

# MSP430: mikrokontrolery, które (prawie) nie pobierają prądu, część 6

## 12-bitowy przetwornik A/C



Mikrokontrolery MSP430 mogą być wyposażone w dużą liczbę różnorodnych peryferiów. Czyni to z nich uniwersalne narzędzie do zadań o zmieniającym się charakterze. Standardem w elektronice jest konieczność „penetracji” świata analogowego, co powoduje, że niezbędnym wyposażeniem mikrokontrolerów są przetworniki analogowo-cyfrowe.

Pierwszy przykład wykorzystania 12-bitowego przetwornika SAR wbudowanego w mikrokontroler MSP430 przedstawiamy w artykule, kolejne za miesiąc.

Mikrokontrolery z rodziny MSP430 mogą być wyposażone w dwa typy przetworników: SAR, 10- lub 12-bitowy, alternatywnie 16-bitowy  $\Sigma$ - $\Delta$  (*Sigma-Delta*). Ponieważ mikrokontroler MSP430F449, zastosowany w zestawie uruchomieniowym (opisanym w EP12/2007), wyposażono w 12-bitowy przetwornik SAR, w artykule skupimy się na prezentacji jego obsługi.

W konfiguracji testowej wykorzystano ośmiocyfrowy wyświetlacz siedmiosegmentowy LCD. Przykłady napisano w C w środowisku IAR (bezpłatna wersja *kickstart*, publikujemy ją na CD-EP2/2008B).

### Architektura ADC12

Sercem przetwornika ADC12 jest 12-bitowy rdzeń SAR. Dokonuje on konwersji napięcia do postaci cyfrowej, posługując się dwoma potencjałami referencyjnymi. Błąd konwersji nie powinien typowo przekroczyć  $\pm 2LSB$ . Czas konwersji zamyka się w 13 taktach zegarowych. Częstotliwość taktowania rdzenia nie powinna przekroczyć 6,3 MHz, co okre-

śla minimalny czas konwersji równy 2,1  $\mu s$  (czyli 475 kHz). Poprawna praca modułu wymaga zasilania w zakresie 2,2...3,6 V.

Moduł przetwornika ma możliwość podłączenia zewnętrznych napięć odniesienia. Wyższy potencjał powinien mieć wartość większą od 1,4 V, mniejszą zaś od napięcia zasilania, a niższy potencjał – wartość mniejszą od 1,2 V, ale oczywiście większą lub równą 0 V. Daje to możliwość dość swobodnego operowania zakresem i rozdzielczością pomiarową przetwornika. Dostępny jest również wewnętrzny moduł referencji o dwóch napięciach 1,5 lub 2,5 V (wybór bitem ADC12CTL0.REF2\_5V). Rozrzut produkcyjny napięć referencyjnych modułu wynosi 4%, zaś stabilność temperaturowa nie jest gorsza niż  $\pm 100$  ppm. Wyboru konfiguracji referencji dokonuje się w rejestrach każdej z pamięci bufora wyników (bity ADC12MCTL0-15.SREF).

Źródłem sygnału taktującego przetwornik może być jeden z głównych zegarów procesora lub niez-

ależny oscylator RC o częstotliwości ok. 5 MHz (bity ADC12CTL1.ADC12SSEL). W miarę potrzeby sygnał zegarowy można dodatkowo podzielić przez wartość całkowitą 1...8 (bity ADC12CTL1.ADC12DIV).

Wyzwalanie konwersji może być zrealizowane programowo (bit ADC12CTL0.ADC12SC) lub sprzętowo z wykorzystaniem jednego z liczników. Można przy tym zdefiniować rodzaj zbrocza wyzwalania (bit ADC12CTL1.ISSH). Sam proces konwersji podzielony jest na dwa etapy. W pierwszym jest zapamiętywana wartość w module

S/H, w drugim odbywa się „właściwa” konwersja. Czas trwania etapu S/H można wybrać (bit ADC12CTL1.SHP) poprzez odpowiednią konfigurację licznika (bity ADC12CTL1.SHT0,1) lub sterować nim bezpośrednio sygnałem wyzwalania SHI.

Multiplekser analogowy pozwala wybrać do konwersji, poza portami procesora, dodatkowo wejścia zewnętrznych referencji, wyjście czujnika temperatury i dzielnika napięcia zasilania. Napięcia wejściowe nie mogą przekroczyć napięcia zasilania.

Przetwornik może pracować w czterech trybach (bity ADC12CTL1.CONSEQ): pojedynczej konwersji jednego kanału, pojedynczej sekwencji kanałów, sekwencyjnej konwersji pojedynczego kanału oraz sekwencyjnej konwersji wielu kanałów. Dodatkowo bit ADC12CTL0.

#### Przetworniki SAR

Przetwornik SAR (*Successive Approximation Register*) występuje w mikrokontrolerach MSP430 wersji 10- lub 12-bitowej. Poza rozdzielczością, odróżnia je jeszcze moduł zapisu wyników konwersji. W ADC12 jest wydzielony oddzielny obszar pamięci o wielkości 16 słów (12-bitowych), zaś w ADC10 jest pojedynczy rejestr wyniku oraz moduł transferu w dowolny dostępny pamięci. Główne bloki funkcjonalne przetwornika SAR to: moduł referencyjny (1,5 V oraz 2,5 V), multiplekser analogowy, moduł S/H (*Sample-And-Hold*), dzielnik do pomiaru napięcia zasilania oraz czujnik temperatury mikrokontrolera.

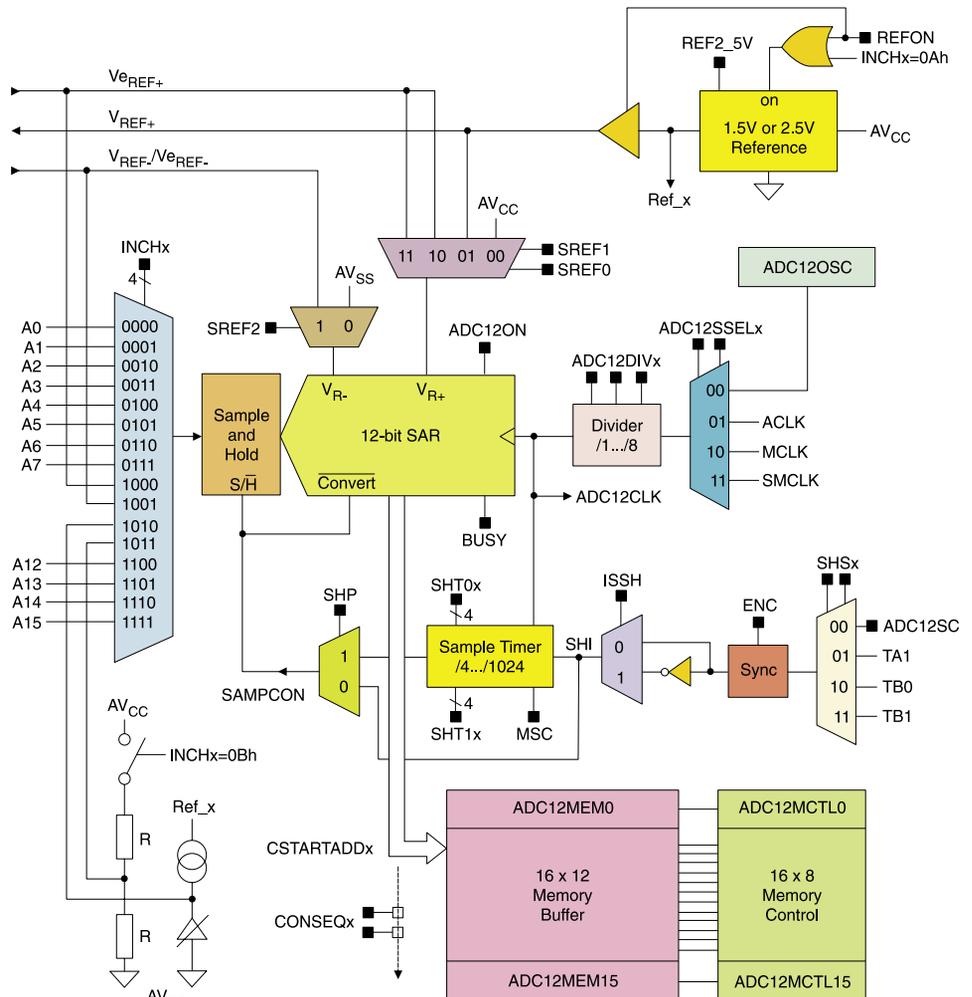
#### Przetworniki $\Sigma$ - $\Delta$

16-bitowy przetwornik  $\Sigma$ - $\Delta$  (*Sigma-Delta*) również cechuje własna referencja (1,2 V), multiplekser analogowy na wejściu i wbudowany czujnik temperatury. Dodatkowo wyposażono go we wzmacniacz PGA (*Programmable Gain Amplifier*) oraz może występować w wersji potrójnej (trzy rdzenie przetwarzające). Wyniki konwersji w tym przetworniku zapisywane są do pojedynczego rejestru.

MSC pozwala na zapętlenie pracy przetwornika bez sygnału wyzwalania.

Wyniki konwersji zapisywane są do 16-pozycyjnego bufora wyników. Każda komórka pamięci bufora ma swoją reprezentację w postaci rejestru konfiguracyjnego. Określa się w nim kanał wejściowy (ADC12MCTL0-15.INCH) oraz konfigurację napięć referencyjnych (ADC12MCTL0-15.SREF). Zapis rozpoczyna się od pozycji określonej poprzez bity ADC12CTL1.CSTAR-TADD. W przypadku sekwencji pomiarów, każdy następny wynik zapisywany jest do kolejnej komórki. Koniec sekwencji określa bit ADC12MCTL0-15.ESC.

Zapis do którejkolwiek komórki pamięci wyników może wywołać przerwanie (rejstry ADC12IE i ADC12IFG), przy czym istnieje możliwość rozróżnienia źródła przerwania (rejstry ADC12IV). Dodatkowo można wywołać przerwanie od przepełnienia lub od przekroczenia czasu S/H.



Rys. 21. Schemat blokowy przetwornika ADC12

**Pomiar temperatury**

W pierwszym przykładzie wykorzystamy czujnik temperatury wbudowany w strukturę mikrokontrolera. Jest nim złącze półprzewod-

nikowe zasilane źródłem prądowym o wydajności ok. 100 µA. W temperaturze 0°C spadek napięcia na czujniku wynosi 986 mV. Niestety

dopuszcza się 5% rozrzut produkcyjny tej wartości. Współczynnik nachylenia krzywej skalowania wynosi 3,55 mV/°C z 3% błędem.

Bez skalowania pomiar temperatury nie jest zbyt dokładny.

W pierwszej kolejności konfigurowany jest przetwornik. Założono, że każda konwersja będzie inicjowana w pętli głównej, a podczas jej trwania procesor będzie usypiany. Wymusza to konieczność wykorzystania oscylatora ADC12OSC do taktowania przetwornika (bity ADC12CTL1.ADC12SSEL). Ponieważ pomiar temperatury nie wymaga dużych prędkości zegara wybrano współczynnik podziału dzielnika przez 8 (bity ADC12CTL1.ADC12DIV). Częstotliwość zegara przetwornika ADC12CLK wynosi zatem:

$$ADC12CLK = \frac{5MHz}{8} = 625kHz$$

Dla przetwornika temperatury wymagany jest czas S/H co najmniej 30 µs. Do

Tab. 3. Rejestry przetwornika ADC12							
Rejestr	Nazwa	Typ	Adres	Stan			
Rejestr kontrolny nr 0	ADC12CTL0	R/W	01A0h	Reset z POR			
SHT1		SHT0					
MSC	REF2_5V	REFON	ADC12ON	ADC12OVIE	ADC12TOVIE	ENC	ADC12SC
Rejestr kontrolny nr 1	ADC12CTL1	R/W	01A2h	Reset z POR			
CSTARTADD		SHS	SHP	ISSH			
ADC12DIV		ADC12SSEL	CONSEQ	ADC12BUSY			
Flagi przerwania	ADC12IFG	R/W	01A4h	Reset z POR			
Flagi zezwoleń przerwania	ADC12IE	R/W	01A6h	Reset z POR			
Wektor ident. przerwania	ADC12IV	R	01A8h	Reset z POR			
Rejestry wyniku 0...15	ADC12MEM0..ADC12MEM15	R/W	0140h...015Eh	niezmieniony			
Rejestry kontrolne wyniku 0...15	ADC12MCTL0..ADC12MCTL15	R/W	080h...08Fh	Reset z POR			
EOS	SREF	INCH					

wyznaczenia wymaganego podziału *SHT* wykorzystamy poniższą formułę:

$$SHT=30[\mu s] \cdot 625[kHz]=18,75$$

Najbliższy (większy) współczynnik to 32 (bity *ADC12CTL0.SHT0,1*).

Ponieważ przetwornik ma być wyzwalany programowo, należy wybrać jako źródło wyzwalania sygnał *ADC12SC* (bity *ADC12CTL1.SHS*). Należy pamiętać również o bicie *ADC12CTL1.SHP=1*, który różni funkcję wyzwalania od bezpośredniego sterowania mechanizmem S/H przetwornika. Tryb pojedynczej konwersji ustawiamy poprzez bity *ADC12CTL0.MSC=0* oraz *ADC12CTL1.CONSEQ=0*.

Zaplanowano, że wynik konwersji będzie zapisywany do pierwszej komórki bufora (bity *ADC12CTL1.CSTARTADD*). Ponieważ przetwornik będzie pracował w trybie pojedynczej konwersji, należy skonfigurować tylko ten rejestr konfiguracyjny, który odnosi się do tej komórki (rejestr *ADC12MCTL0*). W rejestrze tym wybieramy kanał 10 multipleksera (bity *ADC12MCTL0.INCH*) oraz określamy źródło napięć referencyjnych (bity *ADC12MCTL0.SREF*). Założono wykorzystanie wewnętrznego źródła referencyjnego (*ADC12CTL0.REFON*) o napięciu 1,5 V.

Ostatni etap konfiguracji przetwornika to włączenie przerwania po zapisie wyniku do pamięci (bity *ADC12IE.ADC12IE0*). Obsługa przerwania ogranicza się do przepisania przetworzonej wartości do zmiennej globalnej (*temperatura*) oraz wyprowadzenie procesora z trybu uśpienia.

Pętla główna programu, poza wyzwalaniem konwersji, zawiera algorytm przekształcania wartości niemianowanej (w bitach) na temperaturę w °C. Funkcja temperatury jest określona zależnością:

**List. 7. Pomiar temperatury struktury mikrokontrolera**

```
// Plik naglowkowy
#include „msp430x44x.h”

// Zmienne globalne
signed short temperatura = 0;

// Przerwanie po zakonczonej konwersji
#pragma vector = ADC12_VECTOR
__interrupt void ISR_ADC12(void)
{
    temperatura = ADC12MEM0;
    __low_power_mode_off_on_exit(); // LPM exit
}

// Inicjacja przetwornika ADC12
void ADC12Init(void)
{
    ADC12CTL0 = SHT0_3 + SHT1_3 + REFON + ADC12ON; // ADC12, SHTime->32, SingleConv, Ref->1.5V,
turn on ADC12
    ADC12CTL1 = CSTARTADD_0 + SHS_0 + SHP + ADC12DIV_7 + ADC12SSEL_1 + CONSEQ_0; // MEM0, ADCosc, ACLK/8, Single sequence,
// 0..Vref, in -> temp
    ADC12MCTL0 = SREF_1 + INCH_10; // MEM0->przerwanie
    ADC12IE = 0x0001; // MEM0->przerwanie
    ADC12CTL0 |= ENC; // włącz konwersję (czekaj na wyzwolenie)
}

// Głowna funkcja
void main(void)
{
    unsigned short ShortTmp1, ShortTmp2; // Zmienne pomocnicze

    WDTCTL = WDTPW + WDTHOLD; // watchdog, off
    __enable_interrupt(); // system przerw, włączony

    ADC12Init(); // inicjacja przetwornika ADC

    while (true) // głowna petla
    {
        ADC12CTL0 |= ADC12SC; // wyzwolenie pojedynczej konw.
        __low_power_mode_3(); // LPM3

        ShortTmp2 = temperatura; // Przeliczenie wartosci
        ShortTmp2 = ShortTmp2 << 3; // * 13 / 128 -> 0,102
        ShortTmp1 = temperatura << 2;
        ShortTmp1 += temperatura;
        ShortTmp2 += ShortTmp1;
        ShortTmp2 = ShortTmp2 >> 7;
        temperatura = ShortTmp2 - 278; // offset
        __no_operation();
    }
}
```

$$T[^\circ C]=\frac{U_s[mV]-986[mV]}{3,55[mV/^\circ C]}$$

gdzie,  $U_s$  – spadek napięcia na czujniku wyznaczony z poniższej formuły:

$$U_s[mV]=\frac{U_{ref}[mV]}{2^{12}} \cdot C_{temp}$$

$$U_s[mV]=\frac{1500[mV]}{4096} \cdot C_{temp} = 0,366[mV] \cdot C_{temp}$$

gdzie,  $C_{temp}$  – wynik konwersji analogowo-cyfrowej przetwornika 12-bitowego. W wyniku podstawienia otrzymujemy ostateczną zależność:

$$T[^\circ C]=\frac{0,366[mV] \cdot C_{temp}-986[mV]}{3,55[mV/^\circ C]}$$

$$T[^\circ C]=0,103[^\circ C] \cdot C_{temp}-278$$

W języku C taką zależność można zaimplementować wprost, jednak

w przypadku, gdy mikrokontroler nie potrafi sprzętowo mnożyć i dzielić, kompilator dołączy do kodu dodatkową bibliotekę, a samo przeliczanie będzie długo trwać. Jeżeli nie zależy nam na optymalizacji, zarówno wielkości programu, jak i szybkości jego działania, to rzeczywiście nie ma sensu gimnastykować umysłu. Warto jednak wiedzieć, że w większości przypadków optymalizacja z bardzo dobrym skutkiem nie jest skomplikowana. Proszę zauważyć, że wartość 0,103 można zastąpić ułamkiem z dość dobrym przybliżeniem:

$$0,103 \approx \frac{13}{128} = \frac{2^3+2^2+1}{2^7} = 0,102$$

Przy rozkładzie należy kierować się podstawową zasadą: procesor potrafi bardzo szybko mnożyć i dzielić

przez liczbę 2. Wynika to z systemu dwójkowego jakim się posługuje. Dzielenie i mnożenie przez 2 to przeciwieństwo nic innego jak przesuwanie w lewo lub w prawo. Dlatego bardzo ważne aby w mianowniku naszego ułamka znalazła się całkowita potęga liczby 2. Licznik również warto rozłożyć na sumę całkowitych potęg 2. Tak zaimplementowana funkcja translacji będzie wykonywała się znacznie szybciej i zajmowała mniej pamięci kodu. Dopełniając idei optymalizacji wykorzystamy do obliczeń zmienne lokalne umieszczone w rejestrach oraz zapiszmy działania w postaci krótkich formuł. Wyznaczona wartość temperatury znajduje się w zmiennej *temperatura*.

**Piotr Tadrzak  
Contrans TI**