

MSP430 w przykładach (3)

Projektowanie urządzeń „low power”

Projektowanie energooszczędnych urządzeń mikroprocesorowych to proces złożony. W wypadku urządzenia sterowanego mikrokontrolerem, istotne jest nie tylko odpowiednie (energooszczędne) zaprojektowanie obwodów, ale również właściwa implementacja oprogramowania sterującego jego pracą z uwzględnieniem energooszczędnych trybów pracy mikrokontrolera. W artykule skupimy się na oprogramowaniu, szczególnie nacisk kładąc na omówienie trybów pracy MSP430.

W energooszczędnych urządzeniach mikroprocesorowych program sterujący pracą mikrokontrolera powinien „pilnować”, aby mikrokontroler jak najwięcej czasu spędzał w trybie uśpienia. Obudzony przez przerwanie mikrokontroler powinien maksymalnie szybko wykonać powierzone mu zadania (obsługa urządzeń peryferyjnych: pomiary, transmisja danych itp.) i ponownie zostać uśpiony (**rysunek 1**).

By średni pobór prądu był jak najmniejszy, w trybie pracy aktywnej mikrokontroler powinien przebywać jak najkrócej. Zadaniem programisty jest takie napisanie programu sterującego pracą mikrokontrolera, aby w trybie aktywnym program wykonywał się maksymalnie krótko (w jak najmniejszej liczbie cykli maszynowych).

Jedną z wielu zasad, której należy przestrzegać, jest używanie dyrektywy `__inline` podczas korzystania z „krótkich” procedur (2-3 linie kodu). Użycie dyrektywy sprawia, że kod procedury wstawiany jest w miejscu jej wywołania. Zwiększa się wielkość kodu wynikowego programu, ale za to wykonuje się on szybciej i oszczędzamy czas potrzebny na wywołanie procedury. Pisząc oprogramowanie należy również unikać wykonywania skomplikowanych obliczeń. Mnożenie, dzielenie, pierwiastkowanie i inne operacje arytmetyczne są wykonywane przez mikrokontroler MSP430f1232 programowo, bez wsparcia sprzętowego i dlatego czas ich wykonania jest względnie długi. („większe” MSP430 z serii 1xx mają wbudowany sprzętowy układ mnożący MPY16). Pisząc program istotne jest również ograniczenie liczby operacji na zmiennych 32-bitowych (typ `long`, `double`) oraz 64-bitowych (`long long`, `long double`). Mikrokontrolery MSP430 mają rdzeń 16-bitowy i operacje na zmiennych 32/64

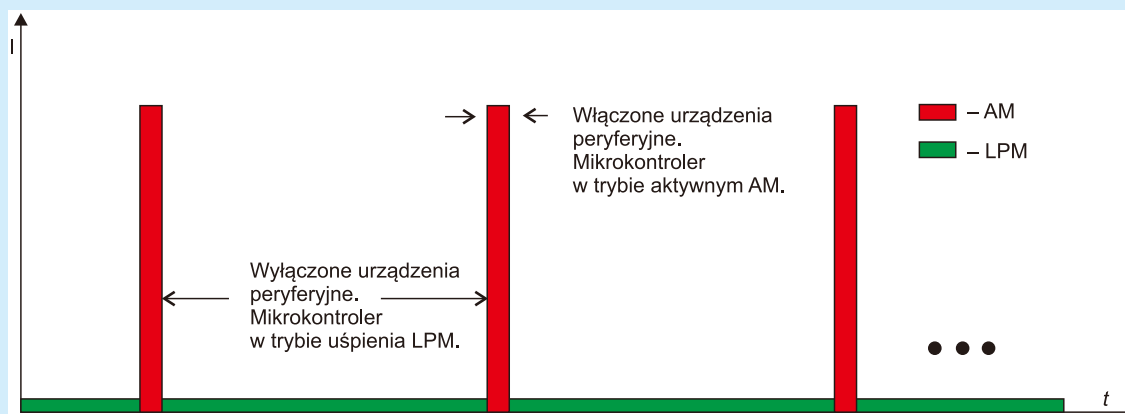
bitowych są wykonywane na „porcjach” zmiennych i znacząco wydłużają czas wykonania programu.

Aby ułatwić programistom optymalizację kodu programu, firma Texas Instruments przygotowała oprogramowanie *ULP Advisor*. Zdefiniowano w nim 15 reguł, których stosowanie powoduje zmniejszenie poboru prądu przez mikrokontroler. W momencie linkowania projektu (*Project -> Make*) oprogramowanie *ULP* analizuje składnię programu i doradza programiście, w jaki sposób można poprawić kod programu, aby zmniejszyć pobór energii. Informacje o proponowanych zmianach wyświetlane są w oknie komunikatów linkera.

W środowisku IAR, oprogramowanie *ULP* włączamy w opcjach projektu (*Project -> Options -> TI ULP Advisor -> Enable TI ULP Advisor*). Wyświetlanie komunikatów aktywujemy w narzędziach kompilatora (*Tools -> Options -> Show build messages -> All*). **Rysunek 2** ilustruje przykład działania programu *ULP*. W prezentowanym przykładzie, *ULP Advisor* doradza programiście, aby opóźnienia realizowane w sposób programowy (funkcja kompilatora `__delay_cycles`) wykonać korzystając z timera, przerwań i z użyciem trybów uśpienia mikrokontrolera. Dodatkowo, oprogramowanie sugeruje, aby instrukcje pętli `for(i=0;i<1000;i++)` zapisać w sposób `for(i=1000;i>0;i--)`. W wypadku proponowanej metody zapisu, pojedyncza iteracja pętli wykona się o jeden takt zegara szybciej, co dla całej pętli da zysk 1000 taktów zegara.

Tryby uśpienia

Zamontowany w module „Komputerek” MSP439f1232 poza trybem pracy normalnej AM (Active Mode), obsługuje 5 trybów oszczędzania energii LPM (*Low Power Mode*). W trybie pracy normalnej AM mikro-



Rysunek 1. Praca mikrokontrolera w urządzeniach energooszczędnych

kontroler pracuje z pełną mocą i ma włączone wszystkie sygnały zegarowe. W uśpieniu zawsze wyłączony jest sygnał zegarowy MCLK taktujący jednostkę centralną CPU mikrokontrolera. Dodatkowo, mogą zostać wyłączone sygnały zegarowe SMCLK, ACLK oraz generator taktujący DCO. W trybie LPM0 wyłączony jest sygnał zegarowy MCLK. W trybie LPM1 sygnał MCLK, oraz generator taktujący DCO. W trybie LPM2 sygnały zegarowe MCLK i SMCLK. W trybie LPM3 sygnały zegarowe MCLK, SMCLK oraz generator taktujący DCO (tylko sygnał zegarowy ACLK jest włączony). W trybie LPM4 wyłączony jest generator taktujący DCO, oraz wszystkie sygnały zegarowe mikrokontrolera, odświeżana jest jedynie zawartość pamięć RAM. Im więcej sygnałów zegarowych zostanie wyłączonych (wyższy jest stopień uśpienia), tym pobór prądu mikrokontrolera jest mniejszy. W tabeli 1 zamieszczono dane katalogowe poboru prądu przez mikrokontroler MSP430f1232 w zależności od trybu pracy.

Tryby uśpienia LPMx konfigurowane są za pomocą bitów rejestru SR (*Status Register*). Bity można modyfikować używając zadeklarowanych w pliku nagłówkowym `intrinsics.h` instrukcji: `__bis_SR_register()`, `__bic_SR_register()`. Instrukcja `__bis_SR_register()` ustawia bity rejestru SR (wyłącza sygnały zegarowe i generator DCO). Instrukcja `__bic_SR_register()` zeruje bity (włącza sygnały zegarowe i generator DCO). Na rysunku 3 pokazano efekty modyfikowania bitów.

Mikrokontroler może być wyprowadzony z trybu uśpienia tylko i wyłącznie przez przerwanie oraz za pomocą sygnału zerowania. W trybach LPM0...LPM3 może być to dowolne przerwanie mikrokontrolera. W trybie LPM4 tylko i wyłącznie przerwanie od portów wejścia-wyjścia lub sygnał zerowania. W momencie wystąpienia przerwania jest uruchamiany generator taktujący DCO (uruchomienie generatora trwa około 6 μ s). Działający generator rozpoczyna wytwarzanie sygnału DCOCLK=MCLK taktującego CPU mikrokontrolera (rdzeń CPU zaczyna pracować). Wówczas rejestr SR jest odkładany na stos, następnie jego wartość jest zerowana i jest wywoływana procedura obsługi przerwania. W procedurze obsługi przerwania mikrokontroler pracuje w trybie aktywnym AM (wyzerowane bity rejestru SR). Po zakończeniu procedury obsługi przerwania, zawartość rejestru SR jest zdejmowana ze stosu, a mikrokontroler powraca do trybu uśpienia. W wypadku, gdy nie chcemy, aby mikrokontroler po powrocie z procedury obsługi przerwania wracał do trybu uśpienia, to można skorzystać z zadeklarowanej w pliku `intrinsics.h` instrukcji: `__bic_SR_register_on_exit()`, która pozwala ona na wyzerowanie bitów odłożonego na stos rejestru SR. Użycie instrukcji spowoduje, że przy wyjściu z procedury obsługi przerwania ze stosu nie zostanie zdjęta odłożona wartość rejestru SR, ale wartość zmodyfikowana.

Napięcie zasilania

W trybie pracy normalnej pobór prądu MSP430 uzależniony jest od wartości napięcia zasilania mikrokontrolera. Dla zainstalowanego w module „Komputerek” MSP430f1232 zależność ta jest opisana za pomocą wzoru 3.1.

$$I_{AM} (MCLK = 1 \text{ MHz}) = I_{AM} [3 \text{ V}] - 120 \mu\text{A/V} \times (V_{cc} - 3 \text{ V})$$

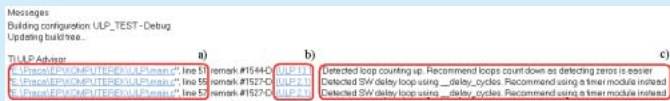
gdzie:

I_{AM} – aktualny pobór prądu w trybie pracy normalnej AM [μ A]

Tabela 1. Pobór prądu MSP430f1232. Dane katalogowe

Tryb pracy	Pobór prądu*
AM	300 μ A
LPM0	55 μ A
LPM1	wartość zależna od aktywności generatora DCO
LPM2	17 μ A
LPM3	1,6 μ A
LPM4	0,1 μ A

*T = 25°C, VCC = 3V, ACLK = 32768 Hz, MCLK=SMCLK = 1 MHz

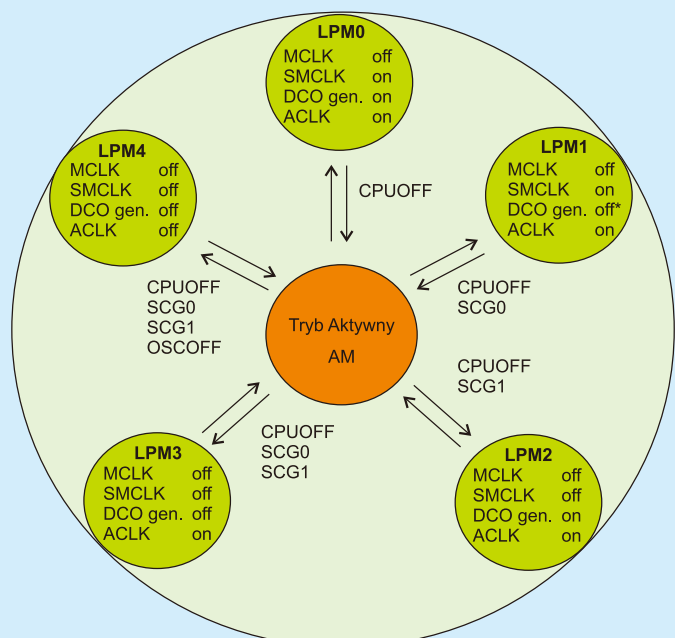


Rysunek 2. Przykład działania oprogramowania ULP Advisor a) link do kodu programu b) link do opisu działania reguły ULP c) skrócony opis reguły ULP

$I_{AM} [3 \text{ V}]$ – pobór prądu przy napięciu zasilania o wartości 3 V [μ A]

V_{cc} – aktualna wartość napięcia zasilania [V]

Przykładowo, zasilając MSP430f1232 napięciem o wartości 3 V uzyskujemy pobór prądu o wartości 300 μ A. Obniżając napięcie zasilania do 2 V, redukujemy pobór prądu do 180 μ A. Jak wynika ze wzoru (3.1), im jest mniejsza wartość napięcia zasilania, tym mniejszy pobór prądu. Budując energooszczędne urządzenie powinniśmy zatem mikrokontroler zasilać napięciem o jak najniższej wartości (zakres napięcia zasilania dla MSP430f1232 to 1,8...3,6 V). Trzeba jednak pamiętać, że obniżenie napięcia zasilania ogranicza maksymalną częstotliwość taktowania CPU (przy zasilaniu 1,8 V maksymalna częstotliwość taktowania MSP430f1232 to 4,15 MHz; przy zasilaniu 3,6 V to 8 MHz). Dodatkowo, w wypadku MSP430 z serii 1xx (a takim układem jest zastosowany w module „Komputerek” MSP430f1232) minimalna wartość napięcia, przy której pracuje kontroler pamięci Flash, to 2,7 V. Jeśli napięcie zasilania MSP430f1232 będzie niższe od 2,7 V, to mikrokontroler nie będzie mógł kasować oraz programować wewnętrznej pamięci Flash. W najnowszych MSP430 (seria 5xx/6xx/



Rysunek 3. Tryby pracy MSP430f122. Bity konfiguracyjne. (*) generator DCO można wyłączyć, gdy nie jest używany

„Ładunek elektryczny pobierany przez urządzenie/czas pracy na baterii – obliczenia”

Rozpatrzmy dwa przypadki pracy urządzenia. Pierwszy, gdy urządzenie pracuje w trybie pracy cyklicznej bez obsługi zdarzeń. Drugi, gdy urządzenie pracuje w trybie pracy cyklicznej z obsługą zdarzeń (rysunek 3). W pierwszym przypadku średni pobór ładunku elektrycznego obliczamy korzystając ze wzoru 1. (obliczenia należy wykonać dla jednego pełnego cyklu pracy urządzenia)

$$(1) \quad Q_{s1} = [(t_a \times I_a) + (t_u \times I_u)] / t_c,$$

gdzie:

ta, tu, tc [s] – czas aktywności/uśpienia/trwania cyklu pracy,

Ia, Iu [A] – pobór prądu w trybie aktywności/uśpienia.

W drugim wypadku obliczając średni pobór ładunku elektrycznego należy uwzględnić ładunek pobierany przez zdarzenie. Obliczenia trzeba wykonać w szerokim okresie analizy (godzina, dzień, tydzień). Trzeba oszacować ile razy (w rozpatrywanym okresie czasu) wystąpiło zdarzenie. Następnie pobór prądu możemy obliczyć korzystając ze wzoru 2.

$$Q_{s2} = [(n \times t_a \times I_a) + (n \times t_u \times I_u) + (m \times t_z \times I_z) + (m \times t_z \times I_z)] / t_o,$$

gdzie:

ta, tu, tz [s] – czas aktywności/uśpienia/trwania zdarzenia,

Ia, Iu, Iz [A] – pobór prądu w trybie aktywności/uśpienia/obsługi zdarzenia,

to[s] – analizowany okres czasu,

n – liczba cykli pracy urządzenia w czasie to (n=to/tc),

m – liczba zdarzeń w czasie to

Żeby obliczyć czas pracy urządzenia na jednym komplecie baterii (*), korzystamy ze wzoru 3. Parametry wejściowe wzoru to: pojemność baterii (zazwyczaj podawana w mAh), oraz średni pobór ładunku elektrycznego urządzenia.

$$T[s] = C[As] / Qs[As]$$

* przybliżony czas pracy urządzenia na jednym komplecie baterii, w praktyce czas działania urządzenia zależy od kilku dodatkowych zmiennych (zjawisko samorozładowania baterii, spadek napięcia baterii pod koniec jej okresu żywotności, wpływ temperatury otoczenia na charakterystykę pracy baterii, etc.)

** konwersja jednostek pojemności baterii C[As] = C[mAh] / 1000 * 3600

CC430) „problem” kontrolera pamięci Flash został rozwiązany i jest możliwa praca od napięcia zasilającego 1,8 V. Dodatkowo, najnowsze układy mają rozbudowany moduł zasilania mikrokontrolera PMM (Power Management Module), który wyposażono w przetwornicę LDO. Za jego pomocą programista może regulować napięcie zasilania rdzenia CPU.

Tabela 3. Pobór prądu MSP430f1232. Dane pomiarowe dla modułu „Komputerek”

Tryb pracy	Pobór prądu (1)
AM	240 μA
LPM0	35 μA
LPM1	35 μA (2)
LPM2	15 μA
LPM3	2.3 μA
LPM4	0.4 μA

(1) T = 25°C, VCC = 3.26 V, ACLK = 32768 Hz, MCLK = SMCLK ≈ 740 kHz, zasilanie USB, zworki w pozycji IO/Off, brak LCD, (SW1, SW2, TX) - wejścia (2) włączony generator DCO

Częstotliwość taktowania

W trybie pracy normalnej, pobór prądu MSP430 jest uzależniony również od częstotliwości sygnału zegarowego MCLK taktującego jednostkę centralną CPU mikrokontrolera. Zależność tę opisano za pomocą wzoru 3.2.

$$IAM (V_{cc} = 3 V) = IAM [1 MHz] \times fMCLK$$

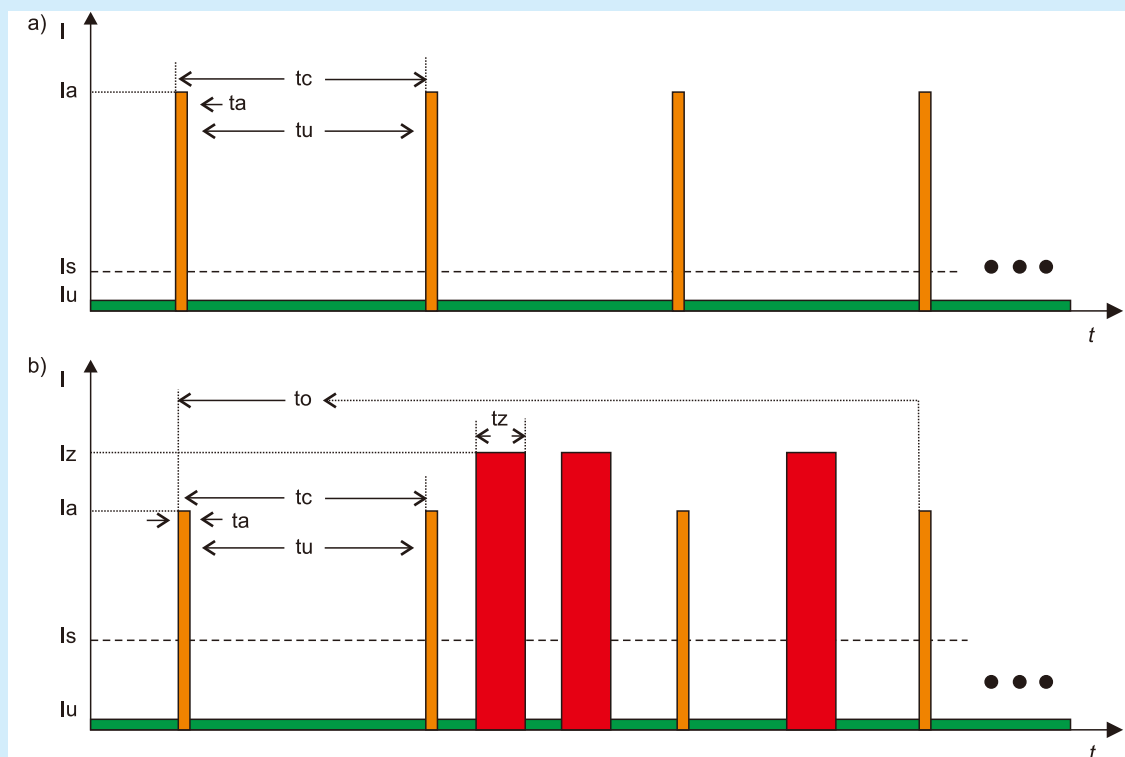
gdzie:

IAM – aktualny pobór prądu w trybie pracy normalnej AM [μA],

IAM [1 MHz] – pobór prądu przy częstotliwości MCLK równej 1 MHz [μA],

fMCLK – aktualna częstotliwość sygnału MCLK [MHz].

W wypadku MSP430f1232 i taktowania rdzenia CPU sygnałem zegarowym MCLK o częstotliwość 4 MHz pobór prądu wynosi 1,2 mA. Zmniejszenie częstotliwości sygnału zegarowego do 100 kHz ogranicza pobór prądu do 30 μA. Obniżając częstotliwość MCLK trzeba jednak pamiętać, że co prawda pobór prądu maleje, ale maleje także wydajność obliczeniowa mikrokontrolera i MSP430 pobiera mniej prądu, ale program wykonuje się dłużej. W praktyce częstotliwość taktowania CPU należy dobrać w sposób, który pozwoli na



Rysunek 4. Cykliczna praca urządzenia a) tryb bez zdarzeń, b) tryb z obsługą zdarzeń

efektywną obsługę urządzeń peryferyjnych (pomiar, transmisja danych, itd.). Peryferia powinny być obsługiwane w trybie przerwań. Procedury obsługi powinny być krótkie i wykonywane jak najszybciej. Pomiędzy przerwaniem mikrokontroler należy wprowadzać w tryb uśpienia LPMx.

Bilans energetyczny

Konstruując urządzenie o małym poborze energii powinniśmy stale monitorować pobór ładunku elektrycznego przez urządzenie. Aby obliczyć ile ładunku elektrycznego zużywa urządzenie, należy zmierzyć pobierany prąd oraz czas, w którym pobór prądu miał miejsce (wzory do obliczeń zamieszczono w ramce). W module „Komputerek” na linii zasilania zamontowano złącze typu CON. Do złącza można dołączyć amperomierz i zmierzyć pobór prądu. W tabeli 3 zamieszczono pomiary dla trybu pracy aktywnej oraz trybów uśpienia MSP430f1232.

W przypadku, gdy mierzony prąd jest sygnałem szybkozmiennym, to pomiar wartości prądu przy użyciu amperomierza może okazać się niemożliwy. Wówczas aby wykonać pomiar, trzeba użyć specjalistycznych urządzeń pomiarowych (szybki rejestrator, oscyloskop cyfrowy z pamięcią) albo zastosować metodę „chałupniczą” do pomiaru krótkich impulsów prądowych, polegającą na wydłużaniu czasu trwania mierzonego sygnału i wykonaniu pomiaru z użyciem amperomierza. Przykładowo, gdy chcemy zmierzyć prąd pobierany przez urządzenie podczas zapisu danych na kartę SD (szybki zapis 1 ms), to powinniśmy wydłużyć czas zapisu danych np.: do 1 sekundy i wówczas wykonać pomiar. Taka metoda pomiarów jest pracochłonna (trzeba wiele razy modyfikować kod programu), ale pozwala na uzyskanie pożądanego rezultatu. Czas trwania impulsów prądowych można zmierzyć przy użyciu oscyloskopu, bądź też licząc takty zegarowe programu. Obie metody wymagają sporego zaangażowania i są czasochłonne. Reasumując, bez używania specjalistycznych urządzeń pomiarowych, pomiar ładunku elektrycznego pobieranego przez urządzenie i obliczenie czasu pracy urządzenia na jednym komplecie baterii nie jest trywialnym zadaniem. Dlatego też inżynierowie z Texas Instruments rozpoczęli prace nad narzędziami *power debugging* dla MSP430 (planowana premiera to pierwszy kwartał 2013 roku). Prowadzone są prace nad oprogramowaniem (integracja ze środowiskiem IAR, oraz CCS) oraz sprzętem diagnostycznym (sonda do pomiaru prądu). Finalny produkt, wzorowany będzie na produkowanym przez firmę IAR *power debugging* dla ARM/Renesas RX. Identycznie jak w wypadku środowiska IAR, sonda pomiarowa będzie miała za zadanie wykonać pomiar prądu i przesłać wynik do oprogramowania. Oprogramowanie zestawi chwilowy pobór prądu z aktualnie wykonywanym kodem programu, a programista otrzyma informacje, które fragmenty kodu programu są bardziej, a które mniej energooszczędne.

W praktyce

Najczęściej używanym trybem uśpienia w MSP430 serii 1xx jest tryb LPM3. W tym trybie jest włączony tylko sygnał zegarowy ACLK pochodzący ze źródła LFXT1. Do budzenia MSP430 z trybu uśpienia LPM3 może być wykorzystany licznik TAR i przerwanie od przepełnienia licznika. Żeby w trybie LPM3 licznik działał poprawnie, trzeba go taktować sygnałem zegarowym ACLK. Konstruując urządzenie należy zamontować element ustalający częstotliwość sygnału ACLK (źródła LFXT1). Najczęściej na płytce drukowanej urządzenia montuje

List. 1 Przykład użycia trybu uśpienia LPM3

```
//zworki JP7, JP8 ustaw w pozycji LF, pozostałe zworki
w pozycji IO/Off
// (źródło LFXT1 <- kwarc zegarkowy)
//Zastosowanie trybu uśpienia LPM3.
#include "io430.h" // rejestry procesora
#include "intrinsics.h" // instrukcje procesora
int main( void ) // program główny
{
//konfigurowanie linii we-wy
P1DIR = 0xfc; P2DIR = 0xff; P3DIR = 0xef;
//konfigurowanie licznika TAR - ustaw czas pracy licznika
TACCR0 = „czas pracy licznika”;
//licznik TAR taktuj sygnałem zegarowym ACLK (źródło
LFXT1)
TACTL = TASSEL_1 + MC_1 + TAIE + TACLR;
//włącz obsługę przerwań
__bis_SR_register(GIE);
// pętla główna programu
while(1)
{
//włącz tryb uśpienia LPM3, tylko sygnał zegarowy
//ACLK jest aktywny - pozostałe są wyłączone )
__bis_SR_register(CPUOFF + SCG0 + SCG1);
//po obudzeniu mikrokontrolera
// * „pogoń” układ Watchdog
// * jak najszybciej wykonaj zadania
// * wykonaj sekwencję zadań (np.: pomiary, transmisje
danych).
//następnie ponownie wykonaj pętlę główną programu,
//oraz uśpij mikrokontroler
};
}

// procedura obsługi przerwania
#pragma vector=TIMER1_VECTOR
// rejestry TACCR1, TACCR2, przepełnienia licznika TAR
__interrupt void Timer_A1 (void)
{
switch( TAIIV )
{
case 2: // rejestr TACCR1 (brak obsługi)
break;
case 4: // rejestr TACCR2 (brak obsługi)
break;
case 10: // przepełnienie licznika TAR
// po zakończeniu procedury
__bic_SR_register_on_exit(CPUOFF + SCG0 + SCG1);
// obsługi przerwania
break;
} // wyjdź z trybu uśpienia LPM3
}
```

się kwarc zegarkowy o częstotliwości 32768 Hz. Cykl pracy urządzenia (interwał pobudek/przerwań) definiuje programista inicjując licznik.

Na listingu 1 pokazano szablon programu przygotowanego dla modułu „Komputerek”, w którym MSP430f1232 pracuje w trybie LPM3. W programie należy skonfigurować układ Watchdog oraz określić czas pracy licznika TAR (oba zagadnienia będą omawiane w kolejnych częściach kursu). Na początku programu konfigurowane są linie wejścia-wyjścia mikrokontrolera (żeby zmniejszyć pobór prądu wszystkie nieużywane linie powinny pracować w trybie wejścia – wyjścia (nie w funkcyjnym!) i być ustawione w kierunku wyjścia. Linie, do których dołączono urządzenia peryferyjne należy skonfigurować w sposób, który zmniejszy pobór prądu przez urządzenie. Ponieważ w module „Komputerek” linie P1.0 i P1.1 są na stałe dołączone do przycisków SW1, SW2, a linia P3.4 do układu nadawczego TX transmisji szeregowej UART, to aby zmniejszyć pobór prądu linie te zostały skonfigurowane jako wejścia. Pozostałe linie mikrokontrolera (zworki IO/Off są ustawione w pozycji Off) nie są dołączone do urządzeń peryferyjnych (nie są używane) i zostały skonfigurowane jako wyjścia.

W pętli głównej programu mikrokontroler cyklicznie wprowadzany jest w tryb uśpienia LPM3 (schemat pracy jak na rysunku 4). Obudzony przez przerwanie od licznika TAR (cykl pobudek/cykl pracy urządzenia określa czas pracy licznika TAR) wykonuje zdefiniowane przez programistę zadania i jest ponownie usypiany.

Łukasz Krysiewicz
lukasz_krysiewicz@interia.pl