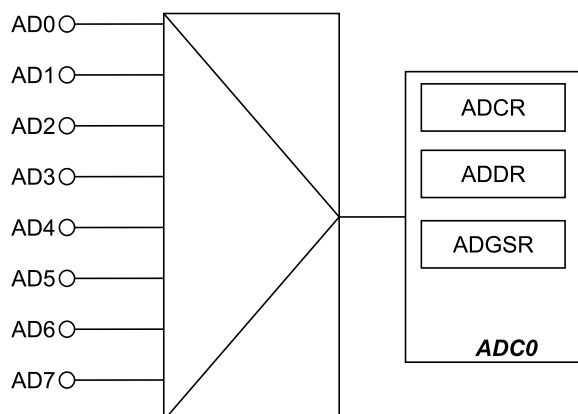


Mikrokontrolery z rdzeniem ARM, część 21

Przetwarzanie A/C i C/A

Przetwornik analogowo-cyfrowy

Mikrokontrolery LPC2000, nie wyróżniają się niczym szczególnym, jeżeli chodzi o przetworniki A/C i zawierają w sobie wszystko to, co obecnie jest uważane za niezbędne minimum w tej dziedzinie. Mniej-sze mikrokontrolery LPC2131/2 oraz LPC2141/2 posiadają jeden przetwornik, natomiast bardziej rozbudowane (21x4/6/8) zawierają po dwa przetworniki A/C. W starszej rodzinie LPC213x każdy z przetworników posiada 8-kanalowy multiplexer. W nowszych mikrokontrolerach LPC214x dodatkowe linie zostały wykorzystane do podłączenia interfejsu USB, dlatego pierwszy przetwornik ADC0 posiada zmniejszoną do 6 liczbę wejść, natomiast drugi przetwornik pozostał 8-wejściowy. Wszystkie przetworniki A/C posługują się trybem sukcesywnej aproksymacji, posiadają 10-bitową rozdzielczość i charakteryzują się czasem przetwarzania 2,44 μ s, można więc powiedzieć że ich parametry są przeciętne. Błąd przetwarzania przetwornika jest na poziomie ostatnich dwóch bitów, a praktyczne próby wykazały, że poziom szumów jest stosunkowo niski w porównaniu z innymi mikrokontrolerami np. ATmega8. Na rys. 63 przedstawiono budowę wewnętrzną przetworników A/C mikrokontrolerów LPC213x/214x.



Rys. 63. Budowa przetwornika A/C w mikrokontrolerach LPC213x/214x

W otaczającym nas świecie posługujemy się głównie wielkościami analogowymi, takimi jak temperatura, napięcie, ciśnienie itd. Dlatego w systemach mikroprocesorowych bardzo często zachodzi potrzeba pomiaru wielkości analogowych, które aby mogły być wykorzystane muszą być wcześniej przetworzone na postać cyfrową. Czasami istnieje również potrzeba, aby sygnał cyfrowy przetworzony przez system mikroprocesorowy zamienić ponownie na postać analogową np. w celu wysterowania jakiegoś układu wykonawczego. Do zamiany sygnału z postaci analogowej na cyfrową i odwrotnie przetworniki A/C i C/A. Większość współczesnych mikrokontrolerów o bardziej rozbudowanej architekturze posiada wbudowany przetwornik A/C, oraz trochę rzadziej przetwornik C/A. Na tle innych mikrokontrolerów możliwości analogowe LPC213x/214x są raczej przeciętne, które w zależności od wersji zawierają do dwóch 10-bitowych 8-kanalowych przetworników A/C oraz jeden 10-bitowy przetwornik C/A. Ponieważ mamy opanowane większość układów peryferyjnych mikrokontrolerów LPC213x w ostatnim odcinku kursu zajmować się będziemy przetwarzaniem sygnałów analogowych.

Do każdego przetwornika podłączony jest multiplexer umożliwiający, wybranie odpowiedniego kanału, z którego wykonywane będzie próbkowanie. W tab. 7 przedstawiono przyporządkowanie poszczególnych linii wejściowych przetworników A/C, linie zaznaczone kolorem szarym występują tylko w mikrokontrolerach 4/6/8.

Jak wspomniano w jednym z poprzednich odcinków każda linia I/O prezentowanych mikrokontrolerów może współpracować z logiką 5 V, ale wyłącznie w trybie GPIO. Gdy linia I/O pełni rolę analogowego wejścia przetwornika A/C, należy pamiętać, aby nie podłączać do niej napięcia o wartości większej od napięcia zasilającego mikrokontroler (3,3 V).

Przetworniki mikrokontrolera mogą pracować w trybie pojedynczego wyzwolenia, gdzie start przetwarzania wyznaczany jest w sposób

programowy, za pomocą ustawienia odpowiednich bitów konfiguracyjnych. Początek przetwarzania może być również inicjowany w sposób sprzętowy za pomocą sygnału pochodzącego od bloków MAT ukła-

Tab. 7. Przyporządkowanie poszczególnych linii wejściowych przetworników A/C (linie zaznaczone kolorem szarym występują tylko w mikrokontrolerach 21x4/21x6/21x8)

Wejście A/C	LPC213x	LPC214x
AD0.0	P0.27	–
AD0.1	P0.28	P0.28
AD0.2	P0.29	P0.29
AD0.3	P0.30	P0.30
AD0.4	P0.25	P0.25
AD0.5	P0.26	–
AD0.6	P0.4	P0.4
AD0.7	P0.5	P0.5
AD1.0	P0.6	P0.6
AD1.1	P0.8	P0.8
AD1.2	P0.10	P0.10
AD1.3	P0.12	P0.12
AD1.4	P0.13	P0.13
AD1.5	P0.15	P0.15
AD1.6	P0.21	P0.21
AD1.7	P0.22	AD1.7



CLKDIV								SEL									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
				EDGE	START						PDN					CLK	BURST
-	-	-	-					-	-								
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		

Rys. 64. Rejestry ADC0CR i ADC1CR

dów czasowo-licznikowych, lub za pomocą zboczy sygnału na odpowiednich liniach CAP mikrokontrolera. Zapewnienie sprzętowego wyzwania umożliwia dokładne określenie momentów próbkowania, co ma istotne znaczenie w przypadku próbkowania niektórych sygnałów. Przetworniki A/C, mogą być również skonfigurowane do pracy w trybie ciągłym, gdzie po uruchomieniu sygnał wejściowy jest cały czas próbkowany z możliwie największą częstotliwością. Informacja o zakończeniu przetwarzania może posłużyć do zgłoszenia przerwania. Skonfigurowanie odpowiedniego trybu pracy przetwornika umożliwia rejestr **ADCR**, dla przetwornika ADC0 **ADC0CR** (0xE0034000) natomiast **ADC1CR** (0xE0064000) dla przetwornika ADC1, którego bity konfiguracyjne przedstawiono na rys. 64.

SEL – Stan tych bitów określa, które kanały będą próbkowane i przetwarzane na postać cyfrową. Każdy bit odpowiada za dany kanał (0...7). W przypadku, gdy przetwarzanie wykonywane będzie w sposób programowy tylko jeden bit określający kanał może być ustawiony, natomiast w przypadku sprzętowego przetwarzania można określić większą ilość kanałów, które będą przetwarzane na postać cyfrową. Na przykład ustawienie przetwornika w tryb ciągłej konwersji i ustawienie bitów 1, 3, 5 spowoduje cykliczne przetwarzanie wartości z kanałów AD1, AD3 i AD5.

CLKDIV – Bity te umożliwiają ustawienie częstotliwości taktującej przetwornik. Częstotliwość tą można wyznaczyć według wzoru: $F_{ADC} = (F_{pclk} / CLKDIV) + 1$, gdzie F_{pclk} jest częstotliwością taktującą sygnały peryferyjne. Aby przetwornik pracował prawidłowo częstotliwość taktowania musi być mniejsza od 4,5 MHz.

BURST – Ustawienie tego bitu powoduje rozpoczęcie przetwarzania cyklicznego z wszystkich kanałów

ustawionych przez bity SEL. Cykliczne przetwarzanie trwa, aż do momentu skasowania tego bitu. Aby uruchomić ten tryb musimy pamiętać, aby bity START miały wartość 000, gdyż w przeciwnym przypadku tryb przetwarzania cyklicznego nie zostanie w ogóle włączony.

CLKS – Bity te umożliwiają ustalenie ilości cykli przetwornika podczas przetwarzania danych, a co z tym się wiąże ustalenie rozdzielczości przetwornika. W przypadku, gdy CLKS jest równe 000 wówczas przetwornik posiada 10-bitową rozdzielczość, a przetwarzanie zajmuje 11 cykli przetwornika (F_{adc}), natomiast, gdy CLKS jest równe 111, wówczas przetwarzanie trwa 4 cykle przetwornika, a jego rozdzielczość wynosi 3 bity.

PDN – Ustawienie tego bitu łączy przetwornik A/C, natomiast jego wyzerowanie powoduje wyłączenie przetwornika.

START – Stan tych bitów określa, sposób rozpoczęcia przetwarzania przez przetwornik A/C, na przykład w sposób programowy, lub poprzez zdarzenie od układu czasowo-licznikowego:

- 000 – Przetwarzanie zatrzymane
- 001 – Przetwarzanie zostanie rozpoczęte natychmiast (wyzwalanie programowe)
- 010 – Przetwarzanie zostanie rozpoczęte w momencie pojawienia się zbocza na linii P0.16
- 010 – Przetwarzanie zostanie rozpoczęte w momencie pojawienia się zbocza na linii P0.22
- 100 – Przetwarzanie zostanie rozpoczęte w wyniku wystąpienia zbocza od układu porównującego licznika MAT0.1
- 101 – Przetwarzanie zostanie rozpoczęte w wyniku wystąpienia zbocza od układu porównującego licznika MAT0.3

DONE	OV	...	-	CHN				...	-	A_VAL								...	-	
31	30	...	27	26	25	24	...	16	15	14	13	12	11	10	9	8	7	6	...	0

Rys. 65. Rejestr AD1DR

110 – Przetwarzanie zostanie rozpoczęte w wyniku wystąpienia zbocza od układu porównującego licznika MAT1.0

111 – Przetwarzanie zostanie rozpoczęte w wyniku wystąpienia zbocza od układu porównującego licznika MAT1.1

EDGE – Bit ten określa rodzaj zbocza, po którym nastąpi rozpoczęcie przetwarzania:

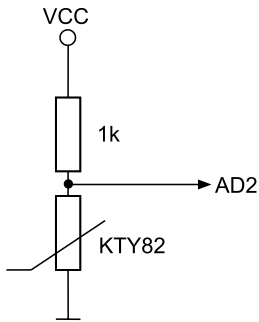
1 – Konwersja rozpocznie się w momencie wystąpienia zbocza opadającego.

0 – Konwersja rozpocznie się w momencie wystąpienia zbocza narastającego.

Mikrokontroler posiada bogate możliwości konfiguracyjne, a rozpoczęcie przetwarzania może odbywać nie tylko w sposób programowy, ale i również sprzętowy, co umożliwi dokładną synchronizację z przetwarzanym przebiegiem. Przetworniki mikrokontrolera nie posiadają wbudowanego źródła napięcia referencyjnego, a napięcie to jest pobierane z pinu Vref. Najczęściej napięcie Vref podłączamy do źródła napięcia zasilającego część analogową mikrokontrolera, ale gdy zależy nam na bardziej precyzyjnych pomiarach należy podłączyć je do zewnętrznego źródła napięcia referencyjnego. Jest to małe niedopatrzanie ze strony producenta mikrokontrolera, ponieważ nawet prosty 8-bitowy mikrokontroler ATmega8 posiada wbudowane źródło referencyjne o wartości 2,5 V, co w niektórych zastosowaniach jest bardzo wygodne. Do odczytywania wyników pomiarów przetwarzania służy rejestr **ADDR** **AD0DR** (0xE0034004) dla przetwornika AD0 oraz **AD1DR** (0xE0064004) dla przetwornika AD1, którego mapę bitową przedstawiono na rys. 65.

A_VAL – Gdy bit **DONE** jest ustawiony wówczas bity te zawierają bitową reprezentację napięcia występującego w danym kanale przetwornika.

CHN – Sprawdzając stan tych bitów możemy określić, z którego kanału pomiarowego pochodzi pomiar. Odczytując je możemy określić, z którego kanału pochodzi pomiar i przepisać go do wybranej komórki pamięci.



Rys. 66. Sposób dołączenia do mikrokontrolera testowego czujnika temperatury

OV – Bit ten wykorzystywany jest w trybie ciągłych pomiarów i jego ustawienie oznacza, że poprzednia wartość nie została odczytana przed wykonaniem kolejnego pomiaru i w związku z tym została utracona. Skasowanie tego bitu następuje automatycznie w wyniku odczytania tego rejestru.

DONE – Bit ten oznacza zakończenie wykonywania pomiarów przez przetwornik A/C, i jest on automatycznie zerowany w wyniku odczytania tego rejestru, lub zapisu do rejestru sterującego. Odczytując ten bit możemy stwierdzić, że rejestr ADDR zawiera prawidłowy wynik pomiaru.

Na podstawie zawartości bitów **CHN**, możemy określić kanał, z którego pochodzi wynik pomiaru, natomiast wartość napięcia występującego w tym kanale przetwornika możemy wyznaczyć według wzoru: $U_{out} = V_{ref} * (A_VAL / 1023)$. Zestaw rejestrów mikrokontrolera LPC213x umożliwiający podstawową obsługę przetworników A/C jest niewielki, a posługiwanie się nimi jest bardzo proste. Po zapoznaniu się z podstawowymi rejestrami mikrokontrolerów LPC213x możemy przejść do napisania bardzo prostego programu (udostępniamy jego źródło na CD-EP8/2007B, w pliku *ep9a.zip*), który w sposób programowy będzie dokonywał pomiaru napięć analogowych i wyświetlał je na terminalu. Do linii AD1 w zestawie ZL9ARM podłączony jest potencjometr umożliwiający ustawienie napięcia, którego zawartość zostanie bezpośrednio wyświetlona na terminalu. Do linii AD2 natomiast podłączony jest dzielnik napięcia, którego jedna gałąź stanowi rezystancyjny czujnik temperatury KTY82 (rys. 66) więc zamiast napięcia na terminalu (po przelicze-

niu) zostanie wyświetlona rezystancja czujnika.

Pomiary przetwornikiem A/C będą inicjalizowane w sposób programowy i nie będziemy używać przerwań tylko oczekiwać w pętli na zakończenie przetwarzania. Program przedstawiono na list. 13.

Do inicjalizowania ustawień czujnika służy funkcja *initADC()*, której jedynym zadaniem jest ustawienie częstotliwości taktującej mikrokontroler na wartość 1 MHz, włączenie przetwornika A/C oraz ustawienie w rejestrze PINSEL1 linii portu P0.28 i P0.29 tak aby pełniły rolę wejścia przetwornika. Do odczytania wartości napięcia za pomocą przetwornika służy funkcja *getADC()*, której argumentem jest numer kanału z którego chcemy odczytać wartość napięcia. W funkcji tej najpierw wybierany jest numer kanału (w trybie pojedynczego próbkowania może to być tylko jeden kanał), oraz ustawiany jest bit **START** rozpoczynający przetwarzanie. Następnie funkcja czeka w pętli *while* na ustawienie bitu **DONE**, będącego informacją o zakończeniu przetwarzania.

Ostatnią czynnością jest zwrócenie przez funkcję wartości zmierzzonego napięcia. W funkcji *main* inicjalizowany jest port szeregowy (9600 bd), wypisywany jest komunikat startowy, oraz inicjalizowany jest przetwornik A/C. Następnie program przechodzi do wykonania pętli głównej, w której wykonywane są pomiary z kanału czujnika temperatury (AD2), oraz potencjometru (AD1), wartości te są przeliczane na napięcie i rezystancje, oraz wyświetlane za pomocą funkcji *printf*. Na końcu pętli umieszczona jest pętla opóźniająca tak, więc pomiary wyświetlane są z częstotliwością około 1 Hz. Wykonanie pomiarów z wykorzystaniem przetworników A/C jest bardzo proste, i sprowadza się do obsługi dwóch rejestrów. Czytelników zachęcam do wykonania samodzielnego ćwiczenia polegającego na rozwinięciu programu o wyświetlanie rzeczywistej temperatury a nie rezystancji czujnika, co można w prosty sposób wykonać za pomocą aproksymacji wielomianowej. W rzeczywistym programie posługiwanie się przetwornikiem A/C,

List. 13. Program obsługi przetwornika A/C

```
#include „lpc213x.h”
#include „uart.h”
#include <stdlib.h>
#include <string.h>
#include <stdio.h>

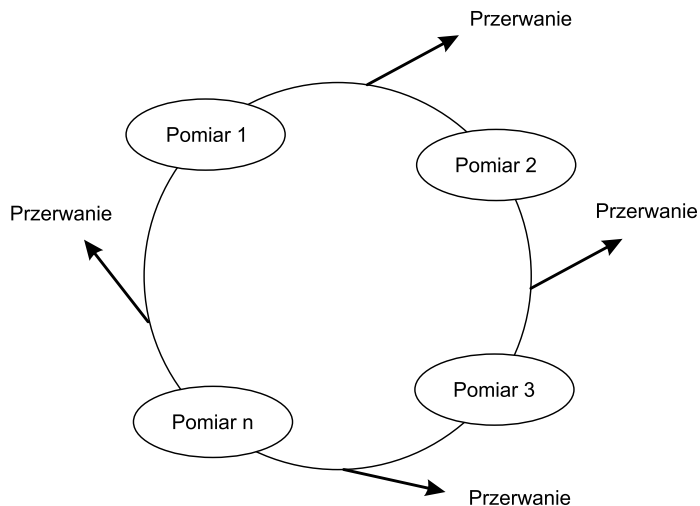
#define P028_ADC01_SEL (1<<24)
#define P029_ADC02_SEL (1<<26)

/* Initialize ADC */
inline static void initADC(void)
{
    ADCR = 60<<ADOCR_CLKDIV_BIT | 1<<ADOCR_PDN_BIT;
    PINSEL1 |= P028_ADC01_SEL | P029_ADC02_SEL;
}

/* Get ADC Value */
int getADC(int Chan)
{
    ADCR &= ~ADOCR_SEL_MASK;
    ADCR |= 1<<Chan & ADCR_SEL_MASK;
    ADCR |= 1<<ADOCR_START_BIT;
    while(!(AD0DR & AD0DR_DONE));
    return (AD0DR >> 6) & 0x3FF;
}

/* Funkcja glowna main */
int main(void)
{
    //Inicjalizacja UART
    Uart0Init(UART_BAUD(9600));
    //Napis zachety
    printf(„Mikrokontrolery z rdzeniem ARM cz IX\r\n”);
    printf(„Przetwornik A/C (Czujnik temperatury)\r\n”);
    initADC();
    int kty,pot;
    const float Vref = 3.3;
    while(1)

        kty = getADC(2);
        pot = getADC(1);
        printf(„Vpot = %.2f V\r\n”, (pot/1023.0)*Vref);
        float Ukty = (kty/1023.0)*Vref;
        printf(„Rkty = %.2f kOhm\r\n”,Ukty/(Vref-Ukty));
        for(volatile int i=0;i<1000000;i++);
    }
    return 0;
}
```



Rys. 67. Tryb pracy ciągłej przetwornika A/C

bez systemu przerwania nie jest zbyt wygodne, zwłaszcza, gdy zależy nam na wykonaniu na bieżąco pomiarów z kilku kanałów, oraz wykonywaniu równolegle innych czynności. W takiej sytuacji należy wykorzystać możliwość zgłaszania przerwania. W przetwornikach A/C nie występuje specjalny rejestr sterujący przerwaniem, i są one zgłaszane do kontrolera VIC zawsze w momencie ustawienia bitu DONE. Aby się nimi posługiwać wystarczy, odblokować przerwanie AD0 lub AD1 w kontrolerze przerwania. Na CD-EP8/2007 umieściliśmy program, który podobnie jak poprzednio dokonuje pomiaru napięcia potencjometru, oraz rezystancji czujnika temperatury, ale zamiast programowego sterowania przetwornikiem posłużono się tutaj ciągłym trybem pracy oraz przerwaniem.

W programie tym przetwornik po zakończeniu przetwarzania automatycznie zgłasza przerwanie, zmienia kanał pomiarowy na następny, oraz automatycznie rozpoczyna wykonywanie nowego pomiaru, co przedstawiono na rys. 67.

Kanały, w których będą wykonywane pomiary ustalane są za pomocą bitów SEL w rejestrze ADCR. Jest to jest jedyny tryb pracy przetwornika, w którym można ustawić więcej niż jeden bit w polu SEL. W momencie zgłoszenia przerwania program obsługi powinien określić numer kanału, z którego pochodzi wartość zmierzona (bity CHN), oraz umieścić wynik przetwarzania w odpowiedniej zmiennej. W naszym przypadku za obsługę przerwania od przetwornika odpowiada funkcja *AdcIrq*, w której najpierw sprawdza-

ny jest kanał, z którego pochodzi pomiar, i w zależności od tego wynik pomiaru przekazywany jest do zmiennej *AdcKty* lub *AdcPot*. Na zakończenie procedury obsługi wysyłana jest informacja do kontrolera przerwania VIC informująca

o zakończeniu procedury obsługi. Funkcja główna programu (*main*), inicjalizuje port szeregowy (9600 bd), wyświetla komunikat informacyjny, a następnie konfiguruje kontroler przerwania VIC, tak, aby przerwanie od przetwornika AD0 zakwalifikowane zostało jak wektoryzowane. Następnie inicjalizowany jest przetwornik A/C za pomocą funkcji *initADC()*, w której włączany jest przetwornik, ustalana jest częstotliwość pracy przetwornika, oraz wybierane są kanały biorące udział w pomiarze ciągłym. Ustawiane są także funkcje alternatywne portów, tak, aby pełniły rolę wejść A/C. Na końcu ustawiany jest bit BURST i od tego momentu przetwornik rozpoczyna ciągle przetwarzanie zgłaszając przerwania, natomiast program główny wchodzi w pętlę nieskończoną, w której co sekundę przelicza i wyświetla zawartość zmiennych *AdcKty*, *AdcPot*. Zmienne te są w sposób automatyczny aktualizowane przez procedurę obsługi przerwania przetwornika A/C. Program główny bezpośrednio nie wykonuje pomiarów przetwornikiem A/C a wszystko odbywa się

w sposób automatyczny. Obsługa przetwornika A/C w trybie ciągłym z wykorzystaniem systemu przerwania jest niewiele bardziej skomplikowana, a uwalnia program główny od zajmowania się przetwornikiem. Jak już wcześniej wspomniano większe mikrokontrolery LPC2134(2144)/6/8 posiadają wbudowane dwa przetworniki analogowo-cyfrowe, dlatego wprowadzono w nich dodatkowy rejestr **ADGSR** (0xE0034008), który umożliwia równoczesne rozpoczęcie wykonywania pomiarów przez oba przetworniki AD0 i AD1. Mapę bitową rejestru **ADGSR** przedstawiono na rys. 68.

Poszczególne bity mają takie same znaczenie w rejestrach **AD0CR** i **AD1CR**, poszczególnych przetworników, z tym, że wykonywane na nim operacje odnoszą się do obu przetworników równocześnie, co umożliwia ich synchronizację. Wykorzystanie trybu ciągłego, powoduje, że przerwania są zgłaszane przez przetwornik bardzo często wraz z zakończeniem każdego przetwarzania. Zadaniem programu obsługi przerwania jest przepisanie wyniku przetwarzania każdego kanału do odpowiedniej komórki pamięci, co jest rozwiązaniem stosunkowo nie efektywnym. Dużo lepiej byłoby wyposażać przetwornik w dodatkowy zestaw rejestrów umożliwiających przechowywanie danych pomiarowych z poszczególnych kanałów, a przerwanie zgłaszać dopiero w momencie wykonania wszystkich pomiarów. Konstruktorzy Philipsa zauważyli ten problem i nowsze mikrokontrolery rodziny LPC214x, wyposażyli w zestaw dodatkowych rejestrów, umożliwiających zgłaszanie przerwania przez poszczególne kanały niezależnie, oraz dodatkowe rejestry przechowujące wyniki pomiarów z poszczególnych kanałów. Zastosowanie rozszerzonego zesta-

-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

				EDGE		START				PDN			CLK		BURST
-	-	-	-					-	-						
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Rys. 68. Rejestr ADGSR

-	...	ADINT	OVERRUN								DONE							
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
31	...	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Rys. 69. Rejestr AD1STAT

INTERNET maker

Aktualności: najciekawsze i starannie wyselekcjonowane nowości z branży internetowej

Inspiracje: przegląd najbardziej efektywnych stron, przeróbki serwisów i prezentacje projektów, o których opowiadają sami autorzy

Magazyn: dowiedz się jak rozpocząć własną karierę w sieci a następnie podpatrz, jak swoje strony planują profesjonalisti

Warsztat: dzięki naszym artykułom oraz przyjaznym przewodnikom krok po kroku w prosty sposób dowiesz się jak tworzyć jeszcze lepsze strony i serwisy internetowe. W dziale Warsztat znajdziesz także cykliczne artykuły o wzorcach projektowych i aplikacjach internetowych

Pytania i odpowiedzi: poznaj rozwiązania najczęściej spotykanych problemów

Felietony: jesteś ciekaw, co o wydarzeniach w sieci myślą twórcy serwisów, które codziennie odwiedzasz? Przeczytaj ich felietony!



W numerze 3/2007 m.in.:

- AJAX w 5 minut – biblioteka XAJAX
- udostępnił swoje skrypty – nie tylko SourceForge.net
- Hibernate – sposób na bazy danych w Javie
- HTML i CSS – witaj w świecie tabel
- ASP.NET 2.0 – pierwsza część minikursu
- instalacja forum phpBB – krok po kroku
- Google SiteMaps – indeksowanie strony – krok po kroku

Nie masz jeszcze prenumeraty?
Czas zmienić zdanie, promocje czekają...

<http://www.internetmaker.pl>

Internet Maker można nabyć we wszystkich EMPIK-ach i większych kioskach z prasą.

Wszelkich informacji udziela

Dział prenumeraty:

tel. 022 568 99 22, faks 022 568 99 00

e-mail: prenumerata@avt.com.pl

01-939 Warszawa, ul. Burleska 9

-	...	ADGINTEN	ADINTEN							
			7	6	5	4	3	2	1	0
31	...	8	7	6	5	4	3	2	1	0

Rys. 70. Rejestr AD1INTEN

wu rejestrów umożliwia sterowanie każdym kanałem niezależnie. Do sprawdzania statusu poszczególnych kanałów służy rejestr ADSTAT (AD0: AD0STAT – 0xE0034004 AD1: AD1STAT – 0xE0060004), którego poszczególne bity przedstawiono na rys. 69.

DONE7...DONE0 – Bity te podobnie jak globalny bit DONE informują o zakończeniu wykonywania pomiarów w poszczególnych kanałach pomiarowych AD0..AD7, oraz o tym, że w rejestrze kanału danego pomiaru znajduje się prawidłowa wartość wyniku pomiaru.

OVERRUN0...OVERRUN7 – Bity te podobne jak globalny bit OV informują, o tym, że w danym rejestrze kanału poprzednia wartość pomiarowa nie została odczytana przed zakończeniem bieżącego przetwarzania, i została nadpisana przez nową wartość.

ADINT – Flaga zgłoszenia przerwania do kontrolera VIC, ustawiana w stan 1 w momencie, gdy przetwornik zgłasza przerwanie.

Sprawdzając stan tego rejestru mamy, globalny pogląd na stan przetwarzania wszystkich kanałów pomiarowych przetwornika. Kolejnym rejestrem umożliwiającym maskowanie przerwania z poszczególnych kanałów jest rejestr ADINTEN (AD0: AD0INTEN – 0xE003400C AD1: AD1INTEN – 0xE006000C), którego mapę bitową przedstawiono na rys. 70.

ADINTEN0...ADINTEN7 – Ustawienie danego bitu powoduje możliwość zgłaszania przerwania przez określony kanał przetwornika w momencie zakończenia przetwarzania, i jest flagą zezwolenia na przerwanie, natomiast wyzerowanie odpowiedniego bitu powoduje zablokowanie zgłaszania przerwania od określonego kanału.

ADGINTEN – Ustawienie tego bitu w stan 1 powoduje, że tylko flaga globalnego zgłoszenia przerwania powoduje generowanie przerwania, co jest zgodne z try-

bem kompatybilności ze starszym bratem LPC213x, natomiast jego wyzerowanie umożliwia zgłaszanie przerwania przez poszczególne kanały, których maskę zezwoleń określają bity ADINTEN0...ADINTEN7. Po wyzerowaniu mikrokontrolera bit ten jest ustawiony w stan 1, co zapewnia tryb kompatybilności ze starszymi modelami.

W mikrokontrolerach LPC213x, nie było żadnych rejestrów kontrolujących przerwania, ponieważ przetwornik potrafił zgłaszać tylko jedno przerwanie w momencie zakończenia wykonywania przetwarzania. Tutaj natomiast stało się konieczne wprowadzenie dodatkowego rejestru konfiguracyjnego, ponieważ każdy kanał przetwornika może zgłaszać osobno przerwanie po zakończeniu wykonania pomiaru. Do przechowywania wyników pomiarów z poszczególnych kanałów służą rejestry ADDR0...ADDR7, których mapę bitów przedstawiono na rys. 71.

Znaczenie poszczególnych bitów jest takie samo jak w globalnym rejestrze ADDR, jednak został on pozbawiony bitów CHN pozwalających odczytać numer kanału, ponieważ kanał jest tutaj ściśle określony przez nazwę rejestru. Warto tutaj wspomnieć że rejestr ADDR w LPC214x został przemianowany na ADGDR i został zachowany dla kompatybilności z poprzednimi mikrokontrolerami. Zastosowanie dodatkowych rejestrów dla przechowywania pomiarów pozwala napisać program w taki sposób, aby dopiero po zakończeniu przetwarzania przez wszystkie interesujące nas kanały było zgłaszane przerwanie. Czytelników posiadających nowszy zestaw ewaluacyjny z mikrokontrolerem LPC214x zachęcam do poprawienia ostatniego przykładu w taki sposób, aby wykorzystać nowe możliwości mikrokontrolerów LPC214x.

Lucjan Bryndza, EP
lucjan.bryndza@ep.com.pl

DONE	OV	...	-	RESULT								...	-		
31	30	...	16	15	14	13	12	11	10	9	8	7	6	...	0

Rys. 71. Rejestry ADDR0...ADDR7