



Instytut Teleinformatyki



Wydział Fizyki, Matematyki i Informatyki
Politechnika Krakowska

Mikroprocesory i Mikrokontrolery

„Zastosowanie przetwornika analogowo-cyfrowego do odczytywania napięcia z potencjometru analogowego”

laboratorium: 12
autor: mgr inż. Paweł Pławiak

Kraków, 2015

Spis treści

Spis treści	2
1. Wiadomości wstępne	3
1.1. Przetwornik analogowo-cyfrowy.....	3
1.1.1. Przykładowy kod.....	4
1.2. Potencjometr analogowy.....	5
1.2.1. Schemat podpięcia.....	5
1.2.2. Pseudokod	5
1.3. Zagadnienia do przygotowania	6
2. Przebieg laboratorium	7
2.1. Zadanie 1. Na ocenę 3.0 (dst)	7
2.2. Zadanie 2. Na ocenę 4.0 (db).....	7
2.3. Zadanie 3. Na ocenę 5.0 (bdb)	7

1. Wiadomości wstępne

Pierwsza część niniejszej instrukcji zawiera podstawowe wiadomości teoretyczne dotyczące omawianego tematu. Poznanie tych wiadomości umożliwi prawidłowe zrealizowanie praktycznej części laboratorium.

1.1. Przetwornik analogowo-cyfrowy

Mikrokontroler ATmega32 wyposażony jest w 10 bitowy przetwornik A/C sukcesywnej aproksymacji i wielowejściowy multiplekser analogowy. Rozdzielczość przetwarzania wynosi 10-bitów z dopuszczalnym błędem ± 2 jednostki kwantyzacji (LSB). Przetwornik umożliwia wzmocnienie 0 dB, 20 dB (10-krotnie) lub 46 dB (200-krotnie) przy pomiarze różnicowym.

Bity konfiguracyjne rejestru ADMUX:

Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- REFS – napięcie odniesienia (wewnętrzne 2.56V, AVCC, AREF)
- ADLAR – jak 10-bitowy wynik będzie rozmieszczony w rejestrach (wyrównany do prawej lub lewej strony)
- MUX – tryb pracy (kanał/kanały wejścia i wzmocnienie)

Bity konfiguracyjne rejestru ADCSRA:

Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- ADEN – włączenie przetwornika
- ADSC – wpisanie 1 zaczyna konwersję
- ADATE – autotrigger (samoczynne zaczynanie konwersji)
- ADIF – flaga zakończonej konwersji (ustawiana przez sprzęt)
- ADIE – włączenie przerwania związanego z flagą ADIF
- ADPS – prescaler wpływający na szybkość i precyzję konwersji

Szczegółowe informacje na temat przetwornika A/C znajdują się w nocie katalogowej na stronie producenta – <http://www.atmel.com/Images/doc2503.pdf>

1.1.1. Przykładowy kod

Obsługa przetwornika A/C:

```
#include <avr/io.h>
//-----
// Get ADC conversion value
//-----
int getADC(char channel)
{
    unsigned int W = 0;
    ADMUX |= channel;
    ADCSRA |= (1 << ADSC);
    while(ADCSRA & (1 << ADIF));
    ADCSRA |= (1 << ADIF);
    W = ADCL;
    W |= (ADCH << 8);
    return W;
}
//-----
// ADC Initialization
//-----
void ADC_Init(void)
{
    ADMUX = (1 << REFS0);
    ADCSRA = (1 << ADEN) | (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0);
}
//-----
// End of file
//-----
```

1.2. Potencjometr analogowy

Zestaw ZL15AVR wyposażono w potencjometr P2, który może zostać wykorzystany do podawania napięcia z zakresu 0...5 V na wejścia przetwornika analogowo-cyfrowego mikrokontrolera AVR. Środkowe wyprowadzenie potencjometru dostępne jest na złączy Con7 (pin oznaczony P2).

Szczegółowe informacje na temat potencjometru analogowego znajdują się w dokumentacji zestaw ewaluacyjny ZL15AVR - http://dl.btc.pl/kamami_wa/zl15avr.pdf

1.2.1. Schemat podpięcia

Potencjometr analogowy:
P2 (con7) - PA0 (con15)

USART (RS232):
RxD (con7) - PD0 (con18)
TxD (con7) - PD1 (con18)

1.2.2. Pseudokod

Obsługa potencjometru analogowego – konwersja sygnału A/C dla napięcia wejściowego regulowanego przez potencjometr P2 oraz przesyłanie przekonwertowanego sygnału przez USART do PC.

```
#include <avr/io.h>  
#include <util/delay.h>  
#include <stdlib.h>  
#include "USART.h"  
#include "ADC.h"
```

```
int main(void)
```

```
{
```

```
Zadeklarowanie zmiennej znakowej w której będzie zapisywana wartość napięcia
```

```
Ustawienie wszystkich linii portu D jako wyjściowe
```

```
Inicjalizacja USART z odpowiednią częstotliwością
```

```
Inicjalizacja przetwornika
```

```
do{
```

Odczytywanie wartości napięcia z potencjometru I przesyłanie jej za pomocą USART do terminala

```
}while(1);  
  
return 0;  
}
```

Proszę pamiętać o ustawieniu odpowiedniej częstotliwości w terminalu (parametr baud) lub w programie.

1.3. Zagadnienia do przygotowania

Przed przystąpieniem do realizacji laboratorium należy zapoznać się z zagadnieniami dotyczącymi:

- o Przetwornika analogowo-cyfrowego
- o Potencjometru analogowego
- o USART

Literatura:

- [1] Rafał Baranowski, „Mikrokontrolery AVR ATmega w praktyce”
- [2] Tomasz Francuz, „Język C dla mikrokontrolerów AVRD od podstaw do zaawansowanych aplikacji”
- [3] Nota katalogowa mikrokontrolera na stronie producenta – <http://www.atmel.com/Images/doc2503.pdf>
- [4] Dokumentacja zestawu ewaluacyjnego ZL15AVR – http://dl.btc.pl/kamami_wa/zl15avr.pdf
- [5] Instrukcje do poprzednich ćwiczeń laboratoryjnych

2. Przebieg laboratorium

Druga część instrukcji zawiera zadania do praktycznej realizacji, które demonstrowują zastosowanie technik z omawianego zagadnienia.

2.1. Zadanie 1. Na ocenę 3.0 (dst)

Proszę napisać program wyświetlający napięcie odczytane z potencjometru i przekonwertowane przez przetwornik ADC w terminalu portu szeregowego na PC przez USART (port RS232). Wyświetlane napięcie ma być przeskalowane do przedziału od 0 do 5 V oraz wyświetlane w formacie dziesiętnym z kropką.

2.2. Zadanie 2. Na ocenę 4.0 (db)

Proszę napisać program który będzie umożliwiał rozpoczęcie konwersji A/C sygnału z potencjometru analogowego po naciśnięciu odpowiedniego klawisza oraz po zakończeniu (wstrzymaniu) przetwarzania będzie zapalał diody.

2.3. Zadanie 3. Na ocenę 5.0 (bdb)

Proszę zmodyfikować poprzedni program w celu przeciwdziałania zakłóceniom poprzez dodanie trybu uśpienia (praca jednostki centralnej jest zawieszona i włączony jest układ redukujący szum).