

Wydział Elektryczny
Katedra Telekomunikacji i Aparatury Elektronicznej

Instrukcja do zajęć laboratoryjnych z przedmiotu:
Architektura i Programowanie Procesorów Sygnałowych
Kod: TS1C510210

Numer ćwiczenia: 5

Temat ćwiczenia:

**Obsługa portu szeregowego.
Współpraca procesora sygnałowego z układami przetworników AC/CA.**

Opracował:
dr inż. Dariusz Jańczak

Temat: Obsługa portu szeregowego. Współpraca procesora sygnałowego z układami przetworników A/C/A.

1. Cel ćwiczenia

Celem ćwiczenia jest pogłębienie i ugruntowanie wiedzy studentów oraz nabycie przez nich umiejętności z zakresu wykorzystania elementów architektury oraz podstawowych funkcji procesorów sygnałowych na przykładzie zmiennoprzecinkowego procesora TMS320C6713. W ramach zajęć studenci zdobywają umiejętność pisania procedur wymiany danych z wykorzystaniem portów szeregowych. Studenci nabywają umiejętność tworzenia programów pozwalających na współpracę procesora DSP z przetwornikami A/C i C/A.

2. Zagadnienie do opracowania przed przystąpieniem do zajęć

Przed przystąpieniem do zajęć należy opracować następujące zagadnienia:

- ◆ architektura sygnałowego procesora zmiennoprzecinkowego TMS320C6713,
- ◆ zasady działania i programowanie portów szeregowych McBSP procesora zmiennoprzecinkowego TMS320C6713 (SPRU580),
- ◆ architektura i zasady programowania układu TLV320AIC23 (tlv320aic23.pdf),
- ◆ współpraca układu TLV320AIC23 (z przetwornikami A/C C/A) z procesorem DSP na płycie DSK,
- ◆ zasady konfiguracji i wykorzystania systemu obsługi przerwań.

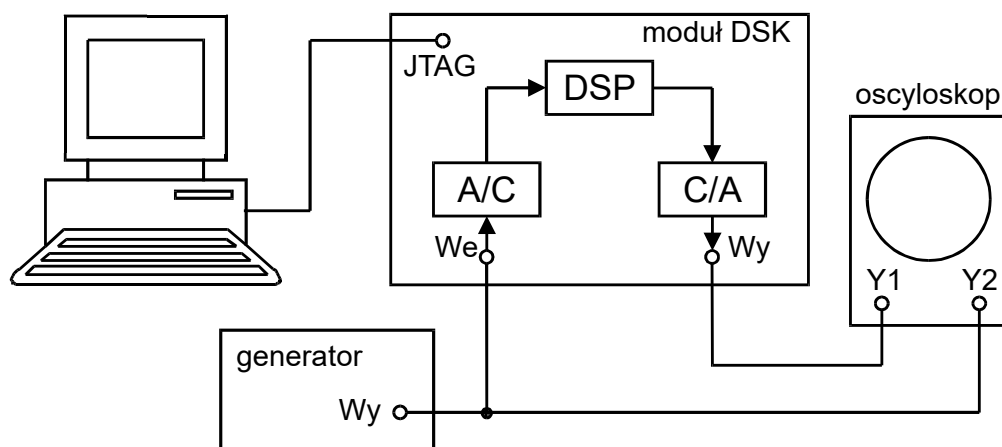
3. Przebieg ćwiczenia

Ćwiczenia laboratoryjne prowadzone są w oparciu o zestaw TMS320C6713 DSP Starter Kit zawierający procesor TMS320C6713 i przetworniki A/C i C/A. W skład stanowiska wchodzi również środowisko programistyczne Code Composer Studio.

W ćwiczeniu należy zrealizować następujące procedury:

- ◆ skonfigurować porty szeregowo McBSP0, McBSP1 do współpracy z układem TLV320AIC23 (układ przetworników A/C i C/A);
- ◆ przeprowadzić konfigurację układu TLV320AIC23;
- ◆ napisać procedury odczytu sygnału z przetwornika A/C oraz zapisu do przetwornika C/A poprzez port szeregowy z wykorzystaniem systemu przerwań. Zapis i odczyt powinny być synchronizowane przerwaniem generowanymi przez odbiornik portu szeregowego.
- ◆ napisać procedurę przepisującą sygnały pochodzące z przetworników A/C do wyjścia C/A. Sygnał w jednym z kanałów należy opóźnić o $10T_d$.
- ◆ napisać procedurę zapisu do bufora sygnału odczytanego z wejścia, a następnie wykorzystując możliwości CCS narysować buforowany przebieg

Badania eksperymentalne należy przeprowadzić w układzie połączonym według poniższego schematu.



Rys. 1. Schemat połączeń stanowiska laboratoryjnego DSP do badania procedury przesyłania sygnałów

Przy testowaniu procedur amplituda napięcia podawanego z generatora nie powinna przekraczać 2V.

4. Sprawozdanie powinno zawierać:

- kody źródłowe wraz z opisem,
- wyniki działania procedur,
- analizę kodów źródłowych stosowanych procedur,
- uwagi i wnioski nasuwające się w trakcie wykonywania ćwiczenia.

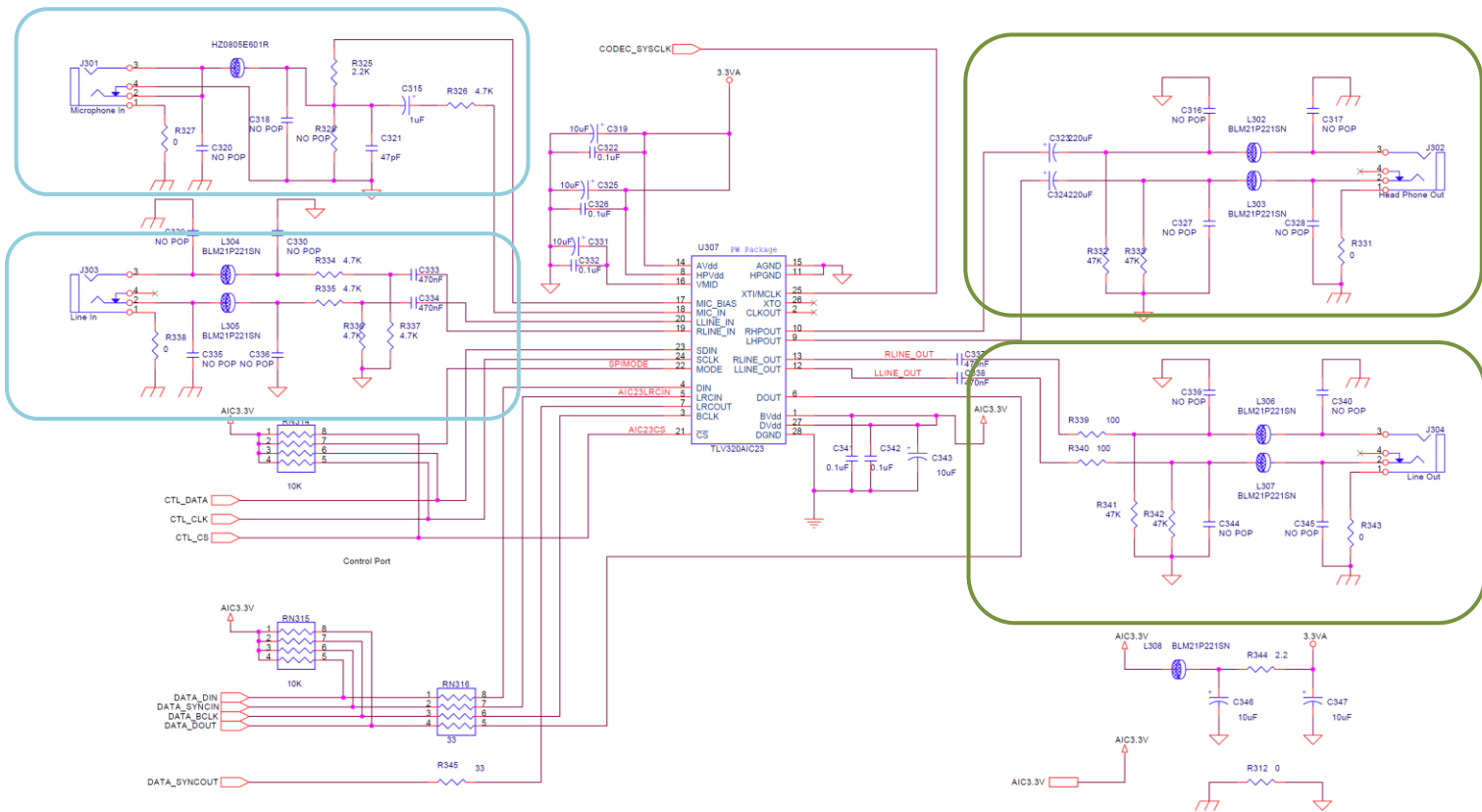
5. Wymagania BHP

W trakcie realizacji programu ćwiczenia należy przestrzegać zasad omówionych we wstępie do ćwiczeń, zawartych w: „Regulaminie porządkowym w laboratorium” oraz w „Instrukcji obsługi urządzeń elektronicznych znajdujących się w laboratorium z uwzględnieniem przepisów BHP”. Regulamin i instrukcja są dostępne w pomieszczeniu laboratoryjnym w widocznym miejscu.

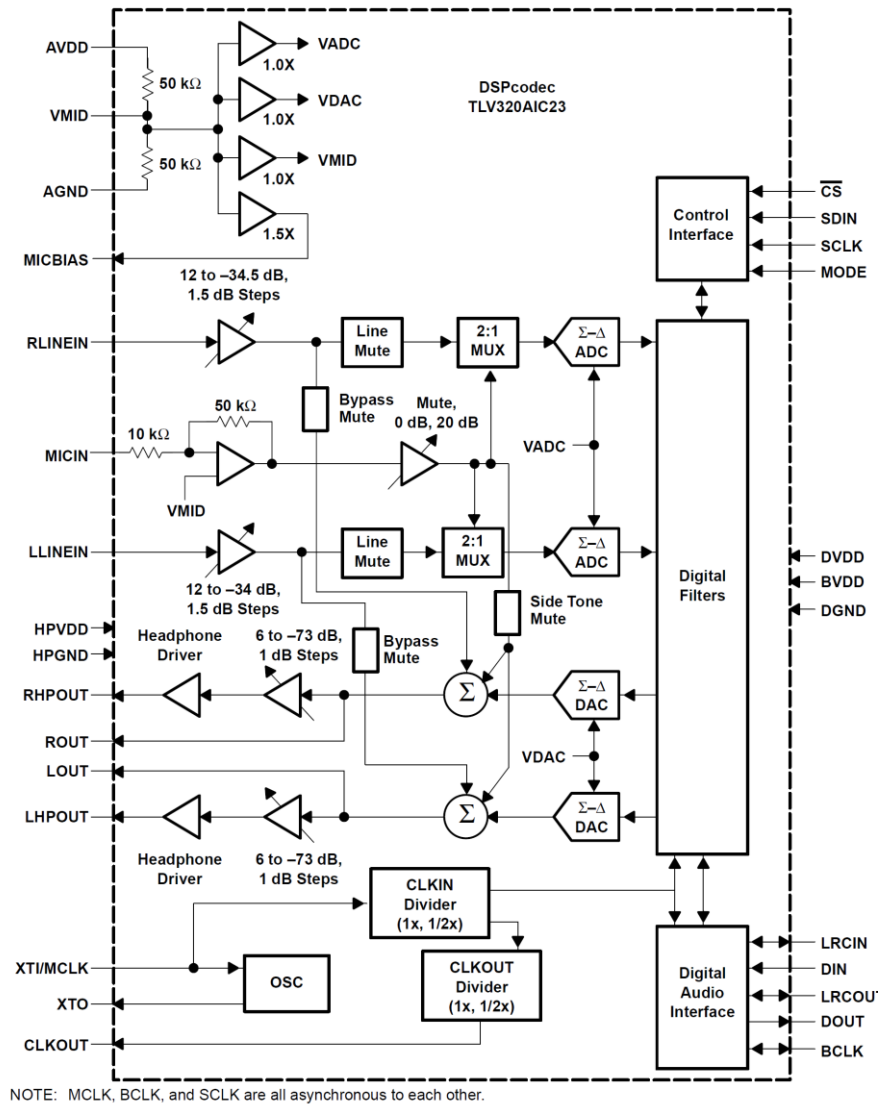
6. Literatura

1. Kowalski H. A., *Procesory DSP dla praktyków*, BTC, Legionowo, 2013
2. Kowalski H. A., *Procesory DSP w przykładach*, BTC, Legionowo, 2012.
3. Dąbrowski A. (red.) *Przetwarzanie Sygnałów Przy użyciu Procesorów Sygnałowych*, Wyd. Politechniki Poznańskiej, Poznań 2000.
4. Zieliński T., *Cyfrowe przetwarzanie sygnałów: od teorii do zastosowań*, WKŁ, Warszawa, 2009.
5. Smith S. W., *Cyfrowe przetwarzanie sygnałów: praktyczny poradnik dla inżynierów i naukowców*, Wydawnictwo BTC, Warszawa, 2007.
6. Texas Instruments, *TMS320C6000 DSP Peripherals Overview*, 2009.
7. Texas Instruments, *TMS320C6000 DSP Multichannel Buffered Serial Port (McBSP) Reference Guide*, 2006.
8. Texas Instruments, *TMS320C6000 Chip Support Library API Reference Guide*, 2004.
9. Texas Instruments, *TLV320AIC23 Stereo Audio CODEC Data Manual*, 2004.
10. Kay A., Green T., (eds.), *Analog Engineer's Pocket Reference*, Texas Instruments, 2015.
11. Texas Instruments, *TMS320C67x DSP CPU and Instruction Set Reference Guide*, 2006.
12. Texas Instruments, *TMS320C6000 Programmer's Guide*, 2006.
13. Kehtarnavaz, N., *Real-Time Digital Signal Processing: Based on the TMS320C6000*, Newnes, 2005.
14. Welch T. B., Wright C.H.G., Morrow M.G., *Real-time Digital Signal Processing from Matlab to C with the TMS320C6x DSPs*, Taylor & Francis, 2012.

Dodatek 1. DSK 6713 - moduł audio



TLC320AIC23 kodek audio (stereo)



Dodatek 3.

Konfiguracja rejestrów McBSP0 (w trybie SPI master)

SPCR0	0x00C31000
PCR	0x0A0A
SRGR0	0x20001363
XCR0	0x10040
RCR0	0 (tylko nadawanie)

Konfiguracja rejestrów McBSP1

SPCR1	0x00010001
PCR1	0x00000003
SRGR1	0x20000001
XCR1	0x000000A0
RCR1	0x000000A0

Rejestry konfiguracyjne AIC23

ADDRESS	REGISTER
0000000	Left line input channel volume control
0000001	Right line input channel volume control
0000010	Left channel headphone volume control
0000011	Right channel headphone volume control
0000100	Analog audio path control
0000101	Digital audio path control
0000110	Power down control
0000111	Digital audio interface format
0001000	Sample rate control
0001001	Digital interface activation
0001111	Reset register

Przykładowa konfiguracja AIC23

Adres	Wartość	Uwagi
0x00	0x001c	wzmocnienie wej_L = 0dB
0x01	0x001c	wzmocnienie wej_P = 0dB
0x02	0x01f9,	
0x03	0x01f9,	
0x04	0x811	wej. - "Line"
0x05	0x000	
0x06	0x000	
0x07	0x0043	
0x08	0x081	fd=48 kHz
0x09	0x001	

Rozdzielenie i złożenie kanałów L/P <-> A/C, C/A w AIC23

32b - int:

kanał lewy - 16b - short	kanał prawy - 16b - short
--------------------------	---------------------------

```
#define PRAWY 0
#define LEWY 1
short probka_L;
short probka_R;
union {
    int int_32;
    short kanal_16[2];
} probka_2x16;
// .....
    probka_2x16 = odczyt_z_AC(); //odczyt probki z A/C (2x16b)
    //rozdzielenie kanałów
    probka_L = probka_2x16.kanal_16[LEWY];
    probka_R = probka_2x16.kanal_16[PRAWY];
// .....
    //złożenie kanałów
    probka_2x16.kanal_16[LEWY] = probka_L;
    probka_2x16.kanal_16[PRAWY] = probka_R;

    zapis_do_CA(probka_2x16); //zapis probki do C/A (2x16)
```