



Zachodniopomorski  
Uniwersytet  
Technologiczny

**WYDZIAŁ ELEKTRYCZNY**  
Katedra Inżynierii Systemów, Sygnałów i Elektroniki

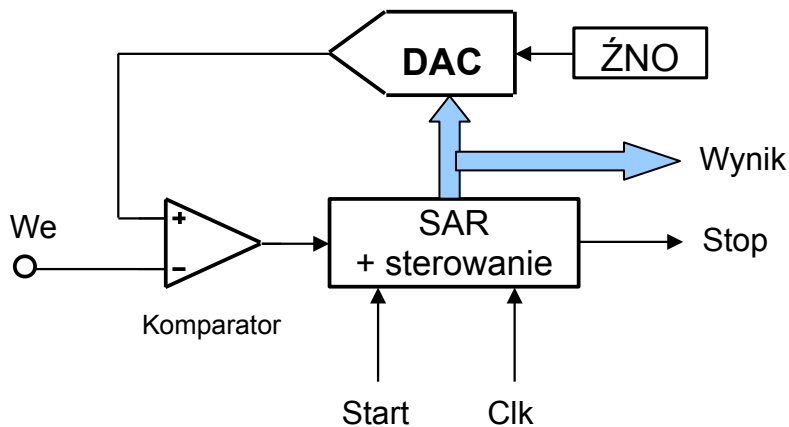
LABORATORIUM

TECHNIKA MIKROPROCESOROWA

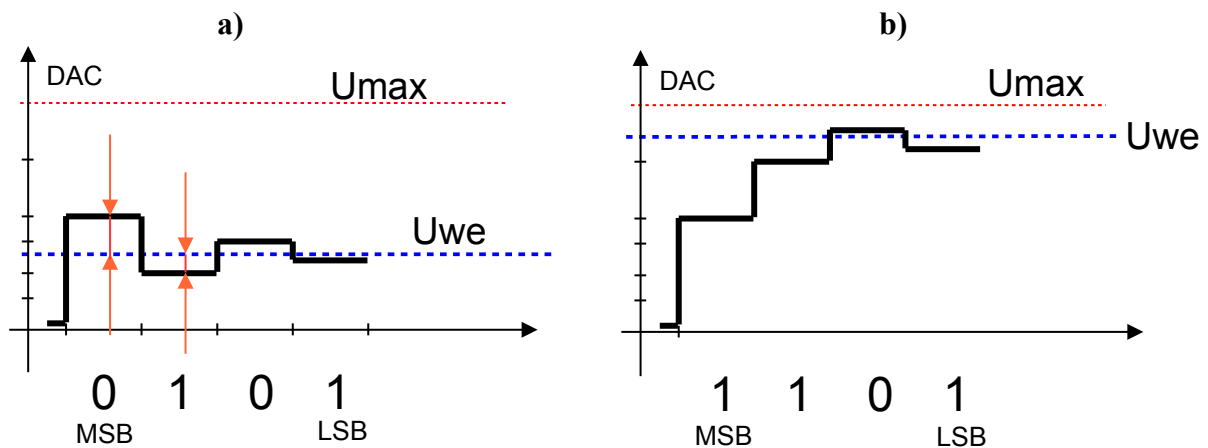
**PRZETWORNIK ADC**  
**w mikrokontrolerach Atmega16 - 32**

Opracował:  
mgr inż. Andrzej Biedka

Praktycznie każdy współczesny mikrokontroler posiada wbudowany przetwornik analogowo-cyfrowy. Stosowane są najczęściej przetworniki wykorzystujące metodę sukcesywnej aproksymacji lub sigma-delta. Ogólny schemat blokowy i przebiegi czasowe w przetworniku typu SAR przedstawiają rysunki 8-1 i 8-2.



Rys. 8-1. Schemat blokowy przetwornika ADC

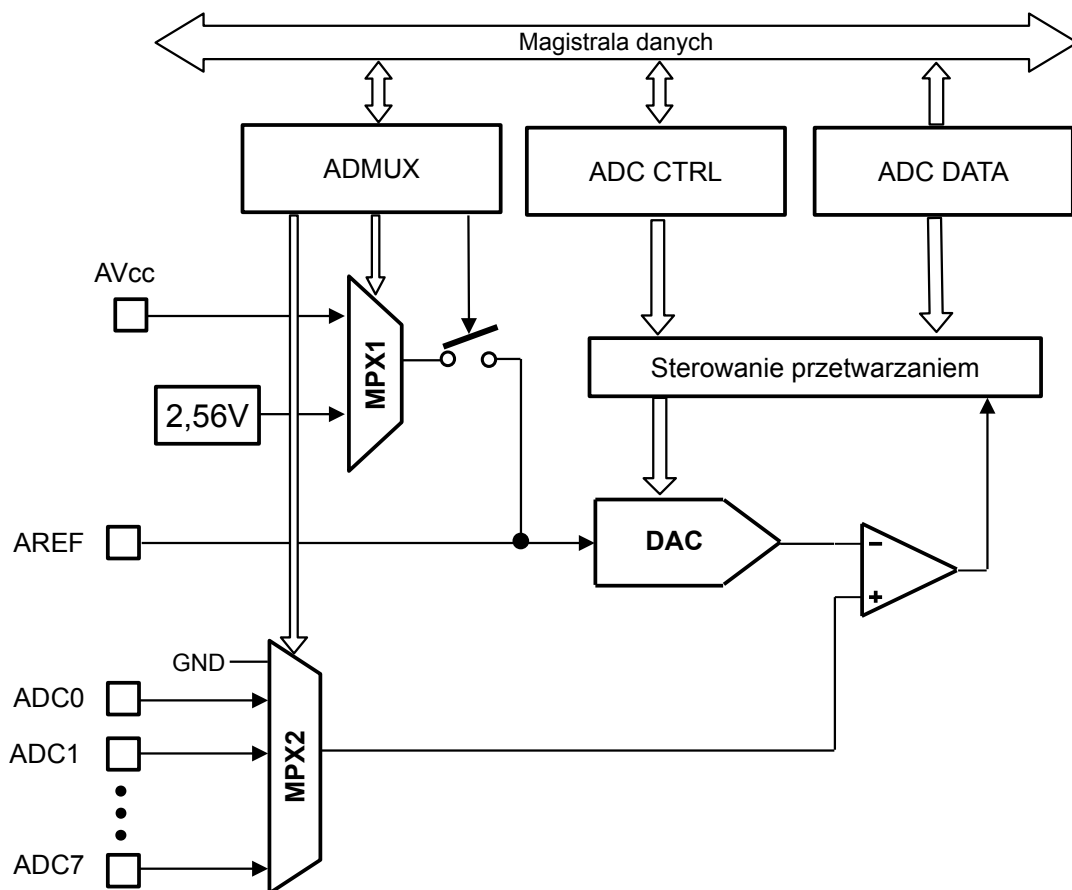


Rys. 8-2. Schemat blokowy przetwornika ADC i przykładowe przebiegi przetwarzania

Proces przetwarzania polega na porównywaniu napięcia wejściowego z napięciem wytwarzanym w wewnętrznym przetworniku cyfrowo-analogowym (**DAC**). Porównywanie dokonywane jest w kolejnych krokach, których liczba równa jest rozdzielczości przetwornika. Do pierwszego porównania brane jest napięcie równe połowie napięcia zakresu przetwarzania (**U<sub>max</sub>**). Jeśli napięcie wejściowe jest mniejsze niż napięcie z **DAC**, w kolejnym kroku odejmowana jest połowa z napięcia pierwszego kroku (czyli 1/4 napięcia zakresu) – rysunek 8-2a. Jeśli napięcie wejściowe jest większe następuje dodanie takiej wartości napięcia – rysunek 8-2b. W kolejnych krokach napięcie wytwarzane przez przetwornik wewnętrzny zbliża się do wartości napięcia wejściowego z dokładnością najmniej znaczącego bitu. Po zakończeniu cyklu pracy przetwornik sygnalizuje gotowość odbioru słowa cyfrowego przez ustawienie bitu w rejestrze sterującym lub zgłasza przerwanie.

Przetwornik ADC jest najczęściej wyposażony w wejściowy multiplexer analogowy pozwalający na pomiar kilku napięć – oczywiście niejednocześnie. Źródło napięcia odniesienia niezbędne do pracy przetwornika może być wbudowane do struktury mikrokontrolera lub przyłączone z zewnątrz.

Schemat blokowy przetwornika ADC mikrokontrolera Atmega 16/32 obejmujący najistotniejsze elementy przedstawiony jest na rysunku 8-3.



Rys. 8-3. Uproszczony schemat blokowy przetwornika ADC w mikrokontrolerze ATmega 16/32

Przetwornik ADC Atmega 16/32 może przetwarzać do ośmiu napięć podawanych na wejścia ADC0-ADC7. Możliwa jest praca niesymetryczna w zakresie napięć 0 – Vcc lub praca różnicowa, gdzie wykorzystywane są dwa wejścia analogowe. Zawsze należy pamiętać o ograniczeniu wartości napięć podawanych do wejść układów CMOS: napięcie wejściowe nie może być niższe niż -0,3V poniżej poziomu GND oraz nie może być wyższe niż +0,3V powyżej napięcia Vcc.

Źródłem napięcia odniesienia może być napięcie zasilania części analogowej AVCC, źródło wewnętrzne o wartości 2,56V lub zewnętrzne źródło o wartości napięcia w zakresie od 2,0V do (AVCC-0,5V).

Konfigurację i sterowanie pracą bloku przetwornika umożliwiają rejestry sterujące ADMUX, ADCSRA, ADCL, ADCH i SFIOR

Rejestr sterowania multiplekserami wejściowymi i źródłem napięcia odniesienia.

### ADMUX

REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
-------	-------	-------	------	------	------	------	------

- REFS1, REFS0** - wybór źródła napięcia odniesienia. Patrz tabela 8-1
- ADLAR** - wybór metody zapisu słowa wyjściowego
- MUX4 – MUX0** - wybór ustawień wejściowego multipleksera analogowego. Patrz tabela 8-2

Tabela 8-1

REFS1	REFS0	Źródło napięcia odniesienia
0	0	Źródło zewnętrzne, źródło wewnętrzne jest wyłączone
0	1	AVCC
1	0	zarezerwowane
1	1	Źródło wewnętrzne 2,56V

Do pinu **AREF** mikrokontrolera powinien być przyłączony kondensator filtrujący o pojemności min. 100nF, drugą końcówką przyłączony do potencjału **AGND**.

Do ustawiania pracy multipleksera wejściowego wykorzystane są bity MUX4 – MUX0. Bity MUX4 i MUX3 służą do ustawiania pracy różnicowej i zmiany wzmocnienia w torze przetwornika. W pracy podstawowej (wejście niesymetryczne, zakres napięć mierzonych 0 – AVCC) stosowane są bity MUX2 – MUX0, przy wartościach MUX4 = MUX3 = 0.

Tabela 8-2

MUX2	MUX1	MUX0	Wejście analogowe
0	0	0	ADC0
0	0	1	ADC1
0	1	0	ADC2
0	1	1	ADC3
1	0	0	ADC4
1	0	1	ADC5
1	1	0	ADC6
1	1	1	ADC7

Rejestr sterujący **ADCSRA** zawiera następujące bity:

ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
------	------	-------	------	------	-------	-------	-------

- ADEN** - odblokowanie przetwornika ADC
- ADSC** - wpisanie '1' powoduje start przetwarzania, powrót bitu do stanu '0' sygnalizuje koniec przetwarzania
- ADATE** - zezwolenie na pracę z automatycznym wyzwalaniem. Patrz tabela 8-4
- ADIF** - flaga zgłoszenia przerwania po zakończeniu przetwarzania
- ADIE** - zezwolenie na zgłoszenia przerwania po zakończeniu przetwarzania
- ADPS2 - ADPS0** - wybór stopnia podziału preskalera. Patrz tabela 8-3

Dla zapewnienia dokładnej pracy przetwornika częstotliwość przebiegu taktującego powinna zawierać się w zakresie 50 – 200 kHz. Zatem w inicjalizacji pracy przetwornika należy dobrać właściwy stopień podziału preskalera, który taktowany jest głównym sygnałem zegarowym flk. Dostępne stopnie podziału podane są w tabeli 8-3.

Tabela 8-3

<b>ADPS2</b>	<b>ADPS1</b>	<b>ADPS0</b>	Współczynnik podziału
0	0	0	<b>2</b>
0	0	1	<b>2</b>
0	1	0	<b>4</b>
0	1	1	<b>8</b>
1	0	0	<b>16</b>
1	0	1	<b>32</b>
1	1	0	<b>64</b>
1	1	1	<b>128</b>

### Rejestry danych **ADCH** i **ADCL**.

Ponieważ rozdzielczość przetwornika mikrokontrolera wynosi 10 bitów, do przekazania wyniku przetwarzania potrzebne są dwa rejestry 8-bitowe. Dla wygody tworzenia oprogramowania przewidziano dwa sposoby zapisu rejestrów danych: z wyrównaniem do lewej lub prawej strony, wybierane bitem **ADLAR** znajdującym się w rejestrze **ADMUX**.

Bit **ADLAR = 0** – wyrównanie do prawej

						<b>ADC9</b>	<b>ADC8</b>	<b>ADCH</b>
<b>ADC7</b>	<b>ADC6</b>	<b>ADC5</b>	<b>ADC4</b>	<b>ADC3</b>	<b>ADC2</b>	<b>ADC1</b>	<b>ADC0</b>	<b>ADCL</b>

Bit **ADLAR = 1** – wyrównanie do lewej

<b>ADC9</b>	<b>ADC8</b>	<b>ADC7</b>	<b>ADC6</b>	<b>ADC5</b>	<b>ADC4</b>	<b>ADC3</b>	<b>ADC2</b>	<b>ADCH</b>
<b>ADC1</b>	<b>ADC0</b>							<b>ADCL</b>

Uruchamianie przetwarzania może być realizowane dwójako:

- jednokrotne uruchamianie cyklu pracy, po którym przetwornik oczekuje na następny rozkaz.
- Automatyczne wyzwalanie pomiaru z sygnalizacją zakończenia każdego cyklu przetwarzania. Źródłem sygnału wyzwalającego mogą być różne sygnały/zdarzenia.

Rejestr **SFIOR** zawiera m.in. bity wyboru źródła sygnału wyzwalania przetwornika ADC:

<b>ADTS2</b>	<b>ADTS1</b>	<b>ADTS0</b>	-	<b>ACME</b>	<b>PUD</b>	<b>PSR2</b>	<b>PSR10</b>
--------------	--------------	--------------	---	-------------	------------	-------------	--------------

**ADTS2-ADTS0** - wybór źródła sygnału wyzwalania przetwornika ADC. Patrz tabela 8-4

Dla uaktywnienia automatycznego wyzwalania przetwornika konieczne jest ustawienie bitu **ADATE** w rejestrze **ADCSRA**

Tabela 8-4

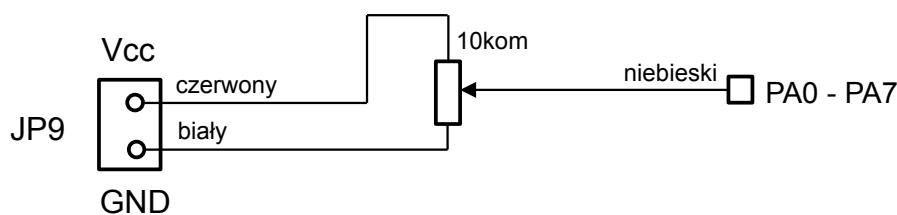
ADTS2	ADTS1	ADTS0	Źródło sygnału wyzwiania
0	0	0	Wyzwalanie ciągle
0	0	1	Wyzwalanie sygnałem komparatora analogowego
0	1	0	Wyzwalanie sygnałem przerwania zewn. EXT0
0	1	1	Wyzwalanie sygnałem zrównania Timera 0 - OCR
1	0	0	Wyzwalanie sygnałem przepełnienia Timera 0
1	0	1	Wyzwalanie sygnałem zrównania Timera 1 - OCR-B
1	1	0	Wyzwalanie sygnałem przepełnienia Timera 1
1	1	1	Wyzwalanie sygnałem przechwycenia Timera 1

**ZADANIA:**

1. Na podstawie karty katalogowej określić kolejność odczytywania rejestrów danych ADCL i ADCH.
2. Dla sygnału taktującego mikrokontroler równego 16 MHz wyznaczyć optymalny stopień podziału preskalera przetwornika.
3. Przyjmując wyniki uzyskane w zadaniu 2 oraz średni czas przetwarzania przetwornika równy 15 cyklom preskalera, obliczyć maksymalną prędkość przetwarzania ADC w jednostkach kSps.

**Przebieg ćwiczenia:**

- Przyłączyć potencjometr do zasilania i wybranego wejścia multipleksera przetwornika według schematu:



- Utworzyć algorytm i napisać program cyklicznego pomiaru napięcia uzyskiwanego z potencjometru z wyświetlaniem 8-bitowego wyniku w postaci binarnej.
- Zmodyfikować program wprowadzając funkcję wysyłania wyniku przetwarzania przez kanał transmisji szeregowej do komputera PC.
- Wykorzystać funkcje biblioteczne konwersji liczby 10 bitowej na ciąg znaków ASCII.

**Literatura:**

- [1] Francuz T. Język C dla mikrokontrolerów AVR. Od podstaw do zaawansowanych aplikacji. Helion, Gliwice, 2011
- [2] Baranowski R. Mikrokontrolery AVR ATmega w praktyce. BTC, Warszawa, 2005
- [3] Kardaś M. Mikrokontrolery AVR. Język C. Podstawy programowania. Atmel, Szczecin, 2011.
- [4] Witkowski A. Mikrokontrolery AVR. Programowanie w języku C. Przykłady zastosowań, PKJS, Katowice 2006
- [5] Karta katalogowa mikrokontrolera Atmega32 firmy ATMEL.