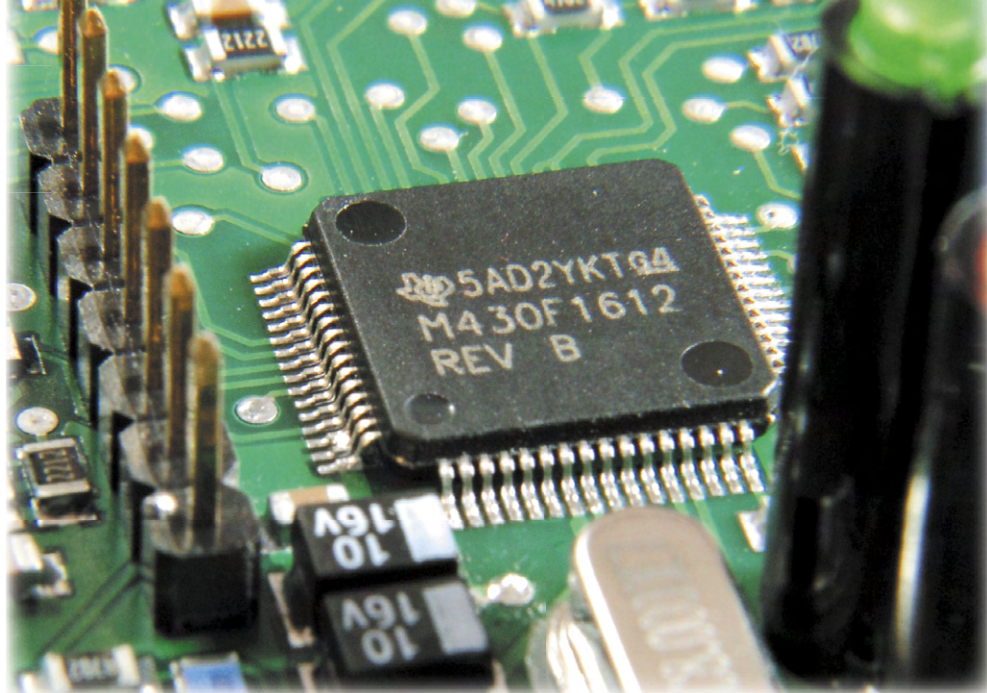




MSP430

μC dla oszczędzających energię

Zagadnienie energooszczędności urządzeń elektronicznych niezmiennie pozostaje w centrum zainteresowania. Nosimy przy sobie masę elektronicznych zabawek, których postępująca miniaturyzacja objęła również źródła energii. Wokół czyha sieć inteligentnych czujników, bezprzewodowych transponderów i podobnych gadżetów, których nowoczesność mierzy się miesiącami pracy bez wymiany baterii. Świat oszczędza energię.

Na pytanie, co sprawia, że niektóre procesory zużywają mniej prądu a inne więcej, niechybnie padnie odpowiedź: energooszczędność mikroprocesora wynika z technologii, w jakiej został wykonany – technologia jest receptą na energooszczędność.

Przykładowo mikroprocesor 80C286, wyprodukowany przez Intersil, będący oszczędną i szybką wersją popularnego PC-towego procesora, przy szybkości pracy 12 MHz pobiera ok. 1,3 W mocy; procesor CoreDuo o zegarze 1,8 GHz ok. 30 W; At89C2051 i Atiny2313 przy szybkości 12 MHz – ok. 17 mW, a MSP430F2131 przy szybkości 12 MHz – ok. 10 mW. Kto jest mistrzem oszczędności? Proste przeliczenie współczynnika poboru mocy do szybkości taktowania czyni zwy-

cięczą MSP430. Gdyby jednak sumiennie policzyć wydajność obliczeniową każdego z konkurentów, bezapelacyjnie wygrałby nowoczesny produkt Intel. Zawdzięcza to nowoczesnemu procesowi technologicznemu i obniżeniu napięcia zasilania. Jednak to MSP430 uważany jest w tym gronie za mikroprocesor energooszczędny – nie technologia jest najważniejsza. Co jeszcze o tym decyduje? Jakie parametry pozwalają na taką klasyfikację? I w końcu, jak zbudować urządzenie bateryjne?

Struktura CMOS, a w takiej technologii produkuje się dzisiaj mikrokontrolery, prawie nie pobiera prądu w stanie statycznym. Przyjmuje się też, że pobór prądu wzrasta w przybliżeniu liniowo wraz ze wzrostem częstotliwości przełączania. Wynika z tego, że wzrost szybkości obliczeniowej musi pociągnąć za sobą wzrost poboru prądu. Zatem redukcja szybkości pracy na pewno przyniesie oszczędność energii. I to jest recepta druga.

Wiele mikrokontrolerów można spowolnić do częstotliwości kilkudziesięciu kHz i uzyskać pobór prądu liczony w dziesiątkach μA. Nie jest to jednak praktyczne rozwiązanie – spowolniony system potrzebuje dużo czasu aby zareagować na bodźce, a wszystkie operacje ciągną się w nieskończoność. Może zamiast spowalniać system, lepiej będzie wyłączać go całkowicie? Pomysł świetny, przynosi ogromne oszczędności, ale taki system

w ogóle przestaje reagować na otoczenie. Recepta druga też nie jest optymalna.

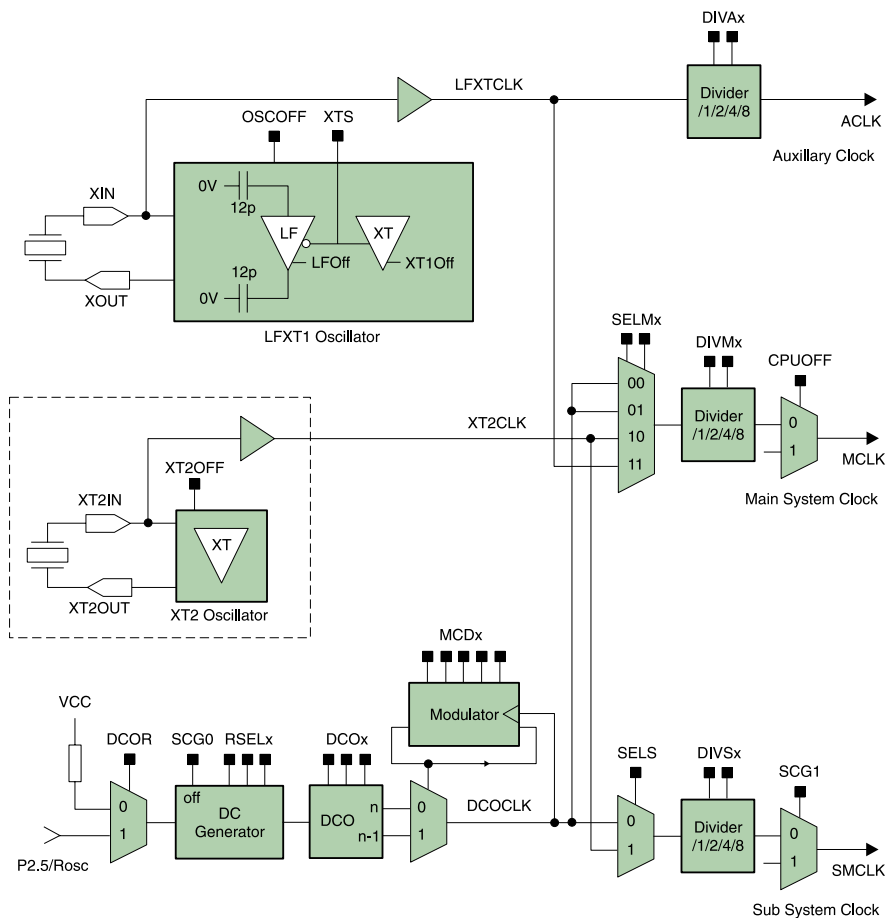
A może złoty środek? Owszem, wyłączyć należy wszystko, co nie jest niezbędne, a pozostawić w pełnej gotowości elementy odpowiedzialne za reakcję na bodźce. W razie konieczności cały system rusza do działania.

Okazuje się, że najmniej potrzebną mikroprocesorowi do pracy częścią jest... sam mikroprocesor, a właściwie jego jednostka centralna, która nie zawsze ma zadania do wykonania, ponadto jest głównym konsumentem energii. Recepta trzecia – wyłączmy jednostkę centralną. Architektura, która pozwala wyłączyć większość bloków i akceptuje beczynność jednostki centralnej, pozwala budować najbardziej energooszczędne aplikacje.

Wykorzystanie takiej architektury implikuje również sposób budowania aplikacji, której zadaniem, to nie żart, jest jak najdłuższe pozostawanie w beczynności.

MSP430 i energooszczędność

W procesorach MSP430 zawarto szereg elementów pozwalających na realizację powyższych recept. Jednostka centralna jest odrobinę bardziej oszczędna podczas pracy z pełną szybkością niż większość procesorów. Oszczędności nie są wielkie, ale można liczyć na zmniejszenie poboru energii o kilkanaście procent bez większego wysiłku. Układy pe-



Rys. 1. Schemat systemu zegarowego procesora MSP430F149

ryferyjne można elastycznie konfigurować, a także całkowicie wyłączyć. Kluczowe są jednak dwa mechanizmy – rozbudowany system taktowania i system nadzorujący bezpośrednio redukcję mocy.

System taktowania

System taktowania obejmuje trzy główne sygnały taktujące, które mogą pobierać sygnał zegarowy z kilku źródeł. Służą one taktowaniu jednostki centralnej i liczników. W procesorze istnieją jeszcze inne generatory taktujące, ale nimi nie będziemy się tutaj zajmować.

Na rys. 1 przedstawiono system zegarowy procesora MSP430F149. Sygnał taktujący można pobierać z trzech niezależnych źródeł. Podstawowe źródło, obecne w każdym modelu procesora MSP430, to generator DCO. W procesorach serii F1xx nie jest on stabilizowany i jego dokładność nie jest lepsza niż 10%. Ponadto DCO jest podatny na wpływ napięcia zasilania i temperatury. Częstotliwość wyjściową generatora DCO można jednak regulować w szerokim zakresie i z niewielkim skokiem. W procesorach F4xx generator DCO jest objęty pętlą stabi-

lizacji FLL, sterowaną sygnałem odniesienia z generatora kwarcowego LFXT1, co znacząco poprawia jego precyzję. Generator LFXT1 (XT1) jest obecny w każdym procesorze. Posiada wbudowane kondensatory i jest przystosowany do współpracy przede wszystkim z zegarkowym rezonatorem kwarcowym (32768 Hz). Kwarc zegarkowy ma tę zaletę, że do pracy wymaga wyjątkowo małej energii. Generator XT1 współpracuje też z rezonatorem dużej częstotliwości i rezonatorem ceramicznym, ale wtedy wymaga przełączenia w tryb HF o większym zużyciu energii. Co

istotne, w trybie HF praca generatora jest nadzorowana, i w razie błędów generowany jest sygnał *Oscillator Fault* (OFIFG). Trzecie źródło taktowania to generator XT2, który działa tylko w trybie HF, a jego praca jest również nadzorowana.

Przykładowy program nadzorujący uruchomienie generatora XT2 w procesorze serii F4xx przedstawiono na list. 1. Program włącza generator XT2 i po skasowaniu flagi błędu generatora i odczekaniu około 200 ms sprawdza, czy generator poprawnie się uruchomił. Jeśli wystąpił błąd, proces jest ponawiany. Dopiero po poprawnym uruchomieniu sygnał XT2 jest kierowany do taktowania systemu.

Wszystkie trzy sygnały mogą być generowane jednocześnie i elastycznie dołączane do szyn zegarowych w trakcie działania programu. Ciekawe, że awaria sygnału taktującego MCLK sterowanego kwarcem powoduje, że CPU jest przełączane na taktowanie z DCO w celu kontynuacji pracy (np. wykonania próby ponownego uruchomienia generatora).

Z dostępnych sygnałów zegarowych tworzone są trzy sygnały taktujące: ACLK, MCLK i SMCLK.

Podstawowym sygnałem taktującym jest sygnał zegarowy MCLK. Służy on taktowaniu CPU i może być wyłączany poleceniem programu. Jego źródłem może być każdy z dostępnych generatorów. Sygnał SMCLK służy taktowaniu peryferiów: interfejsów, przetworników A/C i liczników. Jego źródłem może być DCO lub XT2 (XT1, gdy nie ma XT2). Podobnie jak MCLK, może on być wyłączony przez program. Zwyczajowym źródłem taktowania timerów (szczególnie do odmierzania czasu) jest sygnał ACLK, który w większości aplikacji współpracu-

```
list. 1
FLL_CTL1 &= ~XT2OFF; // włączenie generatora
do // wykonaj w pętli
{
    FLL_CTL0 &= ~XT2OF; // skasuj flagę błędu XT2
    for (i = 50000; i; i--); // poczekaj na start generatora
}
while (FLL_CTL0 & XT2OF); // sprawdź poprawność startu
FLL_CTL1 = SELS+SELM1; // SMCLK, MCLK = HF xtal
```

Tab. 1. Ustawienia trybów pracy									
Tryb pracy	SCG0	SCG1	OSCOFF	CPUOFF	CPU	MCLK	SMCLK	ACLK	DCO
AM	0	0	0	0	T	T	T	T	T
LPM0	0	0	0	1	N	N	T	T	T
LPM1	0	1	0	1	N	N	N	T	warunkowo włączony
LPM2	1	0	0	1	N	N	N	T	częściowo wyłączony
LPM3	1	1	0	1	N	N	N	T	N
LPM4	1	1	1	1	N	N	N	N	N

je z generatorem LFXT1 i kwarcem zegarkowym.

Redukcja poboru mocy

W procesorach rodziny MSP430 wyróżnia się pięć trybów pracy o ograniczonym poborze energii. O trybie pracy, w jakim pozostaje procesor, decyduje zawartość rejestru R2 (tab. 1), pełniącego rolę rejestru statusu (SR).

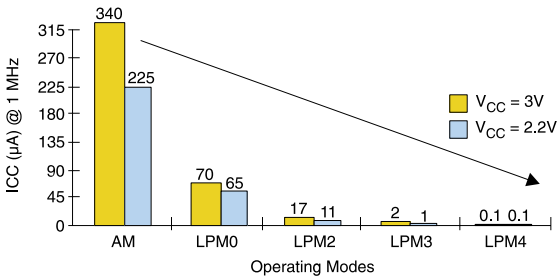
W trybie aktywnym AM (Active Mode), gdy procesor działa z pełną szybkością, pobór prądu wynosi ok. 280 µA przy 1 MHz (rys. 2). Największy spadek poboru mocy osiągamy poprzez wyłączenie CPU (tryb LPM0). Ponieważ jednak generator DCO jest w pełni aktywny, pobór prądu jest nadal spory. W trybach LPM1 i LMP2, DCO jest wyłączony lub częściowo wyłączony, co ogranicza pobór prądu do ok. 15 µA. Tryb LPM3 jest najszerszej używany w aplikacjach o bardzo małym poborze prądu. Do skutecznego użytkowania LPM3 konieczne jest działanie generatora XT1. Włączenie trybu LPM3 i taktowanie częstotliwością zegarkową daje świetne rezultaty. Najniższym trybem pracy jest LPM4, gdzie przy znikomym poborze prądu, podtrzymywany jest stan pamięci RAM, a procesor może reagować na zewnętrzne przerwania (np. od portów P1 i P2).

W praktyce programista nie operuje bezpośrednio bitami SCG0, SCG1,

OSCOFF i CPUOFF, lecz używa makr. Jest to praktyczne i bardziej bezpieczne. Ustawienie innej kombinacji bitów, niż zawarte w tab. 1, może spowodować nieprawidłowe działanie procesora. Modyfikacja stanu rejestru SR może być realizowana przez program lub automatycznie przez system obsługi przerwań w wyniku nadejścia przerwania, które to wymaga obsługi. W takim wypadku procesor zostaje wybudzony ze stanu o niskim poborze energii do trybu aktywnego, co najmniej na czas realizacji podprogramu obsługi przerwania. Jak wygląda taki mechanizm?

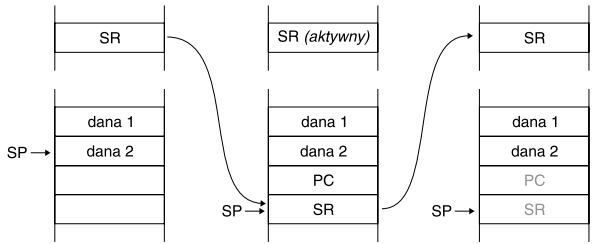
Realizacja podprogramu obsługi przerwania rozpoczyna się skopiowaniem na stos licznika programu PC oraz rejestru SR (rys. 3). Następnie bity odpowiedzialne za tryb pracy, zlokalizowane w SR, są zapisywane takimi wartościami, jak dla trybu aktywnego, w wyniku czego procesor uruchamia wszystkie wyłączone zegary i podejmuje pracę. Do licznika programu zostaje załadowany adres procedury pobrany z wektora przerwania i rozpoczyna się realizacja podprogramu obsługi przerwania. Po zakończeniu podprogramu odtwarzana jest poprzednia wartość licznika programu PC i zawartość rejestru RS, a procesor powraca do trybu pracy, w jakim pozostawał przed pojawieniem się przerwania.

Program z list. 2 działa dokładnie według tego schematu. Program jest przygotowany tak, aby do pracy nie wymagał rezonatora. Rozpoczyna się zatrzymaniem watchdog'a, który w MSP430 uruchamia się domyślnie po starcie programu. Następnie konfigurowany jest port P1



Rys. 2. Pobór prądu dla pięciu trybów pracy procesora MSP430

```
List. 2
void main()
{
    init_sys();
    low_power_mode_0(); // wprowadzenie LPM0
    __NOP();           // ta instrukcja nie zostanie wykonana
}
#pragma vector=TIMER0_VECTOR
interrupt void Timer_A (void)
{
    P1OUT ^= BIT0; // zmiana stanu, dioda LED
}
void init_sys()
{
    WDTCTL = WDTPW + WDTHOLD;
    P1DIR = 0xff;
    TACTL = TASSEL1 + TACLR; // SMCLK taktuje TimerA
    CCR0 = 50000;
    CCTLO = CCIE;
    TACTL |= MC1; // włączenie timera
    __enable_interrupt(); // główne przyzwolenie dla przerwań
}
```



Rys. 3. Algorytm podprogramu obsługi przerwania

(dołączona dioda LED) oraz licznik TimerA tak, aby był taktowany z zegara SMCLK, a pośrednio z DCO. Ostatnią wykonaną instrukcją jest makro __low_power_mode_0(), czyli instrukcja asemblera bis.w #CPUOFF,SR. Procesor zatrzymuje się, a następujące po niej makro __NOP() (No Operation) nigdy nie zostanie wykonane. Przepelnienie licznika generuje przerwanie o wektorze TimerA0_vector, w obsłudze którego występuje zmiana stanu sygnału sterującego świeceniem diody LED. W tym czasie procesor pobiera ok. 280 µA prądu, a podczas oczekiwania na nadejście przerwania ok. 65 µA.

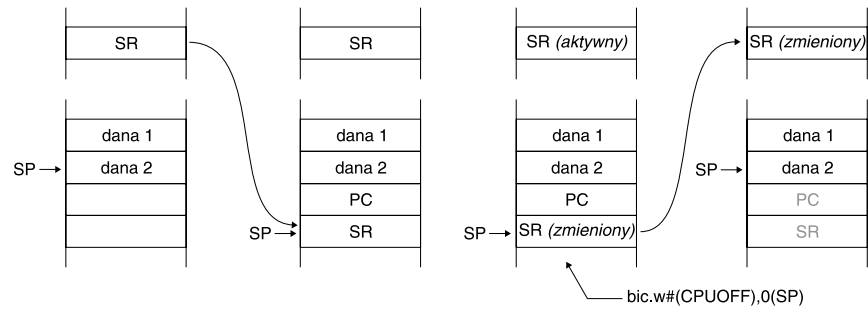
Gdy zachodzi potrzeba, aby w wyniku nadejścia przerwania, została wykonana też część programu zlokalizowana np. w pętli głównej, konieczne jest wykonanie dodatkowej operacji, aby procesor pozostał aktywny również po zakończeniu obsługi przerwania.

Operacja polega na zmodyfikowaniu zlokalizowanej na stosie kopii rejestru SR. Zmieniona wartość powinna odpowiadać wartości charakterystycznej dla trybu aktywnego. W takim wypadku, po zakończeniu podprogramu obsługi przerwania i odtworzeniu zawartości PC i SR ze stosu, procesor pozostaje w stanie aktywnym i może kontynuować wykonywanie programu (rys. 4).

Według tego właśnie schematu działa program z list. 3. Instrukcję zmiany stanu sygnału sterującego diodą LED przeniesiono do pętli głównej programu. Po niej wykonywana jest instrukcja wprowadzenia trybu LPM0. Wykonanie podprogramu obsługi przerwania wraz z makro __low_power_mode_off_on_exit(), czyli asemblerowe bic.w #(CPUOFF+OSCOFF+SCG0+SCG1),0(SP) pozwala na wykonanie programu z pętli głównej, aż do kolejnej instrukcji wprowadzenia trybu LPM0. Należy jeszcze wyjaśnić, że zawartość rejestru SR może znajdować się poniżej szczytu stosu. W asemblerze należy brać to pod uwagę i odpowiednio kontrolować.


```
List. 3
void main()
{
    init_sys();
    while (1)
    {
        P1OUT^=BIT0;          // zmiana stanu, dioda LED
        _low_power_mode_0();  // wprowadzenie LPM0
        _NOP();
    }
}

#pragma vector=TIMER0_VECTOR
__interrupt void Timer_A (void)
{
    __low_power_mode_off_on_exit(); // wyjście z LPMx
}
```



Rys. 4. Algorytm podprogramu obsługi przerwania z usunięciem trybu pracy LPM0

wać argument instrukcji. Na szczęście kompilator języka w C dba o to, aby makro `__low_power_mode_off_on_exit()` operowało we właściwym miejscu stosu.

W odróżnieniu od programu z list. 3, w programie przedstawionym na list. 4 programista zapomniał w podprogramie obsługi przerwania umieścić makro operujące na stosie,

dlatego kod z list. 4 nie zadziała poprawnie.

Zaprezentowane przykłady są najprostsze z możliwych i nie wykorzystują wszystkich możliwości ograniczania poboru mocy. Znacznie lepszym rozwiązaniem byłoby, po zastosowaniu rezonatora zegarkowego, taktowanie licznika TimerA z zegara ACLK i wprowadzenie trybu LPM3, w którym pobór prądu wynosi 1...2 μ A.

Energooszczędne aplikacje – najważniejsze zasady

Na zakończenie kilka zasad, których należy przestrzegać przy projektowaniu aplikacji o bardzo niskim poborze energii. Przede wszystkim, budowanie energooszczędnej aplikacji polega na odpowiednim stosowaniu trybów oszczędzania energii, z naciskiem na użycie trybu LPM3 i wykonywanie programu tylko w następstwie zdarzeń (rys. 5).

Czas, gdy procesor działa w trybie aktywnym, należy ograniczać, co oznacza potrzebę tworzenia programu tak, aby wykonywał się możliwie szybko. W tym celu warto pamiętać o:

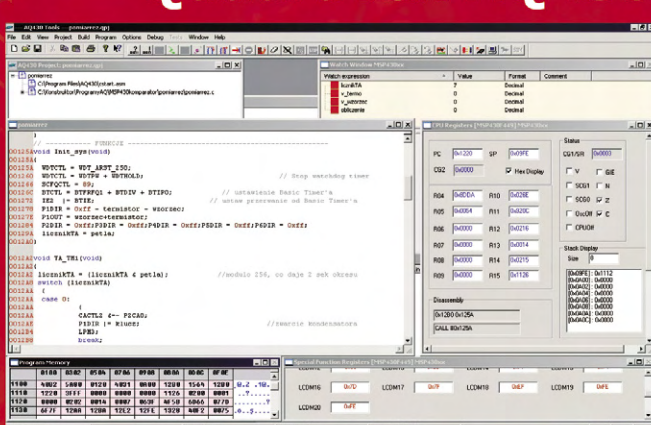
R E K L A M M A

TOOLS

Przypominamy o konkursie dla projektantów MSP 430. Szczegóły na www.contrans.pl

CONTRANS TI

Quadravox AQ430



Quadravox AQ430 – zintegrowane środowisko projektowe dla MSP430 kompilator C, debugging-in-system

W ofercie firmy Contrans również inne narzędzia do procesorów MSP430 i DSP oraz układów radiowych ISM firmy Texas Instruments a także pełna gama układów liniowych, logicznych i cyfrowych z oferty firmy TI.

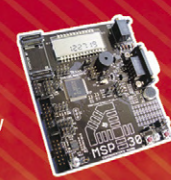
Narzędzia do mikrokontrolerów MSP430 i układów radiowych CCxxxx

MSP430FG4618/F2013 Experimenter Board – płyta ewaluacyjna do układów radiowych i MSP430

EZ430-RF2500 – zestaw ewaluacyjny do układów radiowych i MSP430

FlashPro430/FlashProCC – szybki programator JTAG/BSL i emulator do MSP430/programator do CCxxxx

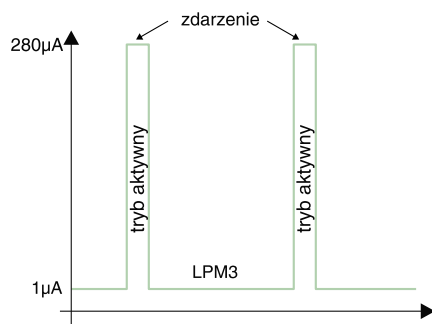
MSP-FET430UIF – emulator JTAG do MSP430 ze złączem USB



MSP 430

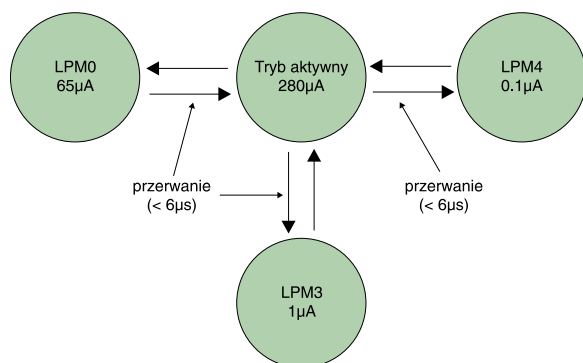
CONTRANS TI Sp. z o.o.
ul. Polanowska 66, 51-180 WROCLAW,
tel. 071/325-26-21...24, fax 071/325-44-39,
e-mail: texas@contrans.pl <http://www.contrans.pl>

TEXAS INSTRUMENTS



Rys. 5. Przebieg pracy procesora w aplikacji energooszczędnej

- budowaniu programu tak, aby wykonywał zadania tylko w następstwie zdarzeń (przerwań),
- ograniczaniu wywoływania podfunkcji (niepotrzebne dodatkowe skoki i operacje na stosie),
- stosowaniu tablic w miejsce obliczeń, jeśli tylko jest to możliwe,
- wykonywaniu skomplikowanych obliczeń na rejestrach (lub zmiennych lokalnych w języku C),
- wyłączaniu zbędnych peryferiów,
- używaniu peryferiów (liczników, bloków *capture/compare*) do wykonywania zadań kontrolnych, zamiast bezpośredniej kontroli programowej



Rys. 6. Diagram przejść stanu pracy procesora w aplikacji energooszczędnej

```

List. 4
void main()
{
    init_sys();
    while (1)
    {
        P1OUT^=BIT0;           // zmiana stanu, dioda LED
        low_power_mode_0();    // wprowadzenie LPM0
        _NOP();                 // ta instrukcja nie zostanie wykonana
    }
}
#pragma vector=TIMERAO_VECTOR
__interrupt void Timer_A (void)
{
}

```

- ustawieniu wszystkich nieużywanych linii I/O w tryb wyjścia,
- odpowiednim połączeniu wprowadzeń rezonatorów kwarcowych.

Ważnym elementem konceptu jest stosowanie generatora DCO do taktowania jednostki centralnej i peryferiów. Producent deklaruje, że generator ten uruchamia się i uzyskuje stabilny sygnał w czasie 6 μ s (rys. 6), a nawet 2 μ s w procesorach serii F2xxx.

Po czasie 6 μ s od pojawienia się przerwania, procesor taktowany z DCO rozpoczyna wychodzenie z trybu niskiego poboru energii (np. LPM3) i podejmuje pracę. Żaden generator kwarcowy nie uruchamia się tak szybko, a ponadto, czas jego uruchamiania może się wahać, powodując zmienny czas reakcji na zdarzenia.

Pierwsze energooszczędne aplikacje

Na zakończenie dwa przykłady energooszczędnych aplikacji. Świetnym przykładem zastosowania redukcji poboru mocy jest zegar czasu rzeczywistego, którego zadaniem jest odliczanie czasu z możliwie niskim poborem energii. W tej aplika-

cji najlepiej sprawdza się procesor serii F4xx, który posiada licznik *BasicTimer*, dedykowany do tego typu zastosowań. Zamiast *BasicTimer*, w procesorach serii F1xx można posłużyć się generatorem *TimerA*. Przy taktowaniu z zegara *ACLK* każdy z tych generatorów może generować przerwanie z częstotnością 1 Hz. W wyniku nadejścia przerwania, w kilku instrukcjach odświeżane są informacje o czasie i dacie, a przez resztę czasu procesor przebywa w trybie LPM3. Drugi ciekawy przykład to pilot zdalnego sterowania. Tutaj również ograniczenie poboru mocy jest bardzo istotne, szczególnie wtedy, gdy pilot nie jest używany. W takiej aplikacji można nawet pokusić się o zastosowanie trybu LPM4. Odpowiednie podłączenie klawiatury (nawet matrycowej) do portów P1 i P2, zapewni niezbędne do obudzenia procesora przerwanie zewnętrzne. Po podjęciu pracy procesor może zająć cię programową stabilizacją generatora DCO lub uruchomieniem generatora z rezonatorem kwarcowym (ceramicznym). Nie brakuje na to czasu, ponieważ bardzo szybka reakcja na wciśnięcie klawisza nie jest potrzebna. Po wykonaniu polecenia procesor powraca do trybu LPM4.

Mariusz Kaczor
Contrans TI

R E K L A M M A

ZAJRZYJ NA TE STRONY

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

info@eryster.pl
sklep.INDUCTORS.pl
ELEMENTY INDUKCYJNE

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

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