wartość jest większa od 0 (wartość równa 0 oznacza, że aktualna „nuta” to pauza i nie trzeba pobudzać brzęczyka). Brzęczyk jest pobudzany przez czas drgania nuty (80% czasu trwania nuty), a następnie jest wyłączany (20% czasu trwania nuty). Do pomiaru czasu trwania pauzy pomiędzy utworami jest używany rejestr TACCR2 oraz procedura obsługi przerwania Timer\_A1 (wektor TAIV bit 4). Melodie rozdziela sekunda ciszy. Zgodnie ze wzorem 6.4 wartość, którą należy wpisać do rejestru TACCR2 wynosi 32768. W odróżnieniu od przypadku z odmierzaniem czasu trwania nuty pomiaru nie trzeba dzielić na etapy (pomiar „krótkiego” odcinka czasu).

Pomiar czasu (krótki, długi, cykliczny) zawsze inicjowany jest w następujący sposób:

1. Inicjowanie zmiennej czasx,
2. Zerowanie rejestru TACCRx,
3. Włączenie obsługi przerwania dla rejestru TACCRx (bit CCIE) oraz ustawienie flagi wystąpienia przerwania dla rejestru TACCRx (bit CCIFG),
4. Zerowanie licznika TAR (bit TACLRL) i włączenie trybu pracy ciągłej licznika (bit MC\_2),
5. Wprowadzenie mikrokontrolera w tryb uśpienia LPM3.

Cały proces sprowadza się do kilku linijek programu, które zamieszczono na **listingu 3**.

Ustawienie flagi wystąpienia przerwania i włączenie obsługi przerwania dla rejestru TACCRx (punkt 3) powoduje, że mikrokontroler automatycznie rozpoczyna wykonanie procedury obsługi przerwania (Timer\_A0, albo Timer\_A1 bit 2 lub 4). W procedurze wartość ze zmiennej czasx wpisywana jest do rejestru TACCRx. Rozpoczyna się pomiar czasu (krótki, długi, cykliczny), a mikrokontroler wprowadzany jest w tryb uśpienia LPM3. Po zakończeniu pomiaru mikrokontroler budzony jest z uśpienia LPM3. Zatrzymywana jest praca licznika TAR (bit MC\_2) oraz jest wyłączana obsługa przerwania dla rejestru TACCRx (bit CCIE):

```
TACTL &= ~MC_2;
TACCTLx &= ~CCIE;
```

**Łukasz Krysiwicz, EP**