

AudioDSP zestaw z procesorem Sigma DSP ADAU1701 (3)

Obróbka sygnału analogowego

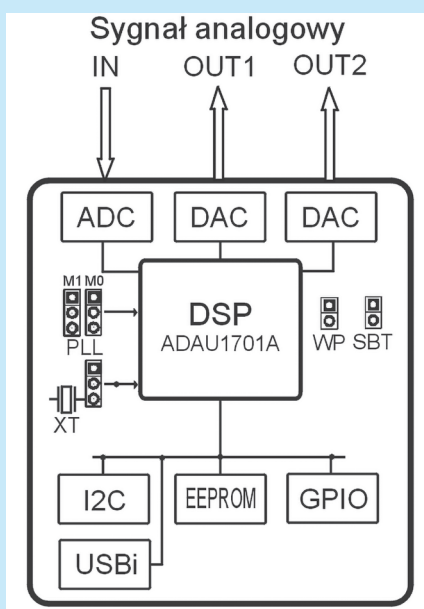
W trzeciej części kursu przedstawione zostaną projekty bazujące na płytce AudioDSP i przeznaczone do obróbki sygnału analogowego. Do przeprowadzenia testów, oprócz zapoznania się z poprzednimi odcinkami kursu, konieczne będzie przygotowanie części sprzętowej, to jest: płytki bazowej AudioDSP, programatora USBi, źródła linowego sygnału analogowego, wzmacniacza z głośnikami oraz okablowania jack stereo 3,5 mm/RCA (w zależności od standardu podłączonych urządzeń), przydatne będą też minimoduły audio i kabel żeński (4-pinowy, jeden do jednego, raster 2,54 mm) do połączenia minimodułów.

Warto na wstępie przypomnieć konfigurację sprzętową płytki AudioDSP, która będzie miała zastosowanie w przykładowych aplikacjach – pokazano ją na **rysunku 1**.

Konfiguracja

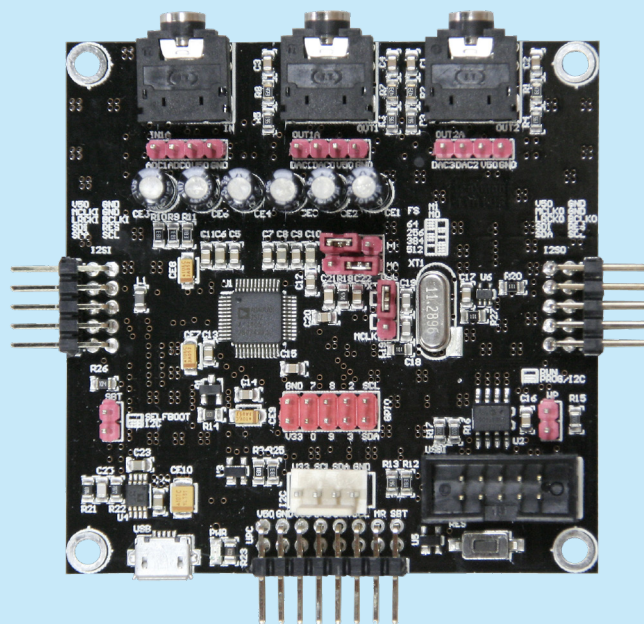
Zestaw AudioDSP (**rysunek 2**) ma zintegrowany 2-kanalowy przetwornik audio A/C o rozdzielczości 24 bitów i maksymalnej prędkości próbkowania 96 kSa/s. Wejściowy sygnał analogowy audio jest doprowadzony do gniazda stereofonicznego jack 3,5 mm oznaczonego IN. Wyjściowy sygnał analogowy audio ze zintegrowanego 4-kanalowego przetwornika C/A jest dostępny w gniazdach stereofonicznych jack 3,5 mm oznaczonych OUT1 i OUT2. Kanały DAC0/DAC1 są doprowadzone do gniazda OUT1. Kanały DAC2/DAC3 do gniazda OUT2. Dla ułatwienia dołączenia modułów rozszerzeń sygnały audio są powielone na złączach szpilkowych, których wyprowadzenia są kompatybilne z minimodułami audio.

Przed przyłączeniem zestawu w pierwszej kolejności jest konieczne skonfigurowanie układów generatora sygnału zegarowego. W wypadku, gdy AudioDSP jest taktowany wbudowanym oscylatorem kwarcowym, jak to będzie miało miejsce w tym odcinku kursu, należy sprawdzić, czy zworka OSC jest założona w pozycji XT. Pozostawienie jej w pozycji MCLK uniemożliwi pracę ADAU1701 ze względu na brak zewnętrznego sygnału zegarowego.



Rysunek 1. Konfiguracja sprzętowa płytki bazowej AudioDSP

Kolejną czynnością jest ustawienie mnożnika układu PLL. W związku z przystosowaniem AudioDSP do obróbki sygnału w standardzie CD z częstotliwością próbkowania $f_s=44,1$ kHz oraz wlutowanym kwarcem 11,2896 MHz jest wymagane ustawienie mnożnika PLL (zworki M0, M1) na krotność 256 ($f_s \times PLL = XT$; $44,1 \text{ kHz} \times 256 = 11,2896 \text{ MHz}$) zgodnie z opisem na płytce drukowanej. Jest to bardzo istotne podczas obróbki sygnału cyfrowego, na przykład z płytki interfejsu cyfrowego audio ADAU1701A_SPDIF lub interfejsu I²S, ponieważ ADAU1701 jest procesorem synchronicznym i musi pracować z częstotliwością zgodną z częstotliwością sygnału obrabianego. AudioDSP nie ma wbudowanych bloków konwertera częstotliwości próbkowania ASRC. Ustawiona częstotliwość próbkowania ma pewien wpływ na czułość i impedancję wejściową, ale przy zmianach z 11,2896 MHz na 12,288 MHz można go pominąć. Wbudowany rdzeń DSP jest taktowany przebiegiem o częstotliwości 49,152 MHz przy częstotliwości próbkowania 48 kHz, 96 kHz,



Rysunek 2. AudioDSP

192 kHz (45,1584 MHz dla wielokrotności 44,1 kHz). Przekłada się to na 1024 instrukcje przy 48 kHz ($1024 \times 48 \text{ kHz} = 49,152 \text{ MHz}$). Zwiększanie częstotliwości próbkowania zmniejsza liczbę instrukcji dostępnych do wykonania w cyklu i przy 192 kHz jest to już tylko 256. Należy zwrócić na to uwagę przy wyborze częstotliwości pracy systemu. Nie warto sztucznie zwiększać częstotliwości próbkowania, gdyż odbywa się to kosztem możliwych do wykonania na obrabianym sygnale operacji.

Układ PLL dopuszcza pewną elastyczność, umożliwiając ustawienie mnożnika f_s dla PLL równego 64, 256, 384 lub 512. Jest to przydatne, gdy zależy nam na zmniejszeniu poboru mocy lub zmianie częstotliwości kwarcu przy zachowaniu liczby instrukcji realizowanych w jednostce czasu. Ustawienie mnożnika opisano w tabeli 1. Przykładowo, możemy skonfigurować go np. z kwarcem 3,072 MHz i krotnością 64 (PLL M0=0, M1=0) dla częstotliwości 48 kHz ($64 \times 48 \text{ kHz}$). Błędne ustawienie źródła sygnału zegarowego i mnożnika PLL jest najczęstszym powodem problemów z uruchomieniem ADAU1701A, dlatego warto zwrócić szczególną uwagę na tę najważniejszą dla DSP ustawienie w tej części kursu:

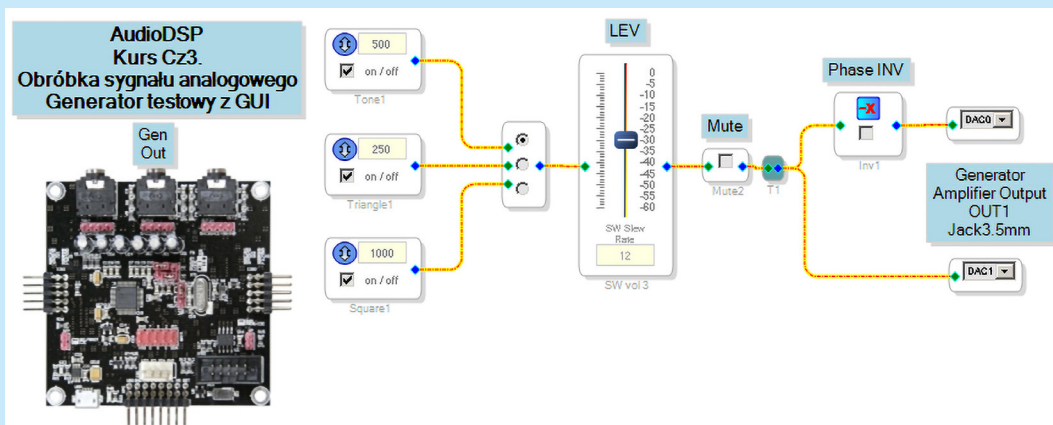
- **OSC** – zwora w położeniu XT (zwarne „górne” piny 2–3),
- **M0** – zwora w położeniu 256 (zwarne „lewe” piny 2–3),
- **M1** – zwora w położeniu 256 (zwarne „prawe” piny 1–2).

Zaletą wyboru mnożnika 256 jest też możliwość przetwarzania sygnałów o częstotliwości próbkowania 48 kHz, 96 kHz lub 192 kHz przy zastosowaniu typowego rezonatora kwarcowego 12,288 MHz (44,1 kHz, 88,2 kHz, 176,4 kHz dla kwarcu 11,2896 MHz).

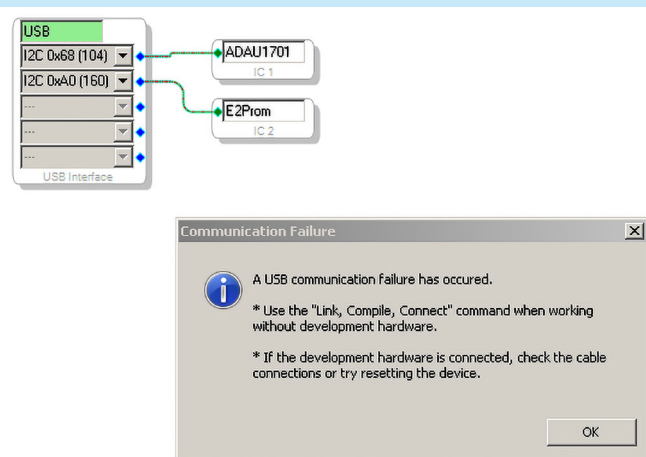
Po skonfigurowaniu układu generatora sygnału zegarowego, konieczne jest określenie trybu pracy DSP. Zwora SBT (Selfboot) umożliwia wybranie trybu pracy samodzielnej DSP, tzw. *selfboot*. W tym trybie przy zdjętej (rozwartej) zworze SBT procesor ADAU1701A, po załączeniu zasilania i poprawnym restarcie, pobiera konfigurację i parametry z zewnętrznej pamięci EEPROM. Uruchamia to zaprogramowaną wcześniej aplikację bez pomocy zewnętrznych układów sterujących. Aplikacja jest wykonywana, aż do wyłączenia napięcia zasilania. Przy jego spadku układ ADAU1701A zapisuje w pamięci EEPROM zmienione przez aplikację parametry. Ten tryb umożliwia pracę DSP bez zewnętrznego procesora sterującego. Zwora SBT pozostaje otwarta także podczas ładowania firmware z SigmaStudio, bez zapisywania do pamięci EEPROM w trakcie tworzenia projektu. W wypadku zwarcia zwory SBT układ ADAU1701A oczekuje na załadowanie aplikacji poprzez magistralę I²C z zewnątrz, na przykład przez procesor nadrzędny lub FPGA.

Ostatnią zworą jest WP (Write Protect EEPROM). Zwora odpowiada za zablokowanie dostępu do zapisu pamięci EEPROM. W trakcie normalnej pracy zwora jest rozwarła, blokując zapis do pamięci. Układ ADAU1701A po wykryciu spadku napięcia zasilającego ma możliwość odblokowania zapisu pamięci na czas niezbędny do zapamiętania parametrów aplikacji zmienionych podczas pracy. Zwarcie zwory WP i odblokowanie zapisu jest konieczne w trakcie programowania pamięci EEPROM za pomocą USBi.

Po skonfigurowaniu zwór dołączamy programator do złącza USBi i łączymy AudioDSP z komputerem PC. Programator powinien być widoczny w Menedżerze Urządzeń systemu Windows, co potwierdzają kontrolki zasilania: zielona na płytce AudioDSP



Rysunek 5. Schemat projektu Generator.dspproj



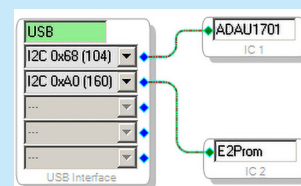
Rysunek 3. Problemy z przyłączeniem USBi

i czerwona w USBi. Powinien też nawiązać połączenie I²C, sygnalizując to zaświeceniem żółtej diody LED. Dioda ta sygnalizuje (przygasaniem) transmisję I²C w trakcie konfigurowania procesora lub programowania pamięci EEPROM.

W tym momencie nie pozostaje nic innego, jak uruchomienie SigmaStudio i przetestowanie pierwszego przykładu tej części kursu AudioDSP. Zgodnie z wcześniejszym opisem należy założyć nowy projekt, skonfigurować część sprzętową, określić ustawienia pamięci oraz przejść do zakładki *Config*. Domyślnie, w oknie jest pokazywana konfiguracja systemu. Po nawiązaniu połączenia pole USB zostanie podświetlone na zielono, a w wypadku problemów z komunikacją z USBi i/lub AudioDSP zostanie wyświetlony odpowiedni komunikat (rysunek 3). Wynika on najczęściej z kiepskiej jakości kabla Mini USB lub odwrotnego przyłączenia złącza programatora USBi (AudioDSP ma gniazdo kluczowane, więc nie powinno to mieć miejsca).

Przykład pierwszy – generator funkcyjny

Pierwszym przykładem jest generator funkcyjny, którego nastawy mogą być zmieniane nie tylko poprzez parametryzację lub wybór nastaw bloków bibliotecznych (w podstawowej wersji projektu), ale też przez graficzny interfejs użytkownika *Control UI* (zakładka *Hardware Configuration, TreeToolbox*) generujący wirtualny panel sterowania (w wersji zmodyfikowanej GUI), do którego można przypisać elementy manipulacyjne i takie jak gałka, suwak, przycisk, wyłącznik.



Rysunek 4. Konfiguracja sprzętowa Generator.dspproj


Projekt podstawowy nazwiemy **Generator.dssproj** i zapiszemy w podkatalogu kursu, najlepiej o nazwie zgodnej z nazwą projektu. Następnie, w zakładce **Hardware Configuration** umieszczamy blok procesora ADAU1701, pamięci EEPROM i łączymy zgodnie z **rysunkiem 4**.

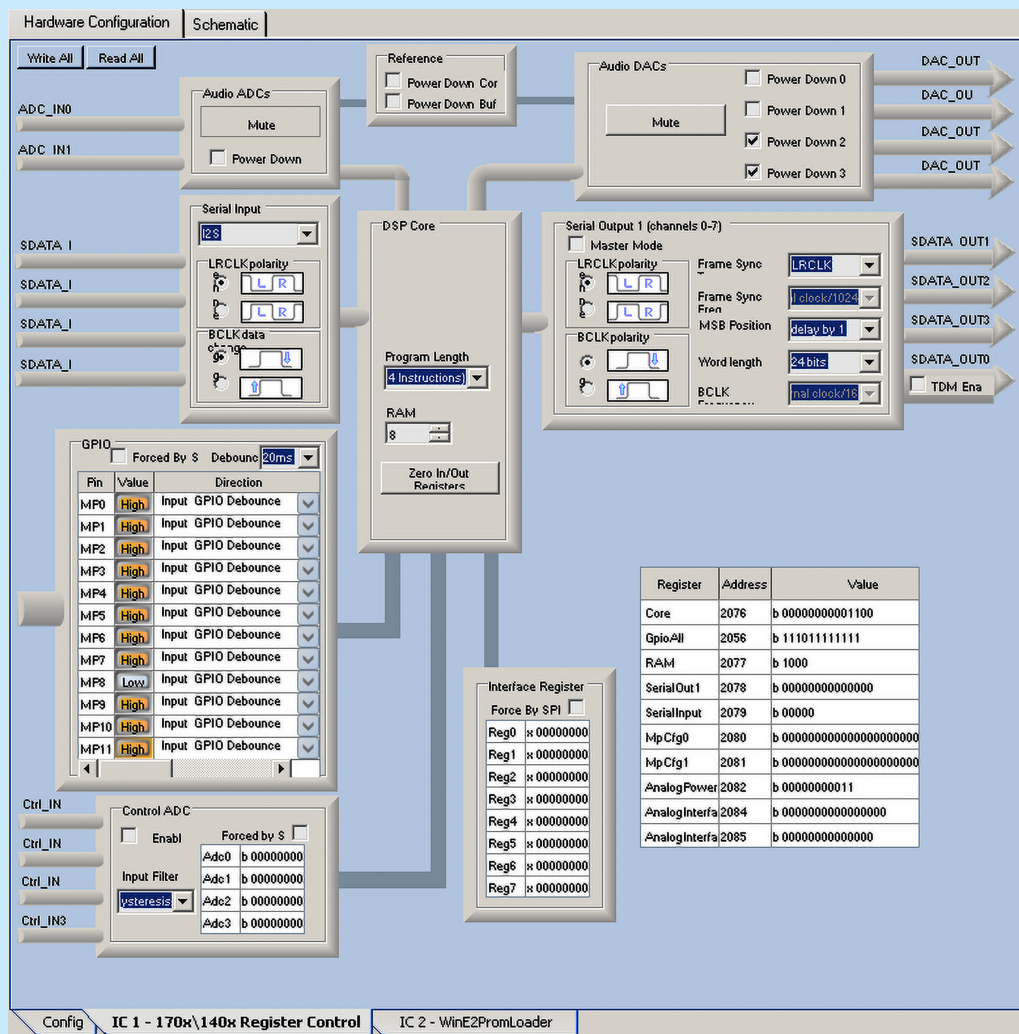
Projekt korzysta z domyślnej konfiguracji DSP, ale dla pewności warto sprawdzić, czy zgadza się ona z wymaganiami aplikacji: procesor musi być aktywny (Reference – Flagi Power Down Core/Bufers = OFF) i nie mogą być wyciszone wyjścia przetwornika C/A (z **rysunku 5**). Wejścia modułów ADC oraz DAC2, DAC3, Control ADC można wyłączyć, ponieważ są w projekcie nieużywane. Wyprowadzenia GPIO domyślnie ustawione są jako wejścia. Częstotliwość próbkowania wynosi 44,1 kHz dla rezonatora 11,2896 MHz lub 48 kHz dla 12,288 MHz. W razie konieczności zmiany częstotliwości fs wybieramy z menu **Action\Set System Sampling rate (CTRL+U)** i aktualizujemy bloki funkcjonalne **Action\Propagate Sampling Rate (CTRL+Q)**.

Po konfiguracji sprzętowej należy narysować schemat generatora.




- Z biblioteki **Sources** wybieramy generatory:
 - przebiegu sinusoidalnego **Sine Tone**,
 - przebiegu trójkątnego **Triangle wave**
 - prostokątnego **Square wave**.
- Z biblioteki **Volume Controls\AdjustableGain\SharedSlider\Clickless SW Slew (RC)** wybieramy regulator poziomu **Single SW slew vol (adjustable)**.
- Z biblioteki **Muxes/Demuxes\Switch\Clickless SW Slew\1Ch** multiplexer **Mono Switch Nx1**.
- Z biblioteki **Volume Controls\Mute** blok wyciszania **Mute**.
- Z biblioteki **Basic DSP\Logic\Invert** odwracacz fazy **Signal Invert**.
- Z biblioteki **IO\Output** wyjścia sygnału **Output**.

Bloki należy połączyć wg schematu z rysunku 5 i wstępnie sparametryzować, to jest załączyć generatory, ustawić ich częstotliwości, ustawić zakresy regulacji regulatora poziomu (klikając prawy klawisz myszy na polu regulatora) według **rysunku 6** oraz przypisać numery wyjść 0,1 do kanałów Output DAC.

Po narysowaniu schematu i skonfigurowaniu DSP (po otwarciu przykładowego projektu Generator.dssproj, konfiguracja odbędzie się automatycznie, ale nie należy iść na skróty, tylko wyklikać projekt samodzielnie, inaczej nie zapoznamy się z zawartością bibliotek) należy projekt podać działaniu linkera, wybierając z menu **Action\Link Project** lub ikonę  albo skrótem (CTRL+I). Po linkowaniu jest wyświetlana informacja o systemie i użytych zasobach i – w najlepszym wypadku – o braku błędów. Następnie projekt musimy skompilować z użyciem ikony **Link Compile Connect**.



Rysunek 5. Konfiguracja ADAU1701 Generator.dssproj

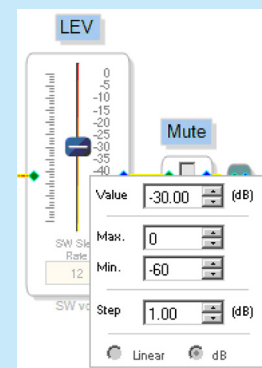
Po skompilowaniu warto sprawdzić prawy fragment paska statusu, ponieważ jest na nim wyświetlana informacja o statusie projektu. Po poprawnym linkowaniu i kompilacji pasek zmienia status projektu na skompilowany i gotowy do programowania  Ready: Compiled. Po wyborze opcji menu **Action\Link Compile Download** lub ikony  projekt zostaje załadowany do pamięci DSP (zdjęte zwory SBT, WP). Poprawne załadowanie potwierdzone jest zmianą statusu na  Active: Downloaded.

Po dołączeniu do wyjścia OUT1 AudioDSP wzmacniacza z głośnikami, na przykład minimodułu wzmacniacza stereofonicznego wzmacniacza o mocy 2x1,4 W (z układem SM2306, **rysunek 7**), zmieniając częstotliwości generatorów, ustawienia typu przebiegu multiplexerem, poziomu sygnału suwakiem regulatora, możemy przetestować działanie aplikacji. Rozbudowując aplikację o dodatkowe elementy, takie jak sumatory, miksery i inne, możemy zbudować generator testowy generujący tony DTMF, częstotliwości strojenia instrumentu muzycznego lub wielotony do testowania intermodulacji.

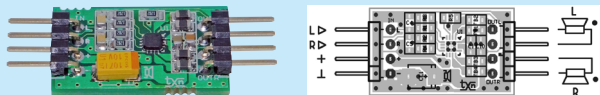
Jeżeli aplikacja działa poprawnie, zapisujemy projekt **Generator.dssproj**.

Przykład drugi – modyfikacja interfejsu graficznego

Drugim przykładem jest zmodyfikowany o graficzny interfejs

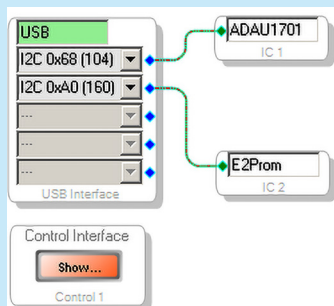


Rysunek 6. Parametryzacja regulatora poziomu



Rysunek 7. Wzmacniacz mocy SSM2306 przeznaczony do współpracy z AudioDSP

użytkownika GUI projekt *GeneratorGUI.dsproj*. W tym celu projekt *Generator.dsproj* zapisujemy pod nazwą zmienioną na *GeneratorGUI.dsproj* w podkatalogu kursu. Ciekawą opcją SigmaStudio jest możliwość tworzenia okienkowych interfejsów użytkownika, mających elementy manipulacyjne zawarte w schemacie aplikacyjnym, takie jak suwak, pokrętło, przycisk, klawisz. Znakomicie, pomimo pewnych ograniczeń interfejsu (tylko 4 predefiniowane elementy manipulacyjne), usprawnia to obsługę aplikacji, ponieważ

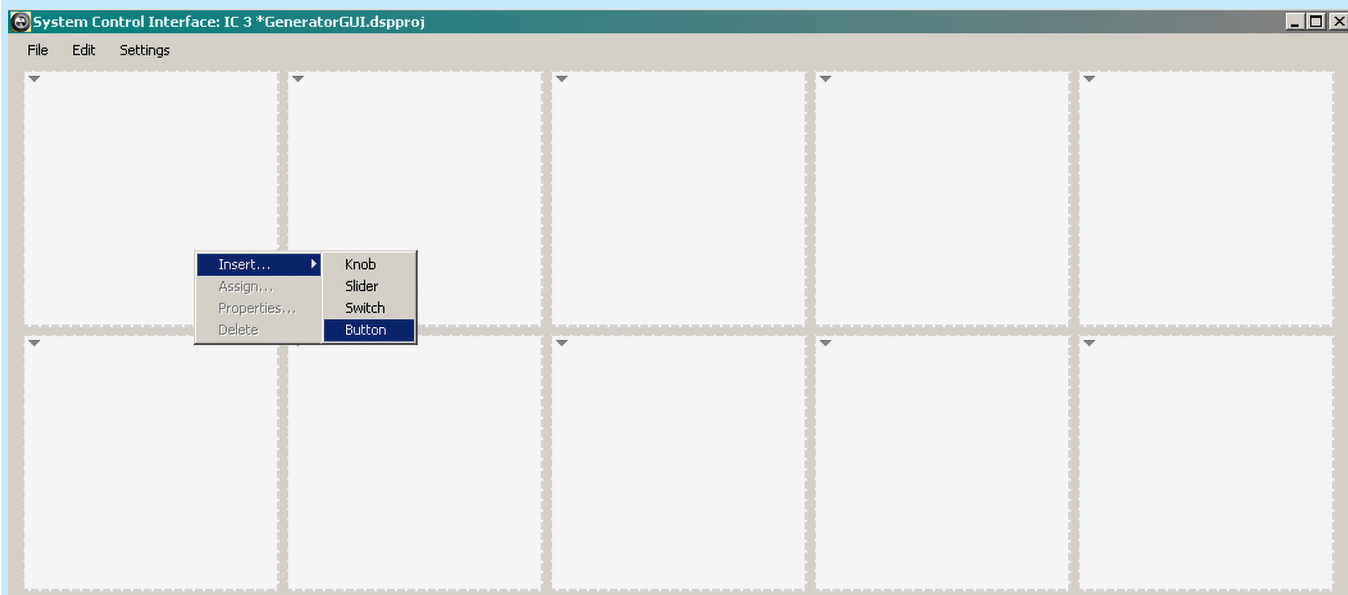


Rysunek 8. Konfigurowanie systemu z blokiem interfejsu użytkownika

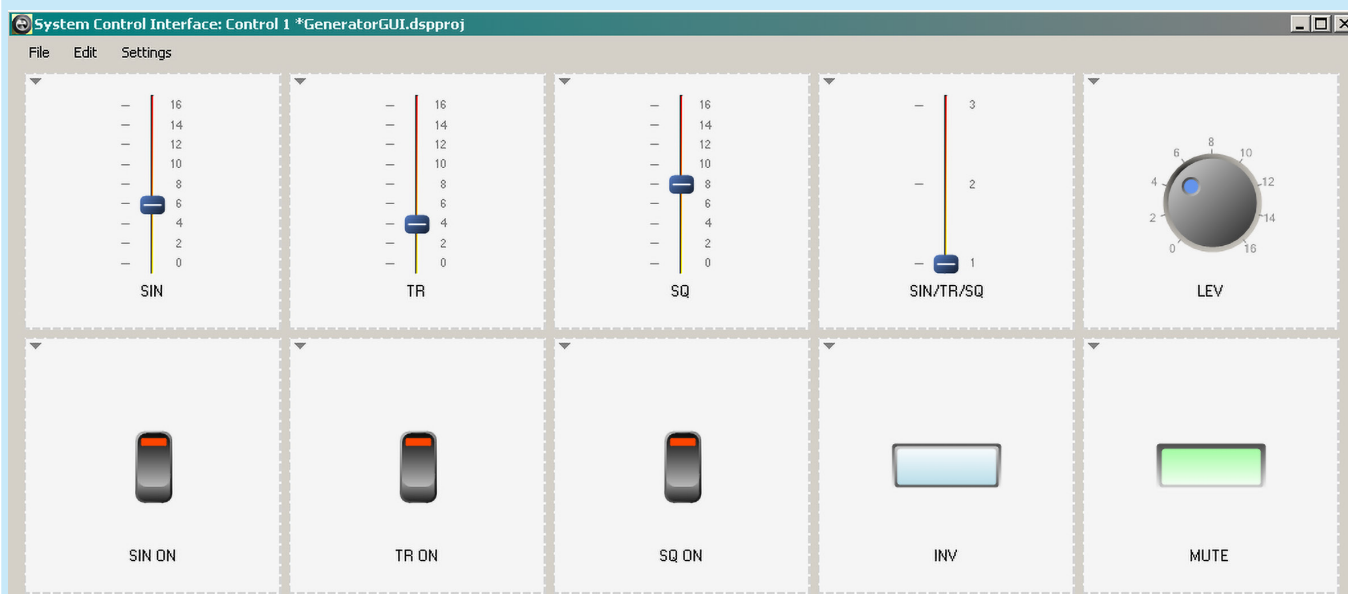
w większości wypadków mamy dostęp nie tylko do podstawowych parametrów bloków zdefiniowanych i wyświetlanych na ekranie, ale także do tych, które wymagają otwarcia menu szczegółowej konfiguracji bloku, na przykład w wypadku filtrów lub zwrotnic sygnałowych.

Aby uzyskać dostęp do nakładki z elementami manipulacyjnymi GUI, należy przejść do zakładki *Hardware Configuration* i z zakładki bibliotek *Processors* przeciągnąć w obszar roboczy blok *Control UI* (rysunek 8).

Interfejs użytkownika wymaga konfiguracji, która jest dostępna po naciśnięciu klawisza *Show...* (rysunek 9). Każde z 10 pól umożliwia skonfigurowanie elementu manipulacyjnego, takiego jak: pokrętło (Knob), suwak (Slider), przełącznik (Switch), klawisz (Button), sprzęgniętego z blokiem funkcjonalnym ze schematu, a odpowiadającym za ustawienie jego parametrów. Elementy dodajemy poprzez menu podręczne, dostępne po przyciśnięciu prawego klawisza myszy (rysunek 9). Na potrzeby naszego projektu testowego jest konieczne zdefiniowanie elementów interfejsu użytkownika (UI) zawierającego suwaki dla zmiany częstotliwości generatorów i wyboru kształtu przebiegu, pokrętło do regulacji poziomu sygnału, klawisze dla wyciszenia i odwrócenia fazy sygnału oraz przełączników dla aktywacji generatorów. Po dodaniu wszystkich elementów



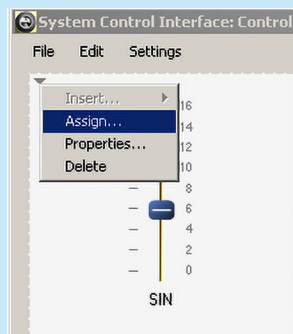
Rysunek 9. Konfigurowanie interfejsu użytkownika GUI



Rysunek 10. Skonfigurowane GUI projektu GeneratorGUI.dsproj

Tabela 1. Wybór mnożnika fs

Ustawienie fs	MCLK 48/96/192 kHz	MCLK 44,1 kHz	PLL_MODE0 Zwora M0	PLL_MODE1 Zwora M1	Mnożnik PLL (49,152 MHz)
64	3.072MHz	2.8224MHz	GND	GND	MCLKx16
256*	12.288MHz	11.2896MHz	GND	V3D	MCLKx4
384	18.432MHz	16.9344MHz	V3D	GND	MCLKx8/3
512	24.576MHz	22.5792MHz	V3D	V3D	MCLKx2

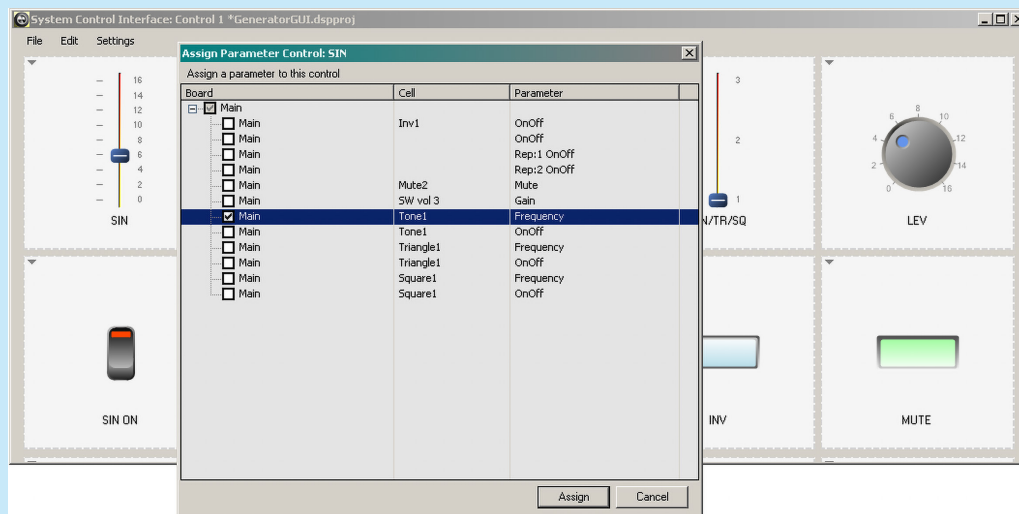


Rysunek 11. Przypisanie zmiennej sterującej elementem

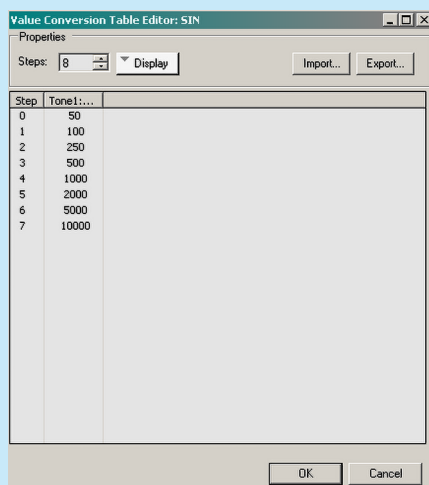
(rysunek 10) warto zmienić nazwy na bardziej zrozumiałe dla użytkownika, klikając lewym klawiszem myszy na pola pod elementami manipulacyjnymi.

Każdy z elementów wymaga indywidualnego przypisania zmiennej sterującej poprzez opcję **Assign...** dostępną po kliknięciu prawym klawiszem myszy w białym polu elementu (nie na samym elemencie) lub poprzez kliknięcie lewym klawiszem na małym trójkątku w lewym górnym narożniku pola (rysunek 11). Lista dostępnych zmiennych zależy od użytych bloków w projekcie, obejmując także istniejące schematy hierarchiczne. Aktualną listę zmiennych z projektu przykładowego pokazano na rysunku 12. Do jednego elementu można przypisać kilka zmiennych z bloku, ustawianych jednocześnie, na przykład zmiana częstotliwości filtra może zmieniać jego dobroć itp.

Dla każdego elementu należy ustalić liczbę kroków regulacji i wartość (wartości) zadaną zmiennej dla danego kroku z menu **Properties**. Przy mniejszej liczbie parametrów można zrobić to, wprowadzając odpowiednio nastawy ręcznie zgodnie z rysunkiem 13. Przy powtarzalnych elementach jest możliwe zdefiniowanie parametrów raz, zapisanie ich na dysku opcją **Export** i ponowne przypisanie pliku opcją **Import** dla pozostałych zdefiniowanych elementów (rysunek 14). Wartości zmiennych zapisywane są w pliku tekstowym (w formacie csv, ale z rozszerzeniem *.txt) i można je edytować dowolnym edytorem, przygotowując je wcześniej np. w arkuszu kalkulacyjnym. Jest to wygodne przy większej



Rysunek 12. Przykładowa lista zmiennych



Rysunek 13. Konfigurowanie elementu manipulacyjnego „suwak”

licznie kroków lub kilku zmienianych jednocześnie parametrach. W przypadku kilku parametrów w pliku, kolejne kolumny zawierają się w nawiasach klamrowych. Jeżeli liczba kroków regulacji nie zgadza się z liczbą linii w pliku, jest automatycznie uzupełniana wartościami zerowymi; jeżeli wartości jest więcej niż kroków, to wpisy są ignorowane. Jeżeli manipulator nie jest już potrzebny w aplikacji, można usunąć go opcją **Delete** z menu podręcznego. Przypisanie kilku zmiennych do elementu manipulacyjnego MUX pokazano na rysunku 15.

Edytor **Value Conversion Table editor** umożliwia podstawową edycję zawartości komórek, dodawanie, usuwanie, kopiowanie i zmianę kolejności aktywnych kolumn (rysunek 16). W celu aktywowania opcji edycji kolumn należy kliknąć prawym klawiszem myszy na nagłówek z opisem kolumny.

Konfiguracja przełącznika (Switch), klawisza (Button) sprowadza się do przypisania odpowiedniej zmiennej **NazwaZmiennej:On/Off** i zdefiniowania dwóch stanów wyłączenia (zmienna =0) i załączenia (zmienna =1), jak na rysunku 17.

Po zapisaniu, skompilowaniu i załadowaniu projektu mamy możliwość obsługi aplikacji z poziomu elementów schematu oraz z okna interfejsu użytkownika. Okienko interfejsu może zostać zamknięte przyciskiem na belce tytułowej. Ponowne jego wywołanie następuje po naciśnięciu klawisza **Show...** bloku **Control UI**. Konfiguracja okienka UI może zostać zapisana w pliku *.xml wraz z parametrami po wyborze opcji **SaveAs** z menu okna interfejsu. Jeżeli domyślne 10 elementów to zbyt mało lub zbyt dużo, to możemy dodać/usunąć kolejne wiersze z menu **Edit/Add/Remove Row**.

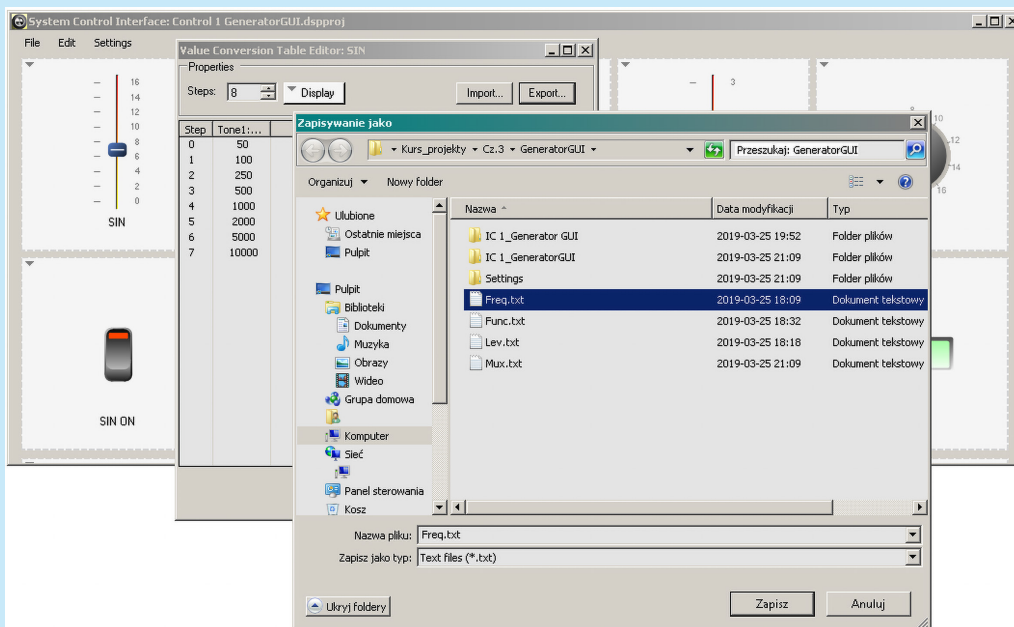
Aplikacja z przygotowanym interfejsem użytkownika może zostać zabezpieczona przed nieautoryzowaną modyfikacją opcją **Freeze** z menu **Action/Freeze Schematic (CTRL+F)**. Można też ukryć zawartość bloków funkcjonalnych, klikając na bloku prawym klawiszem myszy, wybierając opcję **Disable This Control** i podając hasło odblokowujące. Po ukryciu wszystkich bloków i ochronie hasłem, użytkownikowi pozostaje „pusty” schemat i możliwość korzystania tylko ze zdefiniowanych elementów w GUI (rysunek 18). Odblokowanie elementu do ponownej edycji wykonujemy, klikając

prawym klawiszem myszy w obrębie bloku, co wywołuje menu podręczne **Enable This Control**, jak na rysunku 19, po którego wybraniu zostaniemy poproszeni o podanie hasła odblokowującego element (rysunek 20).

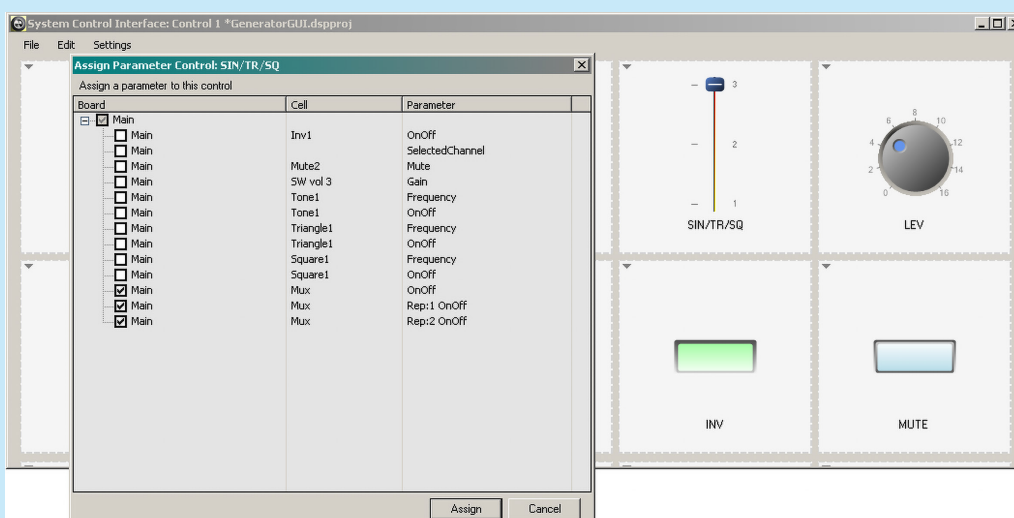
Przykład trzeci – obróbka sygnału z mikrofonu

Trzecim projektem będzie **SuperMic.dspproj**, czyli projekt toru obróbki sygnału z mikrofonu zawierający filtr pasmowy, kompresor dynamiki i bramkę szumów, czyli wszystko, co powinien zawierać najprostsz procesor sygnału z mikrofonu. Konfiguracja sprzętowa AudioDSP jest identyczna z pokazaną na rysunku 4.

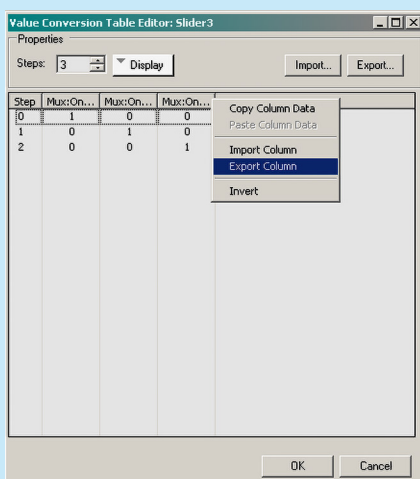
- Z bibliotek w arkusz projektu przenosimy następujące bloki:
- Z biblioteki **Volume Controls\AdjustableGain\SharedSlider\Clickless SW Slew (RC)** wybieramy regulator poziomu **Single SW slew vol (adjustable)**.
- Z biblioteki **Volume Controls\Mute** blok wyciszania **Mute**.
- Z biblioteki **Muxes/Demuxes\Demultiplexers\Switch\Clickless SW Slew\1Ch** multiplexer **Mono Switch 1xN**.
- Z biblioteki **Filters\Miscellaneous** blok usuwający składową stałą **DC Blocking**.
- Z biblioteki **Non Linear Processors\Clippers\Soft Clip** blok ogranicznika **Standard Cubic**.
- Z biblioteki **Filters\Nth Orders\Double Precision\1 Channel** filtr **Nth Order Filter**.
- Z biblioteki **Mixers\Splitters\Single**



Rysunek 14. Zapis i odczyt konfiguracji elementu



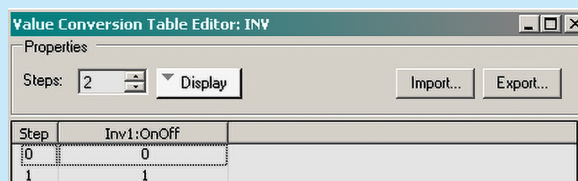
Rysunek 15. Konfigurowanie parametrów elementu manipulacyjnego suwaka (Slider) MUX



Rysunek 16. Konfigurowanie właściwości elementu manipulacyjnego MUX

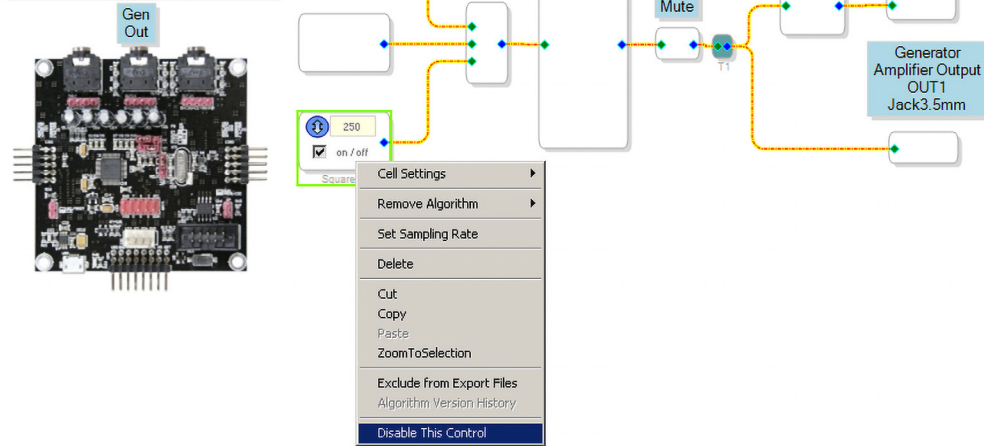
- Z biblioteki **Control\No Slew (Standard)** blok sumatora sygnału **Signal Merger**.
- Z biblioteki **Basic DSP\Adjustable Gain\No Slew** blok wzmocnienia **Gain**.
- Z biblioteki **IO\Output** wejścia i wyjścia sygnału **Input, Output**.
- Z biblioteki **Level Detectors\Lookup tables\Level Detectors\Single Band** wskaźnik poziomu sygnału **Single Level Detector**.
- Z zakładki Schematic Design bloki **Simulation Probe** i **Simulation Stimuli**.

Bloki należy połączyć zgodnie ze schematem pokazanym na rysunku 21. W celu zwiększenia czytelności schematu można wykorzystać etykiety **Alias**, które można przypisać do wyprowadzeń



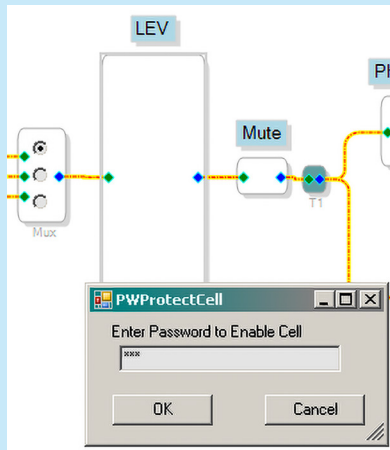
Rysunek 17. Definiowanie stanów przelącznika i klawisza

AudioDSP
Kurs Cz3.
Obróbka sygnału analogowego
Generator testowy z GUI



Rysunek 18 Zabezpieczenie projektu

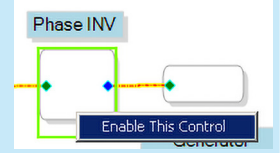
bloków (tylko przed ich połączeniem), zamiennie z bezpośrednim połączeniem przewodem. Etykiety przypisujemy poprzez wskazanie kursorem kropki wyrowadzenia i przyciśnięcie prawego klawisza myszy (rysunek 22). Po wyborze pojawi się para bloków etykiet kojarząca sygnał (rysunek 23). Można je rozmieścić w dowolnej części schematu, co uwalnia od konieczności prowadzenia długich połączeń przez cały arkusz. Do jednego wyrowadzenia może być dołączona tylko jedna etykieta. Jeżeli potrzebujemy sygnał rozdzielić na kilka etykiet,



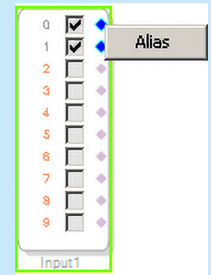
Rysunek 20. Wprowadzenia hasła odblokowującego

analogicznie jak dla połączenia, musimy wykorzystać blok T-Connection i odpowiednią liczbę utworzonych etykiet.

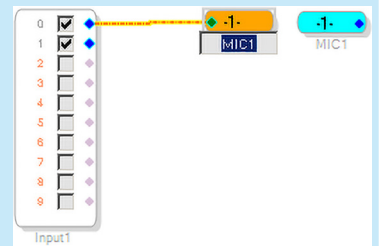
Projekt *SuperMic.dspproj* wykorzystuje kolejny minimoduł Pre_MicMEMS (rysunek 24), który jest stereofonicznym mikrofonem MEMS (SPU0410) przeznaczonym do współpracy z AudioDSP. Mikrofon ma wbudowany przedwzmacniacz o wzmacnieniu 20 dB. Sygnał na mikrofon jest doprowadzony do złącza IN1A za pomocą taśmy SIP4 (przy długości do 10 cm nie musi ona być ekranowana). Do rdzenia DSP sygnał jest doprowadzony przez blok *Input*, skąd po odseparowaniu składowej stałej *DCB1*, ograniczeniu poziomu w bloku *Clipper*, wstępnym wzmacnieniu *Gain* podlega sumowaniu *Mg1*. Blok ogranicznika ma łagodzić skutki przesterowania toru związane z chwilowymi stanami niestabilnymi, jak dla przykładu uderzenie mikrofonem. Układ wykorzystuje funkcje sześcienne dla złagodzenia wierzchołków przebiegu.



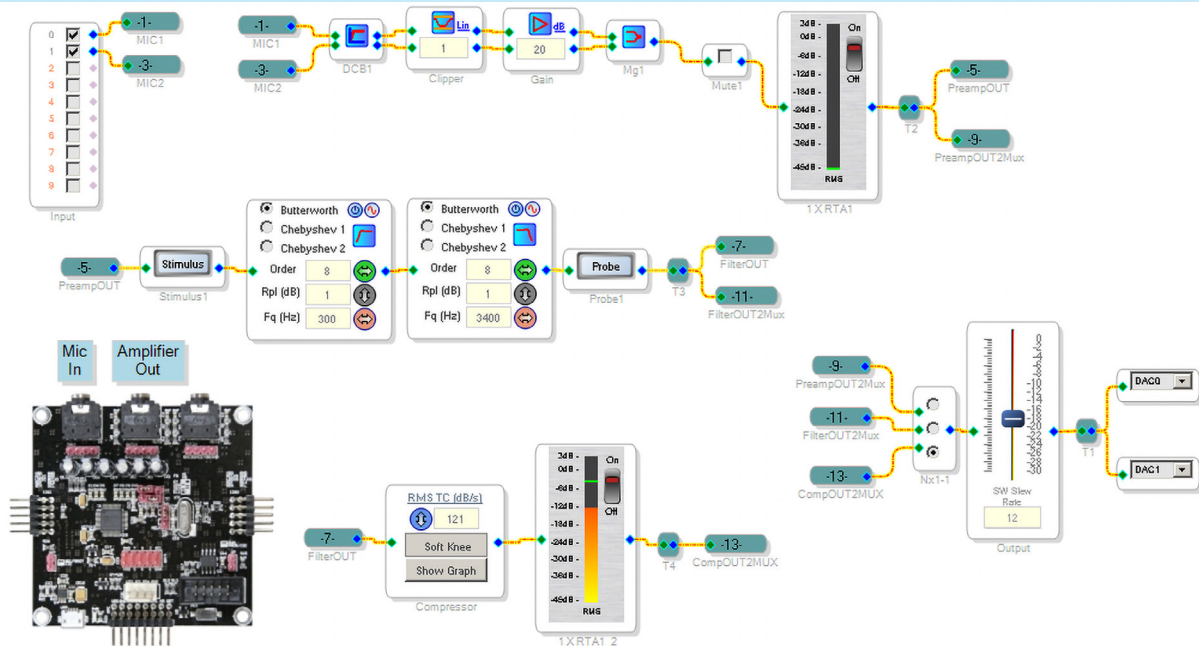
Rysunek 19. Opcja przywrócenia edycji bloku



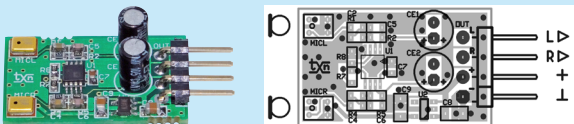
Rysunek 22. Nadawanie etykiety wyrowadzom



Rysunek 23. Edycja etykiet



Rysunek 21. Schemat projektu SuperMic.dspproj



Rysunek 24. Minimoduł stereofonicznego mikrofonu MEMS do współpracy z AudioDSP

Niestety, wartość ograniczania nie jest zdefiniowana w dB lub %. Wartość 2 odpowiada niewielkiemu ograniczeniu na poziomie około 95%. Dokładny opis funkcji jest dostępny w pomocy systemowej.

Po ewentualnym ograniczeniu sygnał jest doprowadzony do bloku wzmacnienia **Gain**, gdzie na drodze czysto programowej jest wzmacniony

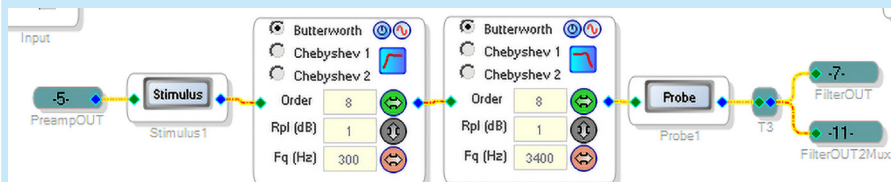
o 20 dB. Z używaniem bloku **Gain** w DSP wiąże się niebezpieczeństwo podniesienia poziomu szumu wzmacnianego sygnału i utraty rozdzielczości przetwornika A/C pracującego w niepełnym zakresie napięć wejściowych. W tej aplikacji można dopuścić taki kompromis, ponieważ dynamikę toru określa i tak poziom szumu mikrofonu MEMS. Następnie sygnał jest doprowadzony do bloku **Signal Merger** – automatycznego miksera, który dopasowuje poziom sygnału wyjściowego do liczby zdefiniowanych wejść, zapobiegając jego obcinaniu podczas sumowania. Sumowanie sygnału z mikrofonów pozwala na niewielką korektę charakterystyki kierunkowej mikrofonów dla sygnałów wspólnych. Sygnał zsumowany przez blok wyciszania **Mute** jest podawany na wskaźnik poziomu sygnału **Single Level Detector**. Oba bloki są przydatne podczas

uruchamiania aplikacji (na przykład, dobierania wzmacnienia) i docelowo mogą być pominięte. Aktywowanie wskaźnika wysterowania następuje po załączeniu przełącznika On/Off.

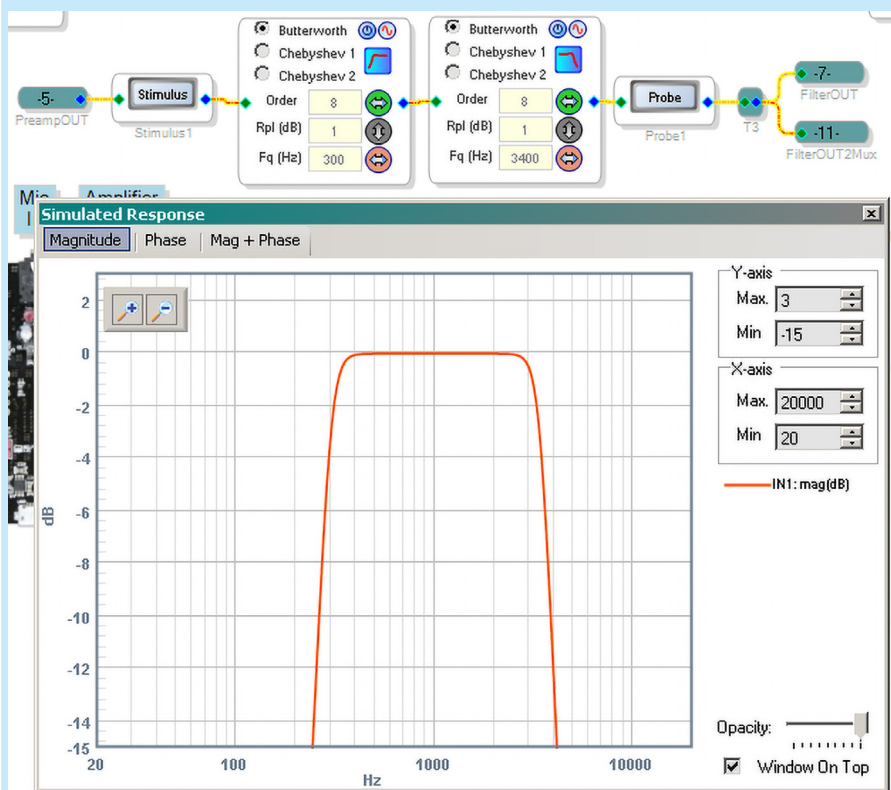
Z przedwzmacniacza sygnał jest podawany na blok filtrów (rysunek 25) kształtujących charakterystykę toru. Blok umożliwia wybór trybu filtru, charakterystyki i rzędu. Z łatwością można skonfigurować filtr wyższych rzędów przez łatwą zmianę parametrów, co jest niewyobrażalne w filtrach realizowanych analogowo. Większość bloków filtrów i korektorów występuje w dwóch wersjach, różniących się dokładnością obliczeń **single/double precision**. Filtry **double precision** wykorzystują 56-bitową arytmetykę do obliczenia odpowiedzi, co pociąga za sobą większe zużycie zasobów DSP. W środowisku SigmaStudio jest dostępne narzędzie do symulowania działania filtrów i korektorów, umożliwiające zdjęcie charakterystyk fazowej i przenoszenia filtru. Na schemacie umieszcza się przed filtrem blok **Stimulus**, a za filtrem blok **Probe**. O ile w filtrach, w których parametryzacja odbywa się za pomocą nastaw odpowiedników analogowych, można się spodziewać efektów, o tyle w filtrach, w których charakterystyka jest określana tablicą współczynników (np. FIR), jest to bardzo pomocne narzędzie. Aby wykreślić charakterystykę, otwieramy okienko wykresu, naciskając przyciski **Probe** i **Stimulus**, co generuje obliczoną odpowiedź filtru. Przykład pokazano na rysunku 26.

Po wyodrębnieniu sygnału mowy z zakresu częstotliwości 300 Hz...3,4 kHz sygnał jest, podawany na blok kompresora. W bloku można elastycznie konfigurować charakterystykę przejścia za pomocą klawisza **Show Graph** (rysunek 27). Możliwy jest też tryb kompresora wokół punktów charakterystycznych z ostrym i łagodnym przejściem (Soft/Hard Knee).

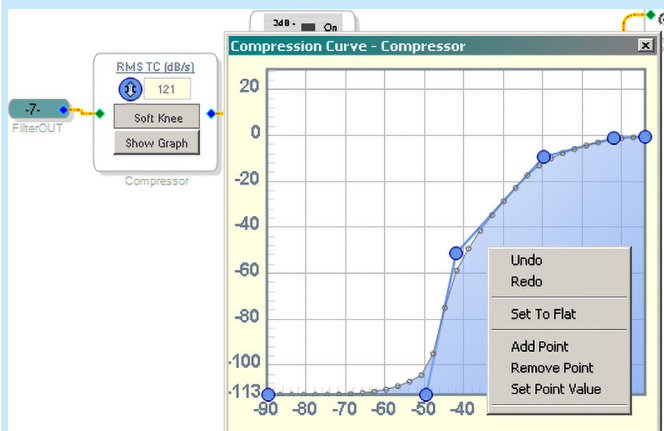
W oknie edycji charakterystyki możemy przesuwać, dodawać i usuwać punkty charakterystyczne. Możliwe jest też wprowadzanie dokładnych parametrów opcją **Set Point Value**. Oprócz kompresora blok przy odpowiedniej parametryzacji umożliwia wykonanie bramki szumów, ogranicznika, wzmacniacza, a także ekspandera sygnału. Przyłączony do wyjścia kompresora wskaźnik poziomu pozwala na wizualną ocenę jego funkcjonowania. Kompresor nie ma możliwości ustawiania czasów Attack/Release, pracuje w trybie TC (time constant), w którym jest określana tylko wartość szybkości odpowiedzi kompresora na zmiany na wejściu (dB/s). Układ sprawdza się podczas kompresji dłuższych, głośniejszych partii, niezbyt agresywnie wpływając na krótkie szybkie sygnały.



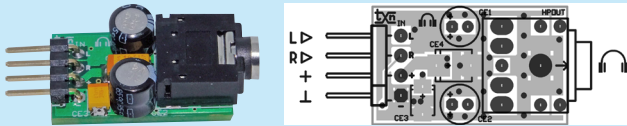
Rysunek 25. Blok filtrów



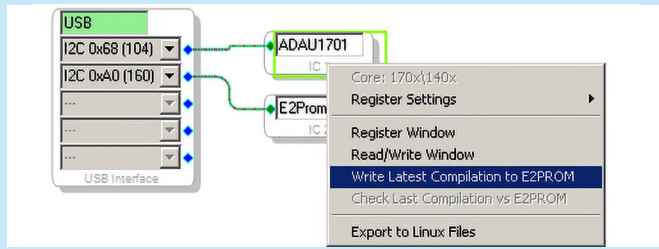
Rysunek 26. Symulowana charakterystyka filtru



Rysunek 27. Konfigurowanie kompresora



Rysunek 28. Minimoduł ze wzmacniaczem słuchawkowym dla AudioDSP



Rysunek 29. Programowanie pamięci EEPROM

Sygnaly z bloków przedwzmacniacza, filtrów i kompresora doprowadzone są do multiplexera i regulatora poziomu wyjściowego. Wybierając odpowiednie wyjście, można sprawdzić skuteczność filtracji i kompresji w porównaniu z czystym sygnałem z przedwzmacniacza. Ze względu na możliwość sprzęgania się mikrofonu z głośnikami podczas uruchamiania warto użyć słuchawek z modulem wzmacniacza słuchawkowego mającego możliwość sterowania słuchawek niskoomowych o rezystancji 16...60 Ω (rysunek 28).

Na koniec

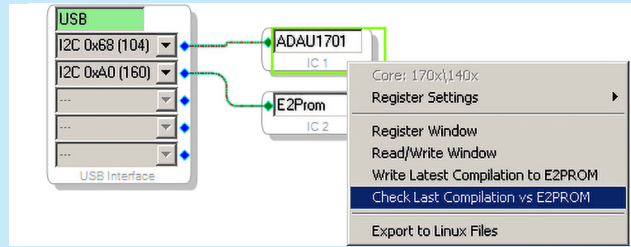
Ze względu na zapis aplikacji bezpośrednio do pamięci DSP, każdorazowo po wyłączeniu zasilania AudioDSP lub restarcie DSP aplikacja i konfiguracja jest tracona i wymaga ponownego załadowania. Aby układ ADAU1701A pracował samodzielnie, należy aplikację wgrać do pamięci nieulotnej EEPROM i użyć trybu *Selfboot*. W celu zaprogramowania pamięci EEPROM po testach funkcjonalnych należy przejść do zakładki **Hardware Configuration** i otworzyć naciśnięciem na elemencie ADAU1701 prawym klawiszem myszy menu podręczne pokazane na rysunku 29. Opcja służąca do programowania EEPROM to **Write Latest Compilation to E2PROM**.

Uwaga! Przed uruchomieniem programowania konieczne jest zwarcie zwory WP, w przeciwnym wypadku pamięć nie zostanie zaprogramowana pomimo potwierdzenia postępu zapisu przez SigmaStudio!

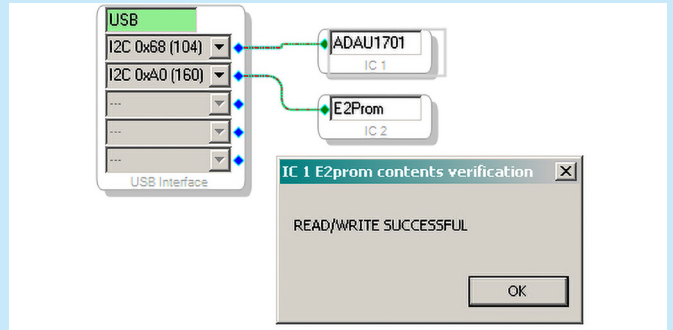
Po zaprogramowaniu warto zweryfikować poprawność zapisu opcją **Check Last Compilation vs E2PROM** (rysunek 30). Zgodna zawartość zostanie potwierdzona komunikatem (rysunek 31).

W razie wystąpienia błędów zapisu jest konieczne sprawdzenie konfiguracji pamięci E2EPROM oraz prawidłowego przebiegu programowania, to jest zwarcie zwory WP. Po poprawnym zapisie po każdym wyłączeniu zasilania lub restarcie procesor ADAU1701 pobierze aplikację z pamięci nieulotnej i rozpocznie samodzielną pracę, co można potwierdzić, przyciskając *Reset* lub wyłączając na chwilę zasilanie AudioDSP. W nielicznych wypadkach zasilania z USBI jest konieczne odłączenie kabla programującego, ponieważ programator aktywuje sygnał *reset*, blokując *start DSP*.

SigmaStudio zawiera także narzędzie do bardziej złożonych manipulacji na pamięci EEPROM. Po otwarciu zakładki **Hardware configuration**



Rysunek 30. Weryfikowanie zawartości EEPROM

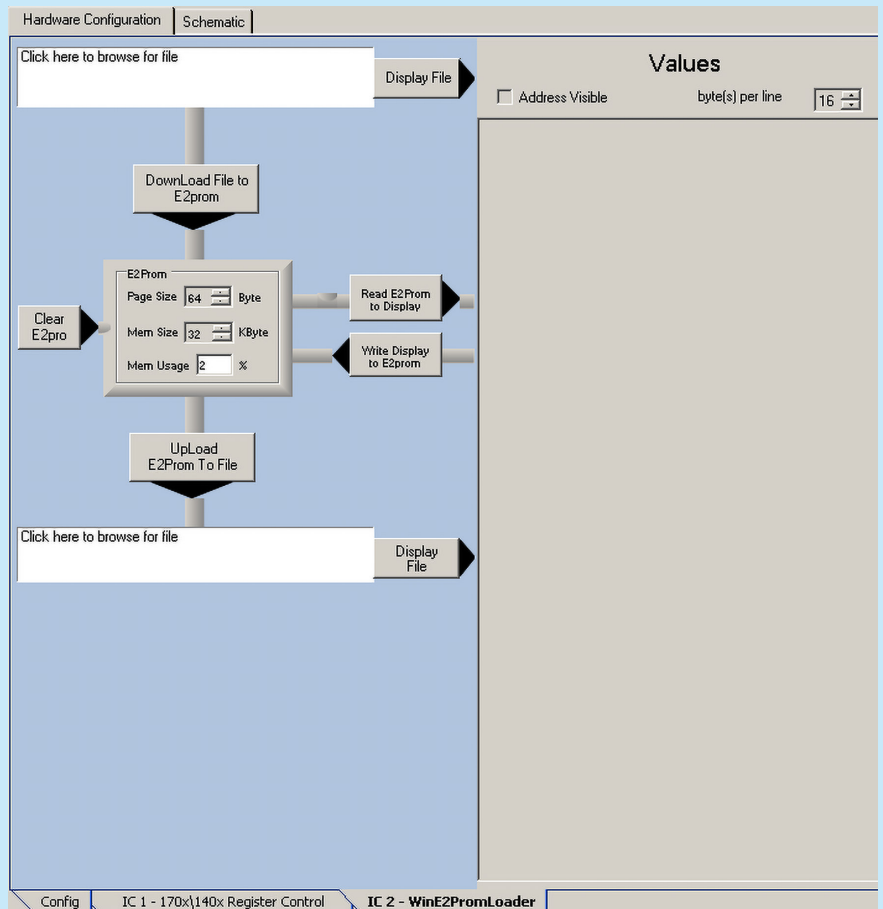


Rysunek 31. Potwierdzenie poprawności zapisu

I2 → **WinE2Promloader** uzyskujemy dostęp do możliwości kasowania zawartości, odczytu, edycji, zapisu pamięci bezpośrednio z pliku i do niego (rysunek 32). Umożliwia to generowanie plików dla zewnętrznych programatorów lub wykorzystanie przygotowanych produkcyjnych obrazów pamięci.

Sposób zmiany parametrów aplikacji poprzez obsługę GPIO przydatny, gdy AudioDSP pracuje samodzielnie, bez komunikacji z SigmaStudio, zostanie przedstawiony w kolejnym artykule.

Adam Tatuś, EP



Rysunek 32. Edytor zawartości EEPROM