

Nauczanie zagadnień cyfrowego przetwarzania sygnałów z zastosowaniem modułów z mikrokontrolerem

Streszczenie. W artykule przedstawiono wybrane aspekty nauczania zagadnień cyfrowego przetwarzania sygnałów (DSP) z zastosowaniem modułów z mikrokontrolerem, które umożliwiają ilustrację działania algorytmów DSP w czasie rzeczywistym. Omówiono systemy wykorzystywane podczas zajęć laboratoryjnych dla studentów kierunku Automatyka i Robotyka pierwszego i drugiego stopnia. Popularny moduł Arduino Uno z mikrokontrolerem ATmega328P uzupełniono o przetwornik cyfrowo-analogowy, tak aby zbudować pełen system przetwarzania sygnałów. Przygotowano instrukcje laboratoryjne oraz oprogramowanie, które prezentują podstawowe operacje cyfrowej generacji sygnałów, filtracji oraz obliczeń dyskretnej transformaty Fouriera za pomocą technik bezpośrednich, FFT i za pomocą algorytmu Goertzela.

Abstract. The paper presents selected aspects of teaching digital signal processing (DSP) with the use of modules with a microcontroller, which allow to illustrate the operation of DSP algorithms in the real time. The popular Arduino Uno module with the ATmega328P microcontroller has been supplemented with an analog-digital converter to build a full signal processing system. Laboratory instructions and software were prepared to present basic operations of digital signal generation, filtering and discrete Fourier transformation calculations using direct techniques, FFT, and the Goertzel algorithm. (**Teaching digital signal processing with the use of modules with a microcontroller**)

Słowa kluczowe: cyfrowe przetwarzanie sygnałów, filtry cyfrowe, Arduino

Keywords: digital signal processing, digital filters, Arduino

Wprowadzenie

Nauczanie zagadnień cyfrowego przetwarzania sygnałów (ang. DSP - digital signal processing), oprócz omówienia podstaw teoretycznych, powinno być ilustrowane układami elektronicznymi czasu rzeczywistego [1, 2, 3], np. za pomocą modułów z procesorami sygnałowymi.

Przykładem modułu edukacyjnego DSP jest TMS320C5515 eZDSP USB Stick (pokazany na rys. 1) [4], który wyposażono także w zintegrowany koder dźwięku TLV320AIC3204, komunikujący się z mikroprocesorem z wykorzystaniem interfejsu szeregowego I2S (*Inter-IC-Sound*). Układ pracuje z szybkością od 8 do 192 kpróbk/s, przesyłając dane mono lub stereo o rozdzielczości 16 bitów. Programowanie modułu odbywa się z wykorzystaniem środowiska Code Composer Studio (CCS), które w przypadku podłączenia modułu akceptowanych przez firmę Texas Instruments oferuje pełną funkcjonalność. Należy dodać, że dla modułu opracowano zestaw materiałów edukacyjnych zawierających prezentacje i przykładowe oprogramowanie obejmujące algorytmy cyfrowego przetwarzania sygnałów audio, między innymi: echo cyfrowe, efekty gitarowe, odszumianie sygnałów, kompresję mowy standardów G.711, generację i detekcję sygnałów DTMF [5].



Rys. 1. TMS320C5515 eZDSP USB Stick

Oprócz najbardziej znanych producentów procesorów sygnałowych (takich jak Texas Instruments, Analog Devices, Freescale, Microchip), ofertę edukacyjną w zakresie cyfrowego przetwarzania sygnałów w czasie

rzeczywistym oferują także producenci znani z rynku mikrokontrolerów. W laboratorium DSP prowadzonym dla studentów 2 stopnia, autorzy artykułu wykorzystują m.in. moduł STM32F407 Discovery zawierający procesor STM32F407 ARM Cortex-M4 [6], pracujący z zegarem 168 MHz. Moduł komunikuje się z hostem PC poprzez interfejs USB, wykorzystując narzędzie ST-LINK do programowania i debugowania. Środowisko Keil MDK-ARM pozwala (na hoście PC) na kompilację oprogramowania napisanego w C/C++, linkowanie i wgranie do pamięci STM32F407. Wejścia/wyjścia (I/O) audio czasu rzeczywistego są udostępnione poprzez kartę Wolfson Pi Audio (zaprojektowaną dla modułów Raspberry Pi), która komunikuje się przewodowo przy pomocy interfejsów I2C oraz I2S. W ramach współpracy uniwersyteckiej można uzyskać roczną licencję (z możliwością przedłużenia) na pełną wersję środowiska ARM Keil® MDK-Professional development tool from ARM oraz zestaw materiałów dydaktycznych, obejmujących typowe zagadnienia DSP (filtry FIR, IIR, LMS oraz realizację FFT).



Rys. 2. STM32F407 Discovery z kartą Wolfson Audio

Z punktu widzenia procesu kształcenia istotne jest, aby w eksperymentach laboratoryjnych koncentrować się na elementach algorytmów DSP a nie na niuansach związanych z konfiguracją (niejednokrotnie żmudną) rozbudowanych środowisk programistycznych czy ze skomplikowaną specyfiką programowania układów wejścia/wyjścia.

Istotnym celem elementarnego kształcenia DSP jest także pokazanie implementacji algorytmów bez wykorzystania gotowych bibliotek, które oczywiście są zoptymalizowane i skuteczne w działaniu, ale nie pozwalają na zapoznanie się z mechanizmami funkcjonowania algorytmów przetwarzania sygnałów. Należy też zauważyć, że ze względu na coraz bardziej skomplikowaną architekturę procesorów, współcześnie w większości przypadków, nie stosuje się już programowania w asemblerze, co było typowe w latach 80-tych i 90-tych XX wieku.

Moduł Arduino UNO jako system DSP

Platforma programistyczna Arduino jest otwartym projektem zapoczątkowanym w 2005 roku [7]. Prostota programowania z zastosowaniem języka C/C++, niskie koszty modułów wykorzystujące mikrokontrolery AVR firmy Atmel spowodowały ogromną popularność tych rozwiązań. Moduł Arduino Uno wykorzystuje mikrokontroler ATmega328P [8], który nie jest typowym procesorem sygnałowym (nie potrafi wykonywać jednoczesnego mnożenia i dodawania MAC - Multiply and Accumulate), ale posiada architekturę harwardzką i sprzętowy układ mnożący, który potrafi dokonywać operacji mnożenia liczb całkowitych [9]:

- MUL - Multiply Unsigned
- MULS - Multiply Signed
- MULSU Multiply Signed with Unsigned

a także stałoprzecinkowych liczb ułamkowych:

- FMUL – Fractional Multiply Unsigned
- FMULS – Fractional Multiply Signed
- FMULSU – Fractional Multiply Signed with Unsigned.

Instrukcje MUL i MULS wymagają 1 cyklu zegarowego, a pozostałe instrukcje – już 2 cykle zegarowych. W przypadku mnożenia ułamkowego czynniki mnożenia są reprezentowane w formacie 1.7, a wynik mnożenia jest otrzymywany w formacie 1.15 (reprezentacja na dwóch bajtach z automatycznym przesunięciem o jeden bit w lewo).

Z punktu widzenia realizacji systemu DSP istotna jest możliwość próbkowania i rekonstrukcji sygnałów analogowych. Mikrokontroler ATmega328P (a tym samym Arduino Uno) posiada wbudowany przetwornik analogowo-cyfrowy pracujący z rozdzielczością 10-bitów przy maksymalnej szybkości akwizycji 15 kpróbek/s. Przetwornik nie posiada filtra antyaliasingowego i dzięki temu można podczas eksperymentów laboratoryjnych zaprezentować problem stosowania zbyt małej szybkości próbkowania.

Mikrokontroler ATmega328P nie jest wyposażony w przetwornik cyfrowo-analogowy (C/A), więc należy go dołączyć samodzielnie. Autorzy artykułu, podczas zajęć dydaktycznych, wykorzystują dwa układy scalone:

- TLC5615 - 10 bitowy przetwornik C/A z komunikacją szeregową SPI [10]
- MCP4725 - 12-bitowy przetwornik C/A z komunikacją I2C [11].

W celu ułatwienia korzystania z modułu i dołączonego przetwornika C/A, zaprojektowano płytkę (tzw. shield) zawierający przetwornik TLC5615 oraz elementy pomocnicze jak wzmacniacz LM386, gniazda BNC i Cinch ułatwiające podłączenie generatora sygnałów i oscyloskopu [12] oraz złącza dla elementów filtru RC (rys. 3).

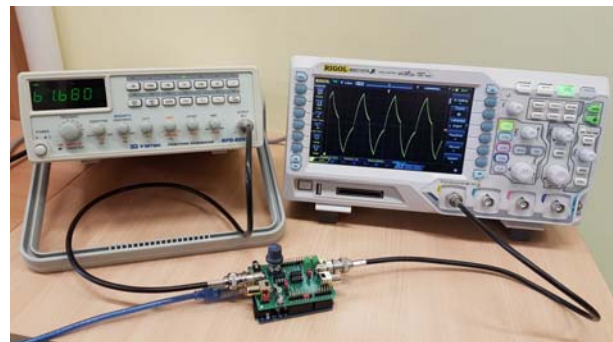
Alternatywnym rozwiązaniem dołączania zewnętrznego przetwornika C/A jest użycie modułu Arduino Due, który zawiera układ konwersji cyfrowo-analogowej [7]. Jednak moduł ten jest zdecydowanie droższy i nie jest tak popularny jak Arduino Uno.



Rys. 3. DSP Arduino Shield

Ćwiczenia DSP z wykorzystaniem modułu Arduino UNO

Podczas zajęć laboratoryjnych do systemu z modułem Arduino UNO i z podłączonym przetwornikiem C/A, dołącza się generator sygnałowy oraz oscyloskop (rys. 4). Do programowania modułu Arduino wykorzystuje się komputer PC.

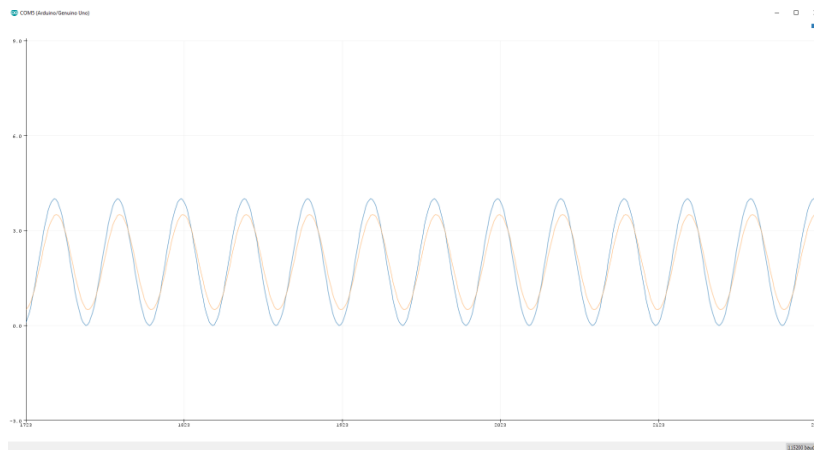


Rys. 4. System edukacyjny DSP z wykorzystaniem Arduino

Alternatywnym rozwiązaniem jest wykorzystanie karty dźwiękowej jako źródła sygnału oraz tzw. „Kreślarki” („Serial Plot”), umożliwiającej graficzną obserwację (w czasie rzeczywistym) wartości wysyłanych za pomocą portu szeregowego (rys. 5).

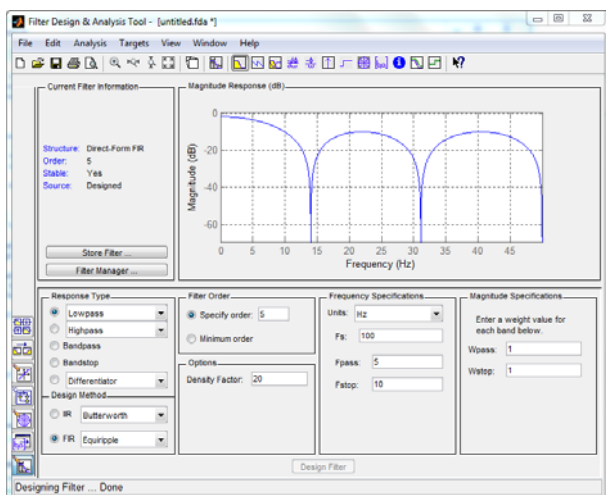
Dla tak przygotowanego systemu, opracowano zestaw ćwiczeń laboratoryjnych (instrukcje ćwiczeń laboratoryjnych i oprogramowanie sterujące przebiegiem ćwiczeń) obejmujący zagadnienia:

- akwizycji sygnałów (próbkowania i kwantyzacji) oraz generacji sygnałów sinusoidalnych z wykorzystaniem tablic próbek, szeregów Taylora oraz filtrów IIR
- realizacji filtrów o skończonej odpowiedzi impulsowej FIR (finite impulse response)
- realizacji filtrów o nieskończonej odpowiedzi impulsowej IIR (infinite impulse response)
- analizy widmowej DFT z użyciem obliczeń bezpośrednich oraz algorytmów FFT (fast Fourier transformation)
- detekcji telefonicznych tonowych sygnałów sygnalizacyjnych (DTMF) za pomocą algorytmu Goertzela
- rekonstrukcji sygnału analogowego z wykorzystaniem technik PWM.



Rys. 5. Przykład użycia narzędzia "Kreślarka" do porównywania przebiegów sygnału wejściowego i wyjściowego

Projektowanie filtrów FIR oraz IIR jest dokonywane za pomocą narzędzia *Filter Design & Analysis Tool* (rys. 6) w środowisku Matlab, przy czym na zajęciach dodatkowo demonstrowane jest „ręczne” wyznaczanie współczynników filtru przy użyciu metody okienkowej.



Rys. 6. Okno programu *Filter Design & Analysis Tool*

Z punktu widzenia przetwarzania w czasie rzeczywistym istotna jest analiza czasochłonności obliczeń. Pomocnym rozwiązaniem w przypadku Arduino jest wykorzystanie polecenia *micros()*, które umożliwia pomiar czasu z dokładnością do 4 μ s. Dzięki temu można oceniać efektywność zastosowanych rozwiązań oraz możliwości pracy algorytmów w czasie rzeczywistym.

Podsumowanie

W artykule przedstawiono możliwości wspomagania procesu dydaktycznego związanego z zagadnieniami cyfrowego przetwarzania sygnałów. Użycie modułów z typowymi procesorami sygnałowymi przykładowo z rodziny TMS czy ARM warto stosować dla studentów drugiego stopnia.

Pomysł wdrożenia modułu Arduino w proces dydaktyczny DSP jest podejmowany w różnych ośrodkach edukacyjnych [13] i opisany w podręcznikach [14]. Zastosowanie systemu z modułem Arduino Uno uzupełnionego o przetwornik C/A jest rozwiązaniem, które wydaje się wartościowe dla studentów pierwszego stopnia. Z uwagi na popularność i stosunkowo nieduży koszt takich modułów, eksperymenty związane z cyfrowym

przetwarzaniem sygnałów studenci mogą także prowadzić samodzielnie w warunkach domowych.

Praca powstała w wyniku realizacji działalności statutowej DS-2018.

Autorzy: dr inż. Tomasz Marciniak, Wydział Informatyki, Instytut Automatyki i Robotyki, Zakład Układów Elektronicznych i Przetwarzania Sygnałów, Politechnika Poznańska, ul. Piotrowo 3a, 60-965 Poznań, prof. dr hab. inż. Adam Dąbrowski Wydział Informatyki, Instytut Automatyki i Robotyki, Zakład Układów Elektronicznych i Przetwarzania Sygnałów, Politechnika Poznańska, ul. Piotrowo 3a, 60-965 Poznań

LITERATURA

- [1] C. Wicks, "Lessons learned: teaching real-time signal processing [DSP Education]", IEEE Signal Processing Magazine, vol. 26, no. 6, pp. 181-185, November 2009.
- [2] A. Dąbrowski, P. Figlak, R. Gołębiowski, T. Marciniak, "Przetwarzanie sygnałów przy użyciu procesorów sygnałowych", Wydawnictwo Politechniki Poznańskiej, 1998.
- [3] F. A. Mujica et al., "Teaching digital signal processing with Stanford's Lab-in-a-Box," 2015 IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE), Salt Lake City, UT, pp. 307-312, 2015.
- [4] "TMS320C5515 eZDSP USB Stick Technical Reference", 512845-0001 Rev A II, Spectrum Digital, 2010.
- [5] http://processors.wiki.ti.com/index.php/Porting_C5000_Teaching_ROM_to_C5535_eZdsp
- [6] "Digital Signal Processing (DSP) Education Kit", <https://developer.arm.com/academia/arm-university-program/for-educators/digital-signal-processing/course-details>
- [7] <https://en.wikipedia.org/wiki/Arduino>
- [8] "ATmega328/P AVR Microcontroller with picoPower Technology", Microchip Technology Inc., 2018, <http://www.microchip.com/wwwproducts/en/ATmega328p>
- [9] "Atmel AVR Microcontrollers Instruction Set Manual", Atmel, 11/2016.
- [10] "TLC5615", Texas Instruments, <http://www.ti.com/lit/ds/symlink/tlc5615.pdf>
- [11] "MCP4725 12-Bit Digital-to-Analog Converter with EEPROM Memory", Microchip, 2007.
- [12] P. Rajek, M. Wrąbel, "Edukacyjny system DSP z wykorzystaniem modułu Arduino", praca dyplomowa inżynierska, promotor: T. Marciniak, Politechnika Poznańska, 2016.
- [13] C. Hochgraf, "Using Arduino To Teach Digital Signal Processing", Proceedings of ASEE Northeast Section Conference 2013, <http://asee-ne.org/conferences/aseene/2013/index.php/aseene/aseene2013/paper/view/235>
- [14] S. Monk, "Arduino dla początkujących – Kolejny krok, Rozdział 13 Cyfrowe przetwarzanie sygnałów", Wydawnictwo Helion, 2015