

AudioDSP zestaw z procesorem Sigma DSP ADAU1701 (2)

Środowisko SigmaStudio

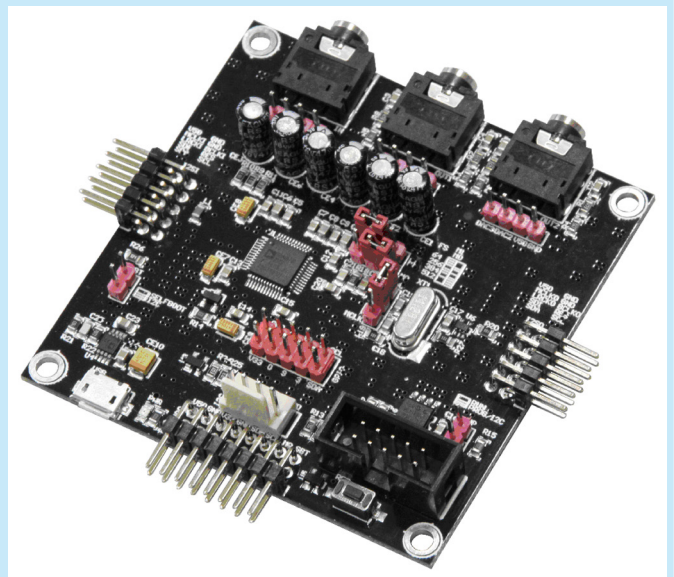
W pierwszej części kursu, w rubryce „projekty” został przeze mnie opublikowany opis zestawu uruchomionego AudioDSP z procesorem ADAU1701. W drugiej części zostanie opisana instalacja i podstawy użytkowania środowiska SigmaStudio, za którego pomocą ten zestaw jest konfigurowany i programowany.

Aby poprawnie wykonywać czynności opisane w artykule, niezbędne jest wyposażenie się w płytkę bazową zestawu (**fotografia 1**), programator USBi, komputer PC z systemem Windows, kartą dźwiękową i głośnikami aktywnymi PC.

SigmaStudio (**rysunek 2**) jest kompletnym środowiskiem graficznym wspomagającym programowanie wszystkich procesorów z rodziny Sigma DSP oraz niektórych układów audio z oferty Analog Devices. Jego wyróżnikiem jest odmienne podejście do tworzenia aplikacji – zamiast języka programowania mamy graficzny interfejs, zbliżony do stosowanego, na przykład, przy programowaniu sterowników PLC (język funkcjonalnych schematów blokowych FBD). Po konfiguracji sprzętowej procesora tworzymy aplikację poprzez łączenie odpowiednio dobranych i wstępnie skonfigurowanych bloków funkcjonalnych. Bloki są dostępne w bibliotekach programu. Łącząc je, tworzymy schemat procesu obróbki sygnału. Ułatwia to rozpoczęcie pracy bez lub z niewielkim doświadczeniem w zakresie programowania procesorów DSP, z minimalną znajomością algorytmów i budowy urządzeń do cyfrowej obróbki sygnału.

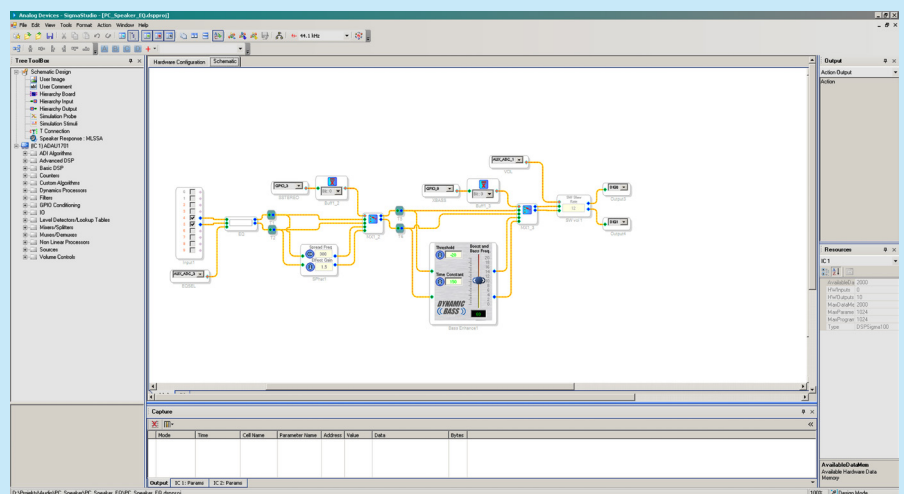
W związku z dostępnością sporej liczby przygotowanych bloków, realizujących zróżnicowane algorytmy obróbki sygnału, możliwe jest tworzenie skomplikowanych aplikacji audio we względnie krótkim czasie. Podejście to zdecydowanie ułatwia pracę, szczególnie osobom, które do tej pory tworzyły systemy audio z fizycznych urządzeń rack 19” lub modułów funkcjonalnych z serii 500, jak to ma miejsce podczas pracy w studio nagraniowym lub na estradzie, gdzie konfigurujemy przepływ sygnału oraz konieczne jego korekcje i modyfikacje wymagane dla uzyskania oczekiwanego efektu końcowego.

SigmaStudio umożliwia tworzenie aplikacji w naturalny dla nich sposób i zastępuje urządzenia elektroniczne odpowiednikami w postaci bloków funkcjonalnych. Nie do przecenienia jest też brak konieczności nauki asemblera DSP, co powoduje, że pierwsza aplikacja może powstać w kilka minut, a osiągnięte efekty i zmiany parametrów ocenione w czasie rzeczywistym. Łatwiejsza jest też analiza projektów przygotowanych przez inne osoby, nie ma konieczności śledzenia tysięcy wierszy programu – wystarczy przeglądnąć schemat funkcjonalny i zapoznać się z konfiguracją użytych bloków i procesora.



Fotografia 1. Płytkę bazową AudioDSP ADAU1701A

Oprogramowanie SigmaStudio jest udostępnione za darmo także w celach komercyjnych. Można je pobrać spod adresu <http://bit.ly/2HFa1AV>. Aktualnie jest dostępna wersja 4.2 (x64).



Rysunek 2. Oprogramowanie SigmaStudio

Oprogramowanie nie ma wygórowanych wymagań sprzętowych. Do pracy jest niezbędny komputer PC z systemem Windows 7/10 (x64), z zainstalowanym Microsoft.Net Framework 4.7, 256 MB pamięci RAM, 80 MB wolnego miejsca na dysku i jednym wolnym portem USB2.0 dla programatora USBi. Zalecany jest monitor o rozdzielczości co najmniej 1024×768 pikseli.

Instalacja z pliku **ADI_SigmaStudio-Rel4.2-x64.exe** przebiega typowo. Po zainstalowaniu SigmaStudio należy dołączyć do komputera programator USBi (poprzez kabel Mini USB), jeszcze bez zestawu AudioDSP. W Menedżerze Urządzeń w zakładce **Kontrolery uniwersalnej magistrali szeregowej** powinna zostać wyświetlona pozycja **Analog Devices USBi**, co potwierdza prawidłową instalację driverów programatora (**rysunek 3**).

Po potwierdzeniu poprawnej instalacji należy uruchomić Sigma Studio. W przedstawionych w kursie projektach wykorzystano wersję 4.2 build1 (Rev1764). Co pewien czas warto sprawdzić, czy nie są dostępne aktualizacje oprogramowania, które oprócz rozbudowy o obsługę nowych układów wnoszą poprawki raportowanych błędów, a niekiedy dodatkowe bloki funkcjonalne.

Budowa interfejsu użytkownika SigmaStudio jest standardowa, okno aplikacji podzielone jest na:

- Obszar menu.
- Obszar ikon poleceń.
- Główny obszar projektu (**Program Window**), w którym po otwarciu projektu dostępne są główne zakładki schematu (**Schematic**) i konfiguracji sprzętowej (**Hardware Configuration**) z zakładkami konfiguracji systemu (**Config**), procesora DSP (**ICx 170x/140x Register Control**) i pamięci EEPROM (**ICx – WinE2PromLoader**). Sigma Studio dopuszcza jednoczesną pracę nad kilkoma projektami.
- ToolBoxy z dostępnymi układami DSP i audio lub elementami bibliotecznymi wyświetlane w zależności od aktywnej zakładki w oknie głównym projektu.
- Okienka statusu wyświetlające informacje o zasobach systemowych DSP i pamięci EEPROM (**Resources**), umożliwiające podgląd zawartości plików generowanych przez kompilator i linker (**Output**) oraz informacje o komunikacji programatora z DSP umożliwiające podgląd przesyłanych danych, parametrów wraz z możliwością ich edycji (**Capture Window**).

Proces tworzenia aplikacji dla DSP składa się z czterech kroków.

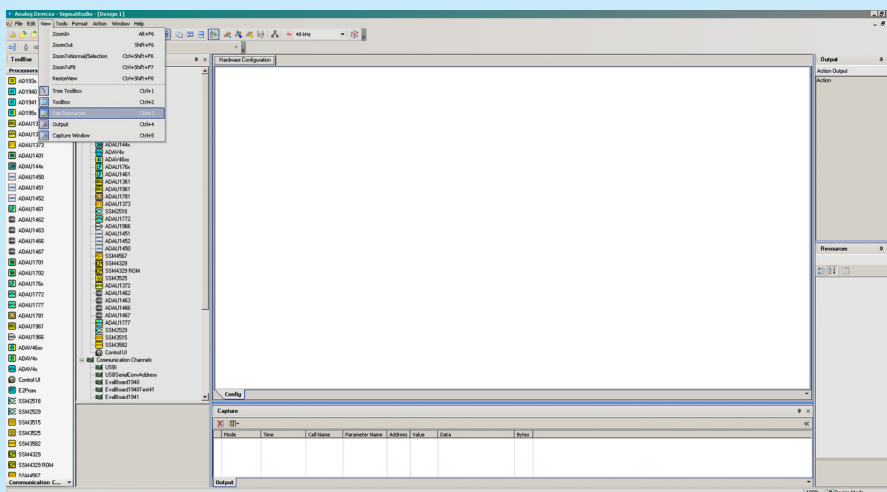
Pierwszym krokiem jest otwarcie nowego projektu z menu **File\New Project (CTRL+N)**. Wszystkie niezbędne pliki podobnie jak w innych IDE zgrupowane są w projekt, zapisywany poleceniem **File\Save (CTRL+S)**, w pliku **Nazwa_Projektu.dspproj**, najlepiej w nowo utworzonym katalogu (w tym wypadku **Projekt_1**). Z menu **View** mamy możliwość skonfigurowania dostępnych na ekranie zakładek narzędziowych oraz ustalenia powiększenia obszaru roboczego (**rysunek 4**). W nowym projekcie przed konfiguracją systemu DSP jest widoczny czysty główny obszar roboczy, który w zależności od podjętych działań będzie uzupełniany o kolejne zakładki zawierające konfigurację sprzętową, konfigurację pamięci oraz właściwy hierarchiczny schemat aplikacji. Do rozpoczęcia pracy niezbędna jest widoczność zakładki **Tree Toolbox**, zawierającej w zależności od wykonywanej czynności biblioteki obsługujących układów i dostępnych programatorów lub biblioteki dostępnych algorytmów.

Drugim krokiem po założeniu nowego projektu jest konfiguracja systemu DSP. W pierwszej kolejności należy ustalić częstotliwość

próbkowania, z którą pracuje DSP. Dla zestawu AudioDSP z wlotowym kwarcem 11,2896 MHz i ustawioną krotnością PLL=256 (położenie zwr M0:2-3, M1:1-2, OSC XT:2-3) jest to 44,1 kHz, czyli częstotliwość odpowiadająca próbkowaniu materiału audio na płycie CD. W celu zmiany fs z listy rozwijanej



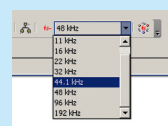
Rysunek 3. Poprawnie zainstalowany USBi



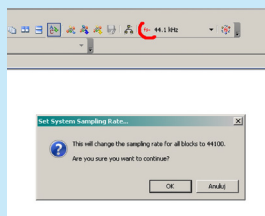
Rysunek 4. SigmaStudio IDE

paska narzędzi należy wybrać 44,1 kHz i ustawić jako częstotliwość próbkowania systemu (**rysunek 5 i 6**). Po zastosowaniu rezonatora 12,288 MHz system może być skonfigurowany do obróbki 48 kHz, czyli materiału audio z DVD. Opisane czynności przy wyborze fs dostępne są też w menu **Action\Set System Sampling rate (CTRL+U) fs-** i **Action\Propagate Sampling Rate (CTRL+Q)**. Zmiana fs możliwa jest także podczas edycji projektu.

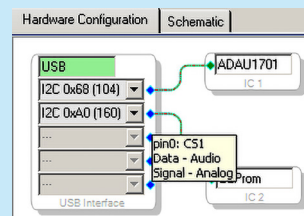
Kolejną czynnością jest ustalenie konfiguracji sprzętowej odpowiadającej płytce AudioDSP. Zakładając pracę „samodzielną” zestawu, w obszar roboczy musimy przeciągnąć z Toolboxa bloczek procesora DSP ADAU1701, bloczek pamięci E2PROM, programator USBi oraz odpowiednio je połączyć. Każdy z bloczków ma „punkty” przyłączeniowe, które w zależności od typu oznaczone są różnymi kolorami (zielony – wejście audio, niebieski – wyjście audio, czerwony – sterowanie oraz GPIO), po ustawieniu kursora myszy nad punktem, otrzymujemy krótką informację o typie sygnału dostępnego na połączeniu (**rysunek 7**). Warto zapoznać się z zawartą informacją, gdyż połączenie nieodpowiednich punktów może powodować błędy lub problemy z działaniem aplikacji. Typy łączonych sygnałów muszą się zgadzać, jest to szczególnie istotne w przypadku wykorzystania GPIO ADC, gdzie bardzo często mylny



Rysunek 5. Ustawienie częstotliwości fs



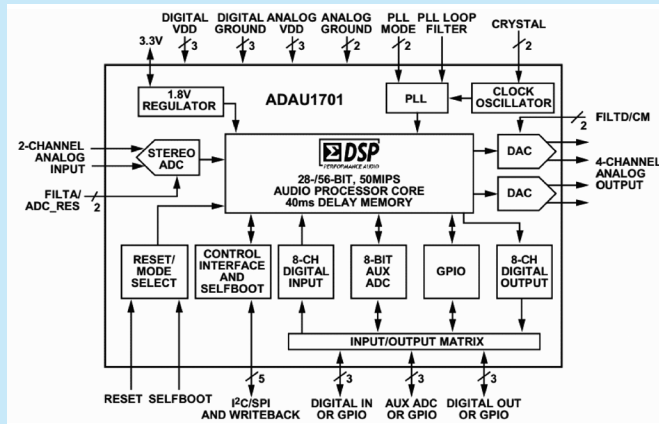
Rysunek 6. Potwierdzenie fs systemu



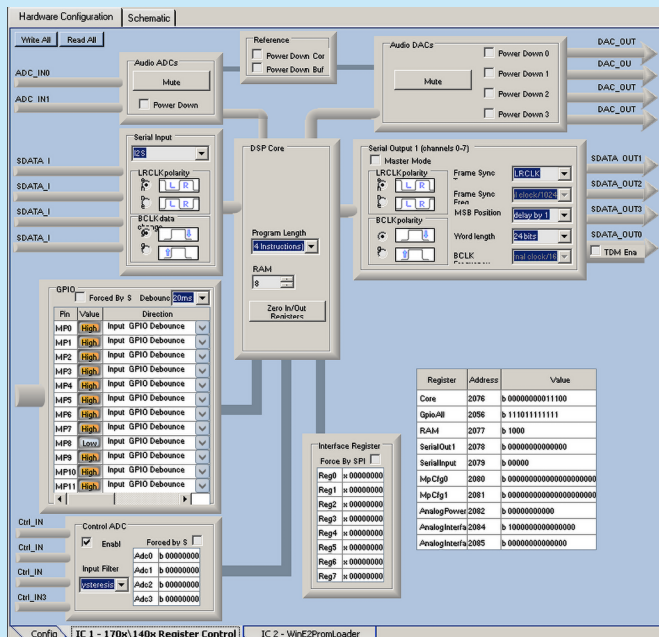
Rysunek 7. Ustalenie konfiguracji sprzętowej AudioDSP

jest sygnał audio z sygnałem analogowym sterującym pochodzącym z przetwornika ADC GPIO. Podobna sytuacja ma miejsce, gdy nie są zachowane typy zmiennych sterujących pracą bloków funkcjonalnych (np.: multiplexerów lub regulatorów głośności), co zostanie opisane w dalszej części kursu przy obsłudze GPIO.

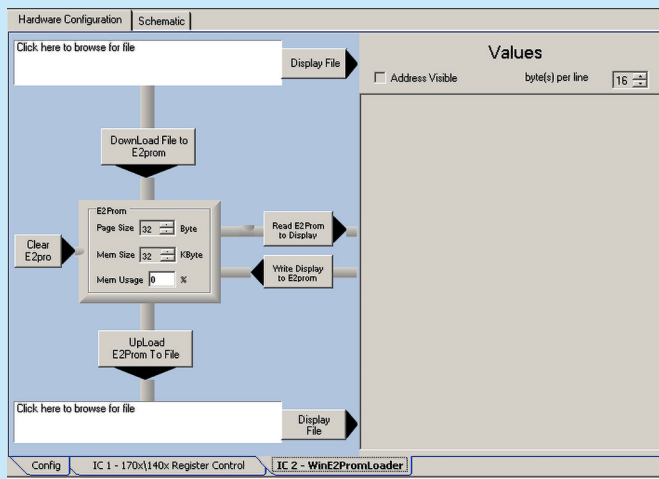
Przy konfigurowaniu USBi należy zwrócić szczególną uwagę na typ magistrali komunikacyjnej i adresy poszczególnych układów.



Rysunek 8. Schemat blokowy ADAU1701A (za notą Analog Devices)



Rysunek 9. Konfiguracja wewnętrzna DSP ADAU1701A

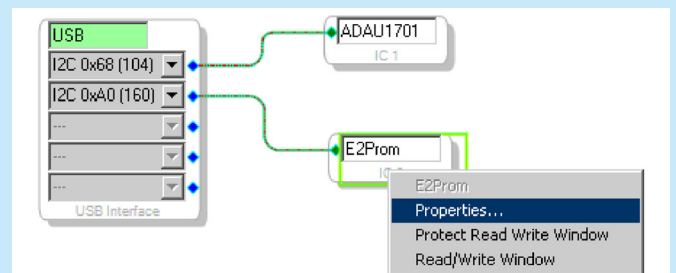


Rysunek 10. Konfiguracja i obsługa pamięci E2PROM

USBi umożliwia programowanie całej rodziny procesorów Sigma DSP przy wykorzystaniu interfejsów I²C i SPI, podobnie jest w wypadku pamięci E2PROM. Procesor ADAU1701 w zestawie Sigma DSP komunikuje się z E2PROM i USBi przez interfejs szeregowy I²C. W związku z możliwością wyboru adresów ADAU1701 i E2PROM należy każdorazowo skontrolować poprawność ich konfiguracji (ADAU1701 – 0x68, E2PROM – 0xA0). Jest to bardzo ważny punkt konfiguracji, ponieważ:

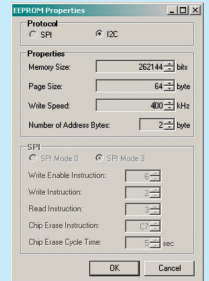
Uwaga: programator USBi ma wbudowaną pamięć E2PROM, której pomyłkowe zaprogramowanie przy błędnym ustawieniu adresu spowoduje jego trwale uszkodzenie. Zawsze przy współpracy z zestawem AudioDSP należy używać domyślnych adresów ADAU1701 – 0x68, E2PROM – 0xA0!

Trzecim krokiem jest odpowiednie skonfigurowanie procesora DSP ADAU1701. Konfiguracja jest oczywiście zależna od realizowanego projektu. Dla przypomnienia warto spojrzeć na budowę wewnętrzną ADAU1701 (rysunek 8) i określić bloki funkcjonalne wykorzystywane w projekcie, które będą wymagały skonfigurowania.

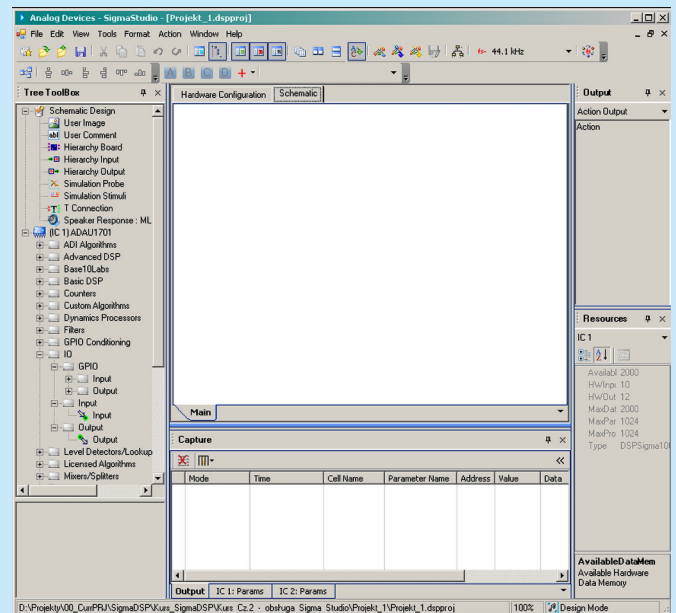


Rysunek 11. Menu właściwości E2Prom

Po skonfigurowaniu systemu w obszarze roboczym zostaną wyświetlone dwie dodatkowe zakładki **IC1-1701x/1401x Register Control** i **IC2 – WinE2PromLoader** odpowiadające konfiguracji elementów składowych AudioDSP. W pierwszej zakładce ustalamy wewnętrzną konfigurację rejestrów ADAU1701 (rysunek 9). Ustalamy cykl programu DSP, wybieramy i konfigurujemy aktywne interfejsy sygnału audio ADC/DAC, konfigurujemy porty szeregowy Serial Input, Output, aktywujemy i konfigurujemy tryb



Rysunek 12. Okienko właściwości E2PROM



Rysunek 13. Edytor schematów Sigma Studio

GPIO oraz współpracujące wejścia GPIO (Control ADC). Szczegółowe opisy konfiguracji zależą od aplikacji DSP i będą przedstawione na przykładach w kolejnych częściach kursu. W drugiej zakładce dostępne są operacje przeznaczone do obsługi pamięci E2PROM (**rysunek 10**). Możliwe jest ustawienie parametrów pamięci, takich jak wielkość i długość bufora zapisu, dostępne są operacje kasowania, programowania i odczytu pamięci wraz z podglądem jej zawartości i możliwością operacji plikowych (zapis/odczyt).

Oprócz zakładki **WinE2PromLoader** konieczna jest jeszcze dodatkowa parametryzacja E2PROM w zakładce **Config**. W tym celu po zaznaczeniu bloku E2Prom klikamy na nim prawym klawiszem myszy w celu wywołania menu podręcznego (**rysunek 11**). Po wybraniu opcji **Properties** uzyskamy dostęp do dodatkowych ustawień konfiguracji pamięci E2PROM (**rysunek 12**).

W tym miejscu możemy skonfigurować oprócz wcześniejszych ustawień dodatkowo prędkość transmisji I²C i długość rejestru adresowego pamięci. Wpisane wartości odpowiadają konfiguracji pamięci AudioDSP. **Uwaga: W razie błędów programowania warto sprawdzić zawartość tej zakładki i zweryfikować poprawność ustawień z kartą katalogową producenta E2PROM.** Po skonfigurowaniu E2PROM należy zamknąć okno i zapisać zmiany w projekcie.

Czwartym i ostatnim krokiem jest wykonanie schematu aplikacji z weryfikacją działania oraz zaprogramowanie DSP. W tym celu przechodzimy do zakładki **Schematic** w obszarze roboczym (**rysunek 13**), po przełączeniu zakładek zmienia się zawartość przybornika toolbox. Zawiera on teraz narzędzia dla tworzenia schematu (**Schematic Design**) oraz biblioteki bloków (**IC1-ADAU1701**) obsługiwanych przez procesor ADAU1701.

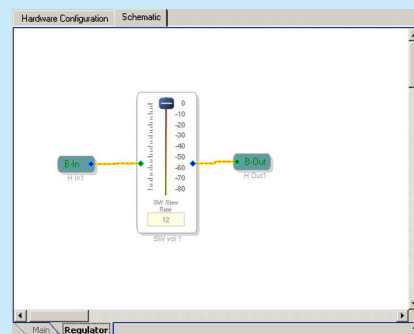
Obsługa edytora jest intuicyjna, na obszar roboczy przeciągamy elementy z biblioteki bloków lub narzędzi do tworzenia schematu, odpowiednio je łącząc (**rysunek 14**). Do schematu można dodawać opisy (**User Comment**), grafiki (**User Image**). Połączenia pomiędzy blokami tworzone są automatycznie, co czasem wymaga korekty przebiegu połączenia. Po wskazaniu pierwszego punktu połączenia i przytrzymaniu lewego przycisku myszy należy przemieścić kursor nad punkt docelowy, połączenie zostanie utworzone (tu jest chyba ułkon twórców oprogramowania w stronę audiofilów, gdyż połączenie do złudzenia przypomina cenione, vintage'owe przewody Western Electric w kolorowym bawełnianym oplocie...). W dowolnym momencie możliwa jest modyfikacja przebiegu połączenia poprzez kliknięcie na przewodzie i odpowiednie przesunięcie podświetlonych uchwytych przewodu. Podobnie przesuwa się bloki funkcjonalne. Aby dodać połączenie (rozgałęzienie) sygnału, należy użyć narzędzia **T Connection**. Należy zwrócić uwagę na użycie odpowiedniego dla procesora DSP bloku **T Connection**, gdyż kompilator nie sygnalizuje błędów, a program nie będzie działał poprawnie np. po wybraniu bloku dla ADAU1772. Podczas rysowania schematu warto zadbać o jego czytelność, oprócz „ręcznego” korygowania przebiegu połączeń dostępnych jest kilka narzędzi edycyjnych, ułatwiających rozmieszczenie i wyrównanie bloków funkcjonalnych dostępnych w menu (**Format/Align**). Wyrównanie odbywa się po zaznaczeniu kilku bloków i wyborze opcji, Center Vertical/Horizontal, Left/Right, Top/Bottom (**CTRL +E/H, +L/R, +T/B**) lub wybór ikony z paska narzędzi

Jak wspominałem, edytor schematów jest hierarchiczny, co ułatwia tworzenie bardziej złożonych aplikacji, wprowadza porządek oraz przyspiesza późniejszą analizę działania układu. Zawsze głównym (nadrzędnym) schematem jest arkusz (zakładka **Main**,

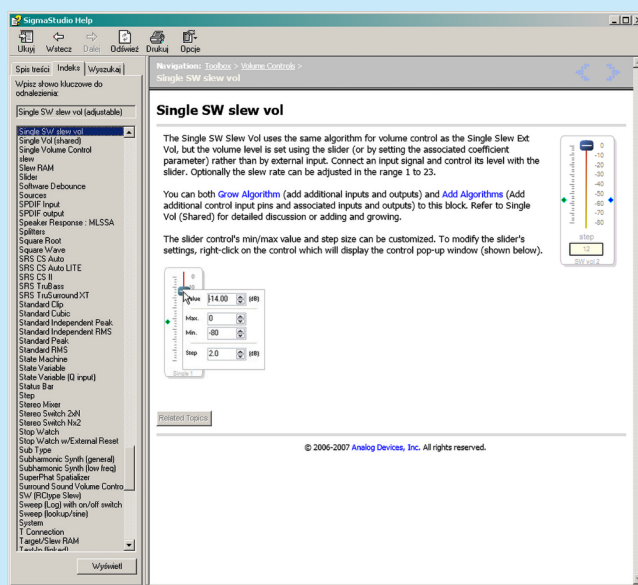
podrzedne schematy tworzone są narzędziem **Hierarchy Board**, zasady tworzenia schematów w blokach są identyczne. Aby umożliwić podłączenie sygnałów do schematów nadrzędnych, należy użyć narzędzi **Hierarchy Input/Output**. Każdy blok hierarchiczny dostępny jest poprzez zakładkę odpowiadającą jego nazwie (**rysunek 15**).

Każdy biblioteczny blok funkcjonalny ma dostępną rozbudowaną pomoc kontekstową (**rysunek 16**), zawierającą opis działania, opis wejść/wyjść i typów sygnałów wymaganych do poprawnego działania. Dostęp do pomocy jest możliwy po zaznaczeniu bloku lewym klawiszem myszy i naciśnięciu klawisza F1. Pomoc jest dostępna także podczas pracy z Sigma Studio, warto z niej skorzystać, gdyż znajduje się w niej wiele cennych i ciekawych wskazówek, a czasami nawet gotowych rozwiązań układowych. Pomocna jest też strona Wiki <http://bit.ly/2U2c5tD> oraz forum DSP: <http://bit.ly/2CCPAQV>.

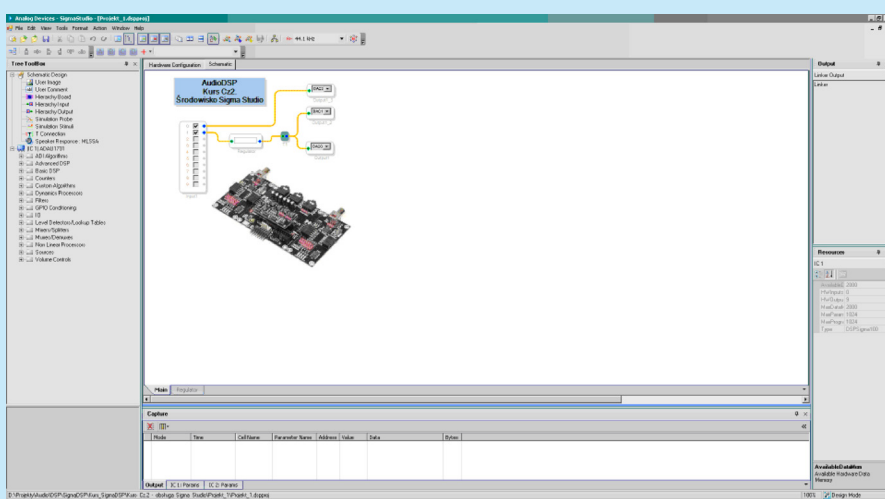
Większość bloków funkcjonalnych wymaga konfiguracji lub parametryzacji. Wywołanie odpowiedniego menu kontekstowego możliwe



Rysunek 14. Tworzenie schematu aplikacji w edytorze Sigma Studio



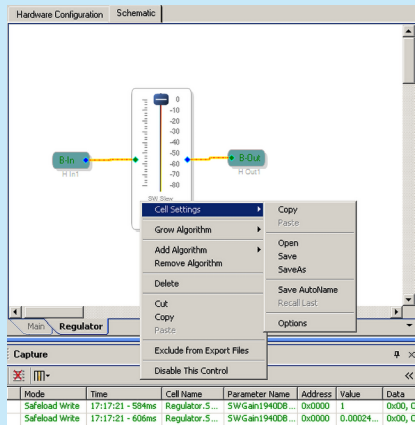
Rysunek 15. Tworzenie bloku hierarchicznego



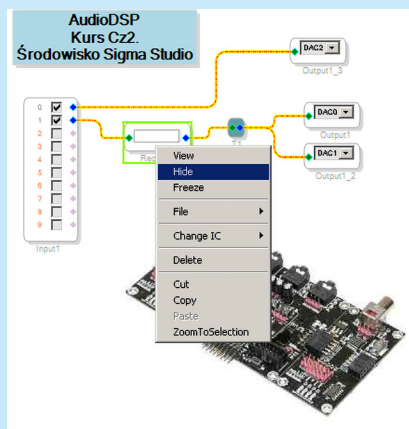
Rysunek 16. Pomoc programu Sigma Studio

jest po naciśnięciu prawego klawisza myszy, gdy kursor jest nad blokiem (rysunek 17).

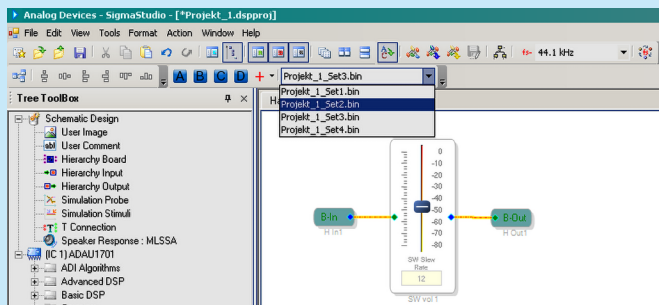
Oprócz narzędzi typowo edycyjnych najważniejszą opcją w menu podręcznym jest opcja **Grow/Reduce Algorithm**, która umożliwia zwiększenie/zmniejszenie liczby obrabianych kanałów w danym bloku, np. przy zmianie typu regulatora głośności z 1 do 5 kanałów. Interesującą opcją jest też **Disable This Control**, ukrywająca nastawy/zawartości bloku i zapewniająca ochronę hasłem. Opcja przydaje się, gdy musimy udostępnić działający algorytm, a nie mamy zamiaru udostępniać sposobu jego realizacji. W podobny sposób po zaznaczeniu bloku w schemacie hierarchicznym



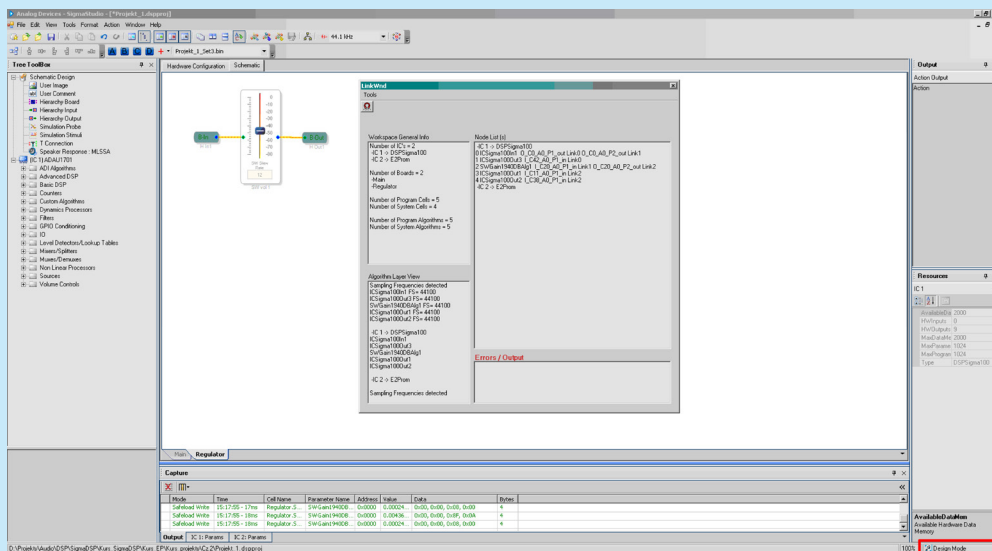
Rysunek 17. Konfiguracja parametrów bloku



Rysunek 18. Opcje ochrony algorytmu



Rysunek 19. Zapis i przywrócenie parametrów



Rysunek 20. Kompilacja projektu

można ukryć lub pokazać zakładkę podrzędną opcją **Hide/Show** albo zablokować hasłem dostęp do edycji schematu podrzędnego opcją **Freeze** (rysunek 18), zabezpieczając możliwość nieautoryzowanych zmian w projekcie.

Podczas uruchamiania i testowania projektu pomocna jest opcja zapisu kilku zestawów parametrów (rysunek 19), które można przywrócić jednym naciśnięciem przycisku ikony A..D. Parametry zapisywane są w plikach w katalogu projektu w podkatalogu **Settings**.

Tak w skrócie wygląda obsługa Sigma Studio. Prawda, że łatwa? Z praktycznych uwag, warto wspomnieć o czasami nieprzewidywalnym zachowaniu się edytora schematów, gdy chcemy kopiować metodą **Ctrl+C/Ctrl+V**, wstawione w ten sposób bloki niekiedy zachowują się jak grafika... Po wklejeniu tracimy dostęp do nastaw, nie polecam więc tej metody, gdy potrzebujemy kilka jednakowych bloków – najlepiej wstawić je po kolei z ToolBoxa, wbrew pozorom zaoszczędzimy czas. Sporadycznie zdarza się też nieprawidłowe działanie opcji powiększenia schematu, po powiększeniu ekran zamraża się i ponownie nie mamy dostępu do parametrów bloków. W tym przypadku pomaga zapisanie projektu i restart SigmaStudio.

W tym miejscu wreszcie można podłączyć programator do zestawu. Do sprawdzenia działania przykładowego projektu wystarczy sama płytka bazowa, bez modułów GPIO i SPDIF. Do wejścia IN podłączamy źródło sygnału, do wyjścia OUT1 głośniczki PC lub wzmacniacz, do gniazda USBi programator. Na płytce programatora powinna zaświecić się czerwona LED sygnalizująca zasilanie USB oraz żółta potwierdzająca detekcję I²C, na płytce AudioDSP powinna zaświecić się zielona LED sygnalizująca obecność zasilania (dostarczonego przez programator USBi).

Po narysowaniu schematu i konfiguracji DSP (po otwarciu projektu Projekt1.dssproj, konfiguracja odbędzie się automatycznie) należy projekt zlinkować, wybierając z menu Action/Link Project lub ikonę albo skrótem (**CTRL+I**). W wyniku kompilacji dostajemy informację o systemie i użytych zasobach no i oczywiście w najlepszym przypadku o braku błędów (rysunek 20). Najczęstsze błędy to niepołączone wejścia bloków na schemacie. Następnie należy projekt skompilować poprzez ikonę Link Compile Connect. Po skompilowaniu warto sprawdzić prawy fragment paska statusu, jest w nim wyświetlana informacja o statusie projektu. Przed podłączeniem USBi Sigma Studio pracuje w trybie edycji projektu **Design Mode**, po bezbłędnym linkowaniu i skompilowaniu status jest zmieniany na skompilowany i gotowy do programowania **Ready: Compiled**.

Po wybraniu opcji menu **Action/Link Compile Download** lub ikony projekt zostaje załadowany do pamięci DSP. Poprawne załadowanie potwierdzone jest zmianą statusu na **Active: Downloaded**, jeżeli

zapomnieliśmy podłączyć programator USBi lub nastąpił błąd podczas programowania, to otrzymamy status **Compiled, Comms Failed**. Możliwy jest też status błędu **Computation Error**, gdy zmienne przekraczają dopuszczalny zakres lub nastąpiła awaria USB, ale z takim w praktyce jeszcze się nie spotkałem.

Zmienne statusy oprócz komunikatu tekstowego zmieniają także kolor belki pod schematem, niebieski oznacza tryb edycji projektu, ciemnozielony poprawną kompilację, jasnozielony poprawne załadowanie projektu do DSP, kolor nie do określenia przez typowego samca gatunku ludzkiego (podobno coś pomiędzy

AVT 5599 Zdalnie sterowany włącznik 4-kanalowy

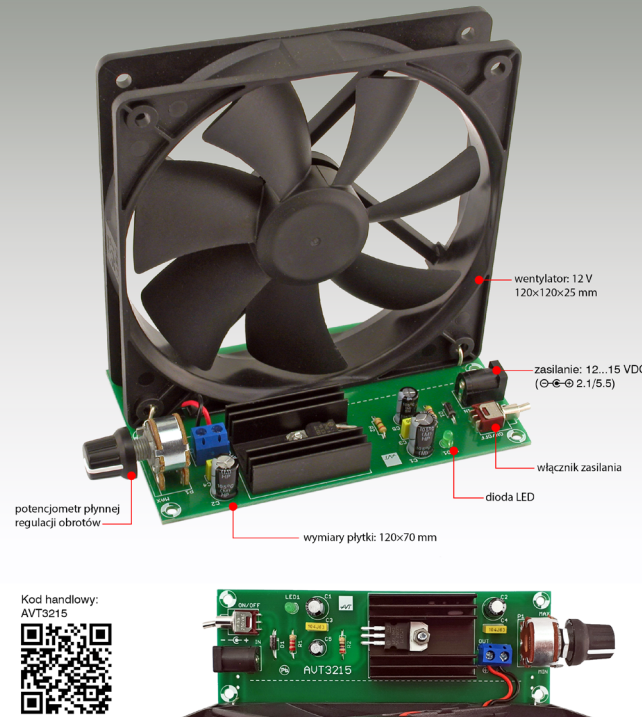
Urządzenie umożliwia zdalne włączanie/wyłączanie czterech urządzeń przy pomocy typowych pilotów na podczerwień do sprzętu powszechnego użytku. Jego niewątpliwym atutem jest możliwość współpracy praktycznie z dowolnym pilotem na podczerwień, a procedura nauki kodów pilota sprowadza się do kilku łatwych czynności.



Kod handlowy: AVT5599
 sklep.avt.pl AVT-Korporacja Sp. z o.o. 03-197 Warszawa, ul. Leszczyńska 11
 Dział Handlowy tel.: 222 578 450 e-mail: handlowy@avt.pl

AVT 3215 Sterownik odciążu oparów

Urządzenie umożliwia płynną regulację obrotów wentylatora stosownie do potrzeb, dzięki czemu możliwe jest znaczne zredukowanie wytwarzanego przezeń hałasu. Rozwiązanie to nie zastąpi dobrego wyciągu, ale w połączeniu z wkładem w postaci filtra węglowego oraz obowiązkowym regularnym przewietrzaniem pomieszczenia pozwoli znacznie zredukować niebezpieczeństwo bezpośredniego wdychania oparów lutowniczych.



Kod handlowy: AVT3215
 sklep.avt.pl AVT-Korporacja Sp. z o.o. 03-197 Warszawa, ul. Leszczyńska 11
 Dział Handlowy tel.: 222 578 450 e-mail: handlowy@avt.pl

Capture

Address	Data Hex	Data 5.23	Data 28.0	Param Name
0	0x00, 0x3D, 0x43, 0xC0	0.88301	4015040	SWGain1940DB ...
1	0x00, 0x00, 0x08, 0x00	0.0002441406	2048	SWGain1940DB ...
2	0x00, 0x00, 0x00, 0x00	0	0	
3	0x00, 0x00, 0x00, 0x00	0	0	
4	0x00, 0x00, 0x00, 0x00	0	0	
5	0x00, 0x00, 0x00, 0x00	0	0	
6	0x00, 0x00, 0x00, 0x00	0	0	
7	0x00, 0x00, 0x00, 0x00	0	0	
8	0x00, 0x00, 0x00, 0x00	0	0	
9	0x00, 0x00, 0x00, 0x00	0	0	
10	0x00, 0x00, 0x00, 0x00	0	0	
11	0x00, 0x00, 0x00, 0x00	0	0	
12	0x00, 0x00, 0x00, 0x00	0	0	
13	0x00, 0x00, 0x00, 0x00	0	0	
14	0x00, 0x00, 0x00, 0x00	0	0	

Output IC 1: Params IC 2: Params

Rysunek 21. Ręczna modyfikacja rejestrów DSP

jasnościowym a cielistym.) błąd komunikacji USBi oraz czerwony Computation Error. Każdorazowo po zmianie edycji schematu, proces kompilacji należy powtórzyć.

Zakładam, że ładowanie odbyło się pomyślnie i najprostsza możliwa aplikacja działa, tj. na wyjściu OUT2 dostępny jest sygnał z wejścia IN bez zmian w DSP, a na wyjściu OUT1 sygnał o regulowanej potencjometrem (w zakładce Regulator) amplitudzie. Warto sprawdzić działanie pamięci nastaw ikonami A.D, poziom powinien zmieniać się w zależności od wybranej ikony. W związku z tym, że program ładowany jest do pamięci ulotnej DSP, po przyciśnięciu przycisku reset lub odłączeniu USBi zostanie utracony i w razie konieczności musi zostać załadowany ponownie. Tryb ten przyspiesza sprawdzenie projektu bez straty czasu na programowanie pamięci EEPROM. Podczas zmian nastaw potencjometru warto przyjrzeć się komunikacji USBi z DSP, w okienku Capture, na bieżąco zmieniają się wartości wysyłane do pamięci/rejestrów procesora, można je modyfikować, wpływając na pracę DSP (rysunek 21). Jest to wykorzystywane, gdy DSP pracuje z procesorem zewnętrznym, który po I²C modyfikuje odpowiednie rejestry, aplikacja jest wgrywana do DSP z pamięci EEPROM po włączeniu zasilania, jest to tzw. tryb Safeload. Wartości modyfikowane mogą zostać zapisane na dysk (poprzez menu podręczne) i użyte przy programowaniu procesora sterującego.

W oczekiwaniu na następny odcinek kursu warto zapoznać się z dokumentacją ADAU1701 oraz Sigma Studio, ze szczególnym zwróceniem uwagi na zawartość bibliotek i sprawność rysowania schematów, gdyż będzie to niezbędne w następnej części kursu, do przygotowania pierwszej aplikacji dla SigmaDSP.

– zapraszam...
Adam Tatuś, EP

REKLAMA

REKLAMA

ELEKTRONIKA PRAKTYCZNA

na facebook

<https://www.facebook.com/ElektronikaPraktyczna>

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie www.media.avt.pl