

AudioDSP zestaw z procesorem Sigma DSP ADAU1701 (4)

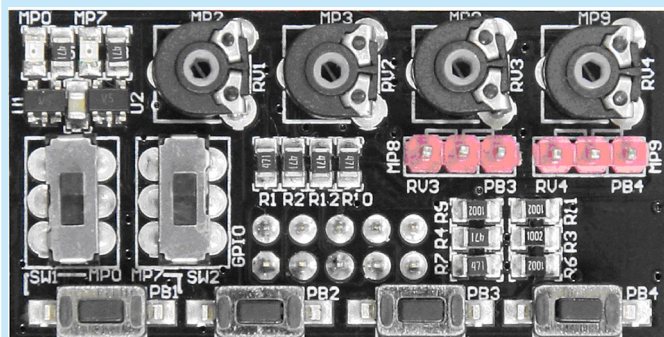
Obsługa GPIO



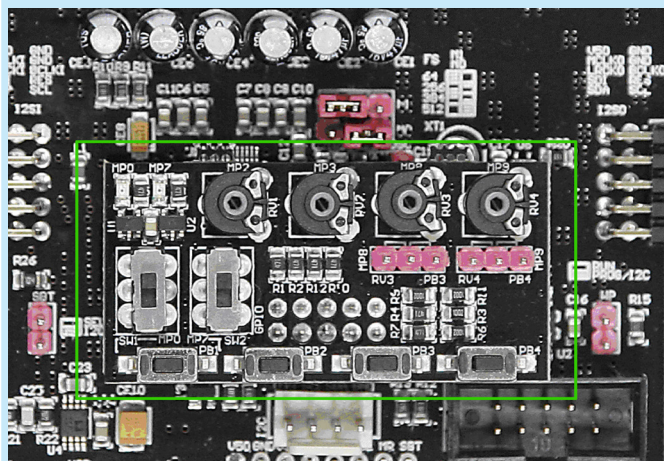
W czwartej części kursu zostanie przedstawiony sposób obsługi interfejsu GPIO, czyli najłatwiejszy sposób na realizację interfejsu użytkownika, bez potrzeby uciekania się do sterowania zewnętrznym procesorem lub stałego połączenia z SigmaStudio.

Do przeprowadzenia testów, oprócz zapoznania się z poprzednimi odcinkami kursu, konieczne będzie przygotowanie części sprzętowej tj.: płytki bazowej AudioDSP wraz z interfejsem użytkownika ADAU1701A_GPIO **rysunek 1**, 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, kabel żeński 4 pin 2,54 mm do połączenia minimodułów oraz płytka enkodera obrotowego.

Płytkę rozszerzeń GPIO montuje się bez dodatkowych przewodów bezpośrednio na płytce bazowej AudioDSP, korzystając z gniazda GPIO **rysunek 2**. Schemat płytki GPIO przedstawia **rysunek 3**.



Rysunek 1. Wygląd płytki interfejsu użytkownika ADAU1701A_GPIO

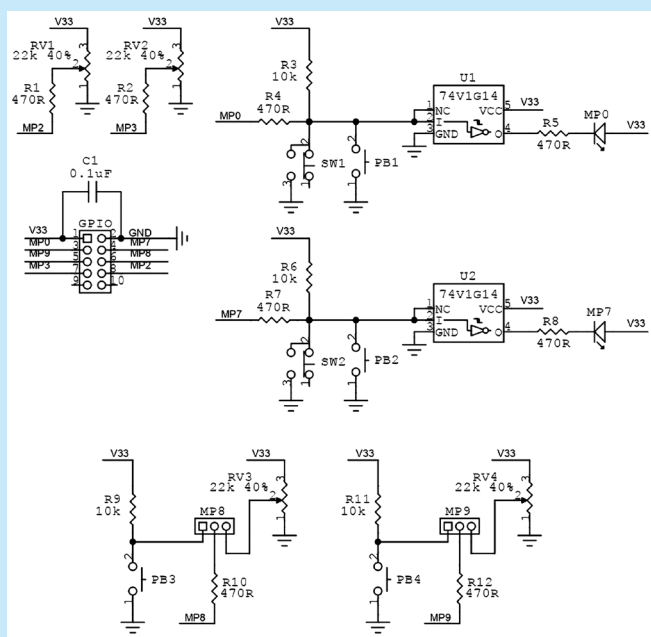


Rysunek 2. Sposób montażu GPIO

Procesor DSP ADAU1701, podobnie jak „klasyczne” procesory, ma programowalne, wielofunkcyjne wyprowadzenia we/wy GPIO (oznaczone MPxx, multipurpose IO). Każde z wyprowadzeń MPxx może zostać indywidualnie skonfigurowane i oprogramowane do realizacji:

- cyfrowych wejść i wyjść sterujących,
- wejść sterujących pomocniczego przetwornika ADC,
- wejściowych i wyjściowych szeregowych interfejsów audio (tryb P'S, LJ, RJ, TDM), np.: do podłączenia zewnętrznych układów przetworników AD/DA, odbiorników/nadajników SPDIF itp.

Interfejs GPIO może zostać wykorzystany do realizacji cyfrowej lub analogowej kontroli parametrów w algorytmie DSP. Pozwala to nawet w złożonych aplikacjach na rezygnację z zastosowania zewnętrznego procesora sterującego. Płytkę DSP ADAU1701A_GPIO ma wbudowane typowe manipulatory, takie jak cztery przyciski, dwa przełączniki suwakowe, cztery potencjometry oraz dwie buforowane diody LED. Potencjometry RV1,2 podłączone są na stałe do wyprowadzeń GPIO2,3. Potencjometry RV3,4 współdzielą porty MP8,9 z przyciskami chwilowymi PB3,4, wybór aktywnej funkcji możliwy jest



Rysunek 3. Schemat płytki interfejsu użytkownika ADAU1701_GPIO

Tabela 1. Zestawienie funkcjonalności GPIO

MP	Tryb GPIO			Złącze GPIO		Złącze I2SI		Złącze I2SO	
	ADC	Serial (I2S,LJ,RJ)	TDM	PIN	Funkcja	PIN	Funkcja	PIN	Funkcja
MP0		SDATA_IN0	TDM_IN	3	LED MP0, SW1, PB1				
MP1		SDATA_IN1				8	SDI		
MP2	ADC1	SDATA_IN2		8	POT RV1				
MP3	ADC2	SDATA_IN3		7	POT RV2				
MP4		INPUT_LRCLK slave				6	LRCKI		
MP5		INPUT_BCLK slave				5	BCLKI		
MP6		SDATA_OUT0	TDM_OUT					7	SDO
MP7		SDATA_OUT1		4	LED MP7, SW2, PB2				
MP8	ADC3	SDATA_OUT2		6	POT RV3, PB3, wybór zworką MP8				
MP9	ADC4	SDATA_OUT3		5	POT RV4, PB4, wybór zworką MP9				
MP10		OUTPUT_LRCLK master/slave						5	LRCKO
MP11		OUTPUT_BCLK master/slave						6	BCLKO

zworkami MP8,9. Porty MP0,1 połączone są z przyciskami chwilowymi PB1,2 z równolegle połączonymi przełącznikami suwakowymi SW1,2. Porty MP0,1 wyposażone są także w diody LED sygnalizujących ich stan, gdy pracują jako wyjścia.

UWAGA: GPIO pracują ze standardem napięciowym 3,3 V IOVDD, przekroczenie tego napięcia trwale uszkodzi DSP.

W trybie pracy jako wejście cyfrowe (**Input GPIO**), wyprowadzenia mają wbudowany wewnętrzny rezystor podciągający o wartości ok. 15 kΩ. Podczas konfiguracji można aktywować wbudowany filtr eliminujący drgania styków (**debounce**) o czterech ustawianych czasach uśredniania.

W trybie wyjścia cyfrowego możliwy jest wybór konfiguracji Push-Pull (**Output GPIO**) lub OC (**Output GPIO Open Collector**). Wydajność prądowa wyprowadzeń jest niewielka (ok. 2 mA). Na płytce GPIO w roli buforów zastosowane są bramki serii Single Gates 1G14. Szeregowe rezystory 470 Ω chronią przed skutkami błędnej konfiguracji GPIO. Przy wykorzystaniu większej liczby wyjść należy pamiętać o sumarycznym pobieranym prądzie. Zalecane jest użycie buforów lub tranzystora pośredniego. Napięcie doprowadzone do GPIO nie może przekraczać VC-CIO (3,3 V) także w przypadku wyjść OC. Dla wyjścia OC może być jednak niższe, co ułatwia sprzężenie z układami zasilanymi np.: 1,8 V, 2,5 V (oczywiście z zewnętrznym rezystorem podciągającym).

Wbudowany pomocniczy przetwornik ADC ma 4 kanały o rozdzielczości 8 bit, zakres napięć wejściowych to 0...3,0 V, co w przybliżeniu daje krok 12 mV/bit, przy rezystancji wejścia ok. 30 kΩ. Wejście ma charakterystykę liniową, najlepiej współpracuje z potencjometrem lub źródłem napięcia o liniowej charakterystyce zmian. Zmiana charakterystyki regulacji powinna być realizowana na drodze programowej w DSP np. przez tablice wartości LUT, zapewni to najwyższą dokładność regulacji i pełne wykorzystanie rozdzielczości ADC.

Interfejs GPIO domyślnie taktowany jest sygnałem fs, co nie znaczy, że należy wykorzystywać go w roli przetwornika sygnału audio (zbyt mała rozdzielczość).

Podczas projektowania własnego systemu istotne jest odpowiednie przyporządkowanie wyprowadzeń, szczególnie np. interfejsów cyfrowych, które „blokują” sporą liczbę pinów GPIO, w tym ADC, możliwą do wykorzystania dla interfejsu użytkownika. Gdy aplikacja wykorzystuje wszystkie piny wyprowadzone na złącze GPIO, nie ma przeciwwskazań do wykorzystania pozostałych MPxx wyprowadzonych na złącza I2SI, I2SO. Kompletnie zestawienie funkcjonalności GPIO przedstawia tabela 1.

Od strony SigmaStudio konfiguracja sprzętowa GPIO dostępna jest w oknie **Hardware Configuration**, a wybór realizowanej przez

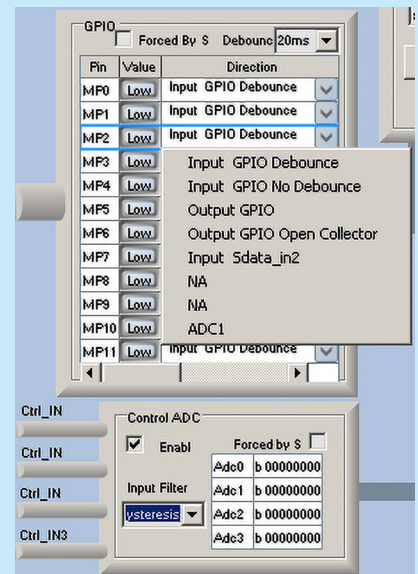
GPIO funkcji dostępny jest w zakładce **IC-1-1701x/1401x Register Control** rysunek 4.

Z menu podręcznego możliwa jest konfiguracja funkcji dla każdego z wyprowadzeń oraz zbiorcza konfiguracja czasu eliminatora drgań styków. W przypadku wykorzystania przetwornika ADC konieczna jest jego aktywacja **Enabl**. Wejścia analogowe mają konfigurowany filtr zakłóceń, w którym możliwe jest ustawienie histerazy wejścia (4, 5 bitów lub brak) oraz wyłączenie filtracji (No Low Pass).

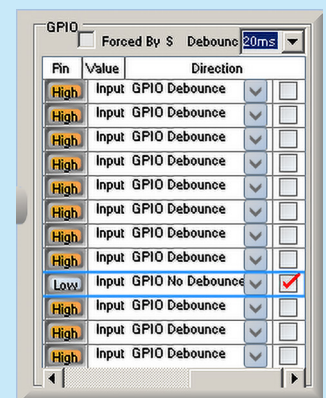
Wejścia cyfrowe mogą zostać sprzętowo zanegowane, niestety na niektórych komputerach (w tym i na moim) pojawiają się problemy z prawidłowym wyświetlaniem zakładki z rysunku 4. Dostęp do nieopisanej w menu funkcji negacji wymaga przesunięcia paska przewijania w prawo i zaznaczenia odpowiedniego pola – **rysunek 5**.

Stan wejść GPIO może być na bieżąco sprawdzany w oknie konfiguracji kontrolką **Value** dla wejść/wyjść cyfrowych oraz odcytem **Adc0...3** przy zmianie poziomu napięcia dla wejść ADC.

Przy obsłudze DSP czeka nas jeszcze jedna niespodzianka związana z wyprowadzeniem MP8. Z niewyjaśnionych do dzisiaj przyczyn, nie działa na nim eliminator drgań styku. Wybór opcji **Input GPIO Debounce**, kończy się odcytem stałego poziomu niskiego MP8, pozostałe funkcje działają poprawnie. Jeżeli aplikacja wymaga eliminacji drgań, przy projekcie sprzętu należy wykorzystać inne



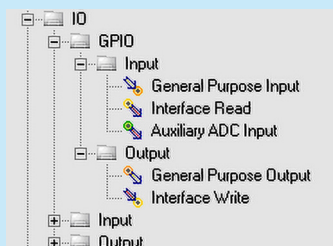
Rysunek 4. Konfiguracja sprzętowa GPIO



Rysunek 5. Negacja wejścia cyfrowego

wyprowadzenie MPx lub blok programowy *Debounce*.

Po konfiguracji sprzętowej, w obszar roboczy schematu, wyprowadzenia GPIO pobiera się z biblioteki *IO\GPIO*, wejścia/wyjścia cyfrowe jako *General Purpose Input\Output*, wejścia ADC jako *Auxiliary ADC Input* (rysunek 6).



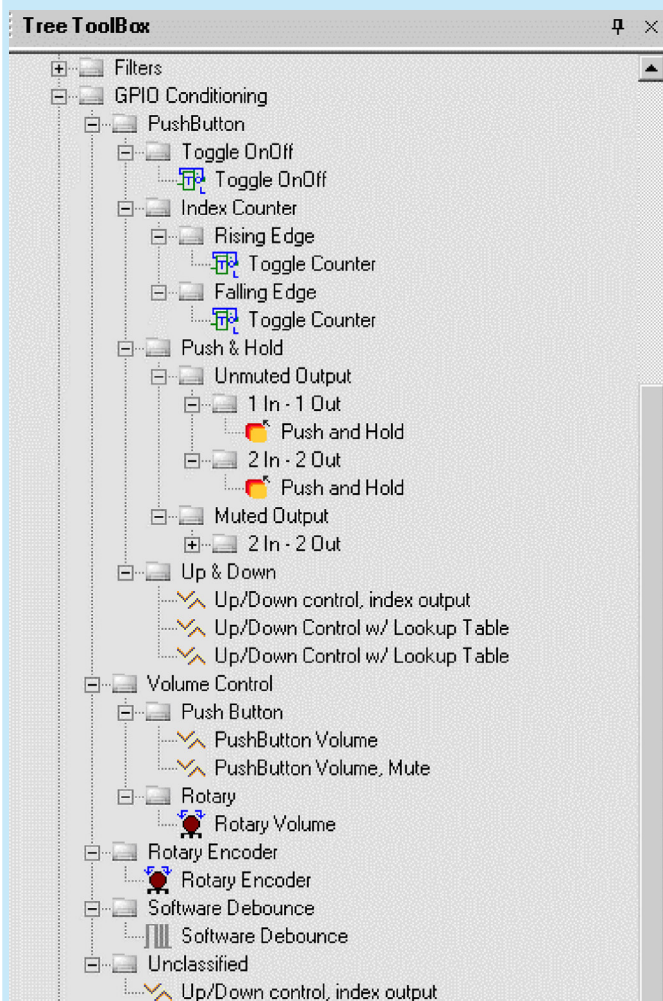
Rysunek 6. Biblioteka IO

W aplikacji do obsługi GPIO dedykowane są biblioteki *GPIO Conditioning* (rysunek 7).

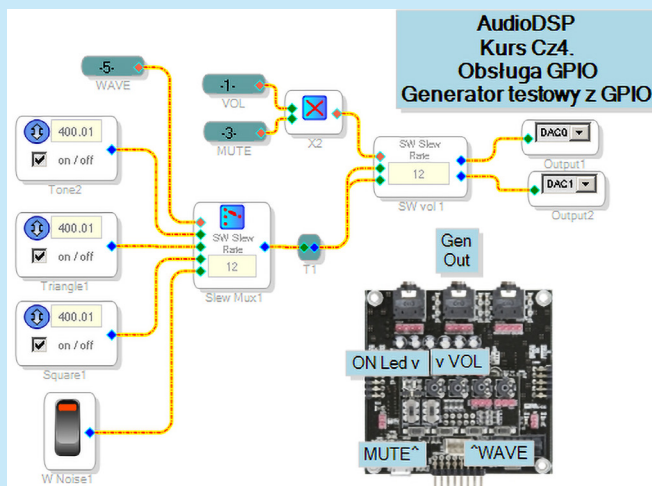
Biblioteka zawiera gotowe elementy do obsługi przycisków, enkoderów w najczęściej wykorzystanych zastosowaniach, np.: regulacji poziomu sygnału, obsługi przycisków, zliczania naciśnień. Sterowanie funkcjami z GPIO wymaga odpowiedniego połączenia bloków sterujących i regulacyjnych. Wyprowadzenie bloku odpowiedzialne za sygnał sterujący wyróżnione jest kolorem czerwonym i nie należy go łączyć bezpośrednio z wyprowadzeniami sygnałów audio (zielony – wejście, niebieski – wyjście) gdyż w większości przypadków nie zapewni to poprawnego działania.

Dla sprawdzenia obsługi w praktyce przygotowałem trzy aplikacje testowe wykorzystujące elementy bibliotek GPIO. Pierwszą jest sterowany generator sygnału *GeneratorGPIO.dspproj* wykorzystujący dwa zewnętrzne przyciski do aktywacji i przełączania kształtu przebiegu, potencjometr do regulacji poziomu oraz LED sygnalizująca działanie generatora. Schemat generatora przedstawia rysunek 8.

Generator wykorzystuje następujące bloki biblioteczne:



Rysunek 7. Biblioteka GPIO Conditioning



Rysunek 8. Schemat generatora *GeneratorGPIO.dspproj*

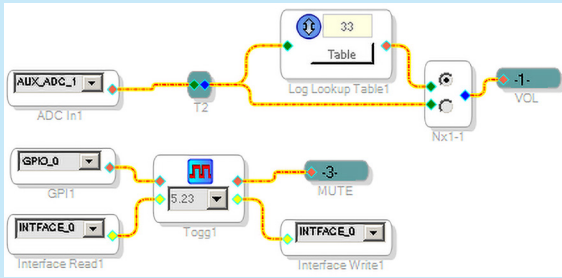
z biblioteki *Sources\Oscillators* generatory:

- sinusoidy *Sine Tone*,
- przebiegu trójkątnego *Triangle wave*
- prostokątnego *Square wave*,
- z biblioteki *Sources\White Noise\Simple*:
- źródło białego szumu *White Noise*,
- z biblioteki *Muxes\Multiplexers\Index Selectable\Clickless SW Slew\Mono*:
- multiplexer *Index Selectable Slewing Mux*,
- z biblioteki *Volume Controls\AdjustableGain\Ext Control\Clickless SW Slew* :
- regulator poziomu *Single slew ext vol*,
- z biblioteki *Basic DSP\Logic\Arithmetic Operations*:
- mnożarkę *Multiply*,
- z biblioteki *IO\Output*
- wyjście sygnału audio *Output*.

Bloki przeciągamy w obszar roboczy, zwiększamy liczbę wejść multiplexera i regulatora poziomu, i łączymy zgodnie z rysunkiem 8. Zasada działania części generacyjnej jest prosta. Generator ma cztery źródła sygnału (sinus, trójkąt, prostokąt, szum), które przełączane są multiplexerem sterowanym sygnałem z GPIO. Wybrany sygnał trafia do bloku regulacji głośności sterowanego sygnałem z GPIO i stąd na wyjścia. Każdy z bloków sterowanych sygnałem zewnętrznym wymaga sprawdzenia zakresu i typu zmiennej sterującej, opis dostępny jest w pomocy. Po sprawdzeniu formatu należy w przeważającej ilości przypadków opracować element pośredni odpowiadający za prawidłowe sterowanie z GPIO.

Procesor ADAU1701 używa znormalizowanego formatu liczb A, B gdzie A – liczba bitów części całkowitej po lewej stronie przecinka, B – liczba bitów części ułamkowej po prawej stronie (ten przecinek ze względu na amerykańskie pochodzenie DSP jest kropką... warto o tym pamiętać, aby ustrzec się nieprzewidywalnego działania aplikacji). Wymiana danych z rdzeniem DSP jest 24-bitowa, 4 dodatkowe bity są dodawane dla zwiększenia marginesu przesterowania (headroom), a kolejne 4 dla utworzenia pełnych 4 bajtów (32 bit). W DSP 28 bitów może być wykorzystane w różny sposób, np.: sygnał audio jest praktycznie zawsze reprezentowany w formacie dziesiętnym 5.23 (kilka bloków wykorzystuje okrojony do 5.19), a sygnały sterujące w formacie całkowitym 28.0. Wprowadza to pewne zamieszanie i wymaga każdorazowej analizy i odpowiedniego przeliczenia, na szczęście przy wsparciu bloków bibliotecznych.

Przykładowa reprezentacja liczb całkowitych dodatnich (ujemne są przedstawiane w kodzie U2, ale DSP i Sigma Studio ich nie wykorzystuje) w formacie 28.0 odpowiadającym reprezentacji binarnej: $0 = 0, 1 = 1, 10 = 2, 100 = 4$, itd. po uzupełnieniu zerami do 4 pełnych bajtów mamy:



Rysunek 9. Obsługa przycisku MUTE i potencjometru VOL

```
0000 0000 0000 0000 0000 0000 0000 0000 = 0
0000 0000 0000 0000 0000 0000 0000 0001 = 1
0000 0000 0000 0000 0000 0000 0000 0010 = 2
0000 0000 0000 0000 0000 0000 0000 0100 = 4
...
0000 0000 1000 0000 0000 0000 0000 0000 = 8388608
(poziom odniesienia pełnej skali ADC/DAC 0dB0 w formacie 28.0)
...
0000 0111 1111 1111 1111 1111 1111 1111 = 134217727
(2^27-1)
(maksymalna liczba dodatnia 28.0)
```

Sygnal audio w przeciwieństwie do sygnałów sterujących reprezentowany jest liczbą dziesiętną 5.23, pełna skala 0 dBfs odpowiada wartości 1, tłumienie -6 db wartości 0.5 itd. Podobnie jak w przypadku formatu 28.0 dodawane są 4 dodatkowe bity dla zwiększenia marginesu przesterowania, a kolejne 4 dla utworzenia pełnych 4 bajtów (32 bit).

Przykładowa reprezentacja liczb dziesiętnych (ujemne są przedstawiane w kodzie U2) w formacie 5.23 odpowiadające reprezentacji binarnej:

```
0 = 0, 1 = 1, 10 = 2, 100 = 4, itd.
dla liczb dodatnich:
0000 0000 0000 0000 0000 0000 0000 0000 = 0.0
0000 0000 0100 0000 0000 0000 0000 0000 = 0.5
0000 0000 1000 0000 0000 0000 0000 0000 = 1.0 (0 dBfs)
0000 0010 0000 0000 0000 0000 0000 0000 = 4.0
0000 0111 1111 1111 1111 1111 1111 1111 = (16.0 -1 LSB)
dla liczb ujemnych (U2):
1111 1000 0000 0000 0000 0000 0000 0000 = -16.0
1111 1110 0000 0000 0000 0000 0000 0000 = -4.0
1111 1111 1000 0000 0000 0000 0000 0000 = -1.0
1111 1111 1100 0000 0000 0000 0000 0000 = -0.5
1111 1111 1111 1111 1111 1111 1111 1111 = (1 LSB poniżej 0.0)
```

W przypadku zapisu liczby w formacie 5.23 do portów szeregowych audio lub bezpośrednio do przetwornika DAC liczba zostanie obcięta do wartości 0 dBfs.

Gdy konieczne jest przeliczenie pomiędzy formatami, można wykorzystać wzory:

$$n_{5.23} = n_{28.0} \cdot 2^{-23}, n_{28.0} = n_{5.23} \cdot 2^{23}$$

Po tych drobnych kwestiach formalnych wracamy do przykładu. Sygnały sterujące generatorem przygotowane są oddzielnie dla każdego bloku. Przycisk chwilowy MUTE podłączony do GPIO0 (rysunek 9) wymaga zastosowania bloku:

z biblioteki **GPIO Conditioning\Push Button**:

– przerzutnika **Toggle OnOff**,

zliczającego narastające zbocza sygnału z wejścia, konwertujący przycisk chwilowy do funkcji przycisku stabilnego, którego stan wyjściowy zmieniający jest kolejnymi naciśnięciami (zliczonymi zboczami), a format określony jest konfiguracją (będzie przełączany 23 bit cyklicznie na 0/1, przy pozostałych zawsze równych 0). Dodatkowe porty **Interface Read/Write** pobieramy:

z biblioteki **IO\GPIO\Input**

– odczyt z rejestrów **Interface Read**,

z biblioteki **IO\GPIO\Output**

– zapis do rejestrów **Interface Write**

Bloki **Interface Read/Write** służą do manipulacji na rejestrach DSP dostępnych dla użytkownika. Wartości są zapamiętywane i przywracane z pamięci E2PROM po włączeniu zasilania. Rejestrów użytkownika jest osiem, stan ich można podglądać w oknie **Hardware Configuration** (rysunek 10). Rejestry nie są w oknie odświeżane automatycznie, każdorazowo należy przyciskiem **Read All** odczytać ich stan.

Sygnal sterujący potencjometrem SW Vol 1 pobierany jest z wejścia GPIO1, skonfigurowanego jako przetwornik ADC. Dla sprawdzenia sposobu, w jaki można kształtować charakterystykę potencjometru, w tor włączona jest tablica LUT i multiplexer:

z biblioteki **Level Detectors\Lookup Tables\Lookup tables\Log table\Lower Range (-90 to +6 dB)**:

– tablica logarytmiczna LUT **Log Lookup Table**,

z biblioteki **Muxes/Demuxes\Multiplexers\Switch\Clickless SWSlew\1CH**:

– multiplexer **Mono Switch Nx1**

Multiplexerem podczas uruchamiania możemy wybrać charakterystykę regulacji skorygowaną tablicą LUT lub linową bezpośrednio z ADC GPIO. Przykładową zawartość tablicy przedstawia rysunek 11. Jest to tablica odpowiadająca krokowi regulacji od -90 dB/+6 dB

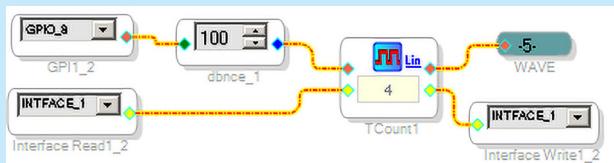
z skokiem 3 dB (plik Log.txt). Blok interpoluje wartości pomiędzy krokami regulacji, przy maksymalnym zakresie sygnału wejściowego 2.00 (+6 dB). Gdy potrzebna jest tablica o szerszym zakresie, możemy wymnożyć sygnał wyjściowy lub skorzystać z tablicy bez ograniczenia **Level Detectors\Lookup Tables\Lookup tables\Log table\Log Lookup Table**, w której sygnał wejściowy może wynosić 15,999. Przygotowane sygnały VOL i MUTE doprowadzone są do mnożarki **Multiply**. Wymnożenie przez 0 wycisza sygnał wyjściowy sterujący potencjometrem. Taka nieco okrężna metoda wyciszania wynika z braku odpowiedniego bloku funkcjonalnego (wyciszanie sygnałem zewnętrznym). Realizacja wyciszania jest możliwa także w oparciu np. o multiplexery, ale warto wtedy sprawdzić, czy przełączanie nie odbywa się ze stukiem, może być to niedopuszczalne w aplikacji. Po odpowiedniej parametryzacji układ można

Interface Register	
Force By SPI	<input type="checkbox"/>
Reg0	0x00000000
Reg1	0x00000002
Reg2	0x00000000
Reg3	0x00000000
Reg4	0x00000000
Reg5	0x00000000
Reg6	0x00000000
Reg7	0x00000000

Rysunek 10. Rejestry użytkownika

Index	Value
1	1.58E-05
2	2.24E-05
3	3.16E-05
4	4.47E-05
5	6.31E-05
6	8.91E-05
7	0.000126
8	0.000178
9	0.000251
10	0.000355
11	0.000501
12	0.000708
13	0.001
14	0.0014
15	0.002
16	0.0028
17	0.004
18	0.0056
19	0.0079
20	0.0112
21	0.0158
22	0.0224
23	0.0316
24	0.0447
25	0.0631
26	0.0891
27	0.1259
28	0.1778
29	0.2512
30	0.3548
31	0.5012
32	0.7079
33	1

Rysunek 11. Zawartość tablicy LUT (Log.txt)



Rysunek 12. Sterowanie multiplexerem

wykorzystać także do skokowej zmiany wzmacnienia o ustaloną wartość, pamiętając o ryzyku przesterowania wyjścia mnożarki.

Sterowanie multiplexerem przełączającym kształt przebiegu (rysunek 12), odbywa się za pomocą dedykowanego bloku licznika: z biblioteki **GPIO Conditioning\Push Button\Index Counter\ Rising Edge**:

- licznik **Toggle Counter**,
- i pomocniczego eliminatora drgań styków z biblioteki **GPIO Conditioning\Software Debounce**:
- eliminator drgań **Software Debounce**.

Sygnal z przycisku podłączonego do GPIO8 podawany jest na blok programowego eliminatora drgań styków z ustawionym czasem 100 ms, a następnie na blok licznika impulsów (po 4 impulsie następuje reset, zliczanie jest liniowe 0-1-2-3-0...). Bloki **Interface** podobnie jak poprzednio zachowują i przywracają wartości zliczania po włączeniu zasilania. Sygnal z licznika doprowadzony jest do wejścia wybierającego multiplexera, gdzie odpowiada za wybór odpowiadającego stanowi licznika wejścia.

Sterowanie diodą LED sygnalizującą działanie aplikacji przedstawia **rysunek 13**. Układ składa się z generatora impulsów o regulowanym wypełnieniu, komparatora wartości zerowej i bramki AND: z biblioteki **Sources\Oscillators**:

- generator impulsów **Pulse with dynamic duty cycle**,
- z biblioteki **Basic DSP\Logic\Invert**:
- komparator zera **Zero Comparator**,
- z biblioteki **Basic DSP\Logic**:
- bramka logiczna And **Logic - And, Or, Nand, Nor**.

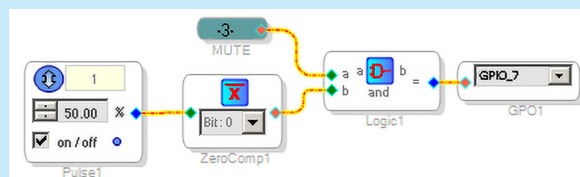
Częstotliwość generatora ustawiona jest na 1 Hz, wypełnienie na 50%. Sygnal wyjściowy doprowadzony jest do komparatora wartości 0 (z zanegowanym wyjściem), na wyjściu którego otrzymujemy sygnal 0 (na ustawionym bicie 0), gdy sygnal wejściowy jest różny od zera. Następnie sygnal jest podany na wejście bramki logicznej AND wraz z sygnałem MUTE, a stąd do wyjścia. Funkcja bramki wybierana jest poprzez naciśnięcie ikony na bloku, każdorazowa zmiana wymaga kompilacji i załadowania programu do DSP. Z bramki sygnalysterowuje wyjście GPIO7 i podłączoną do niego LED. Jeżeli wyjście generatora nie jest wyciszone, LED sygnalizuje pracę generatora miganiem 1 Hz, jeżeli wyciszanie jest aktywne, pozostaje zgaszona.

Drugą aplikacją jest filtr pasmowy o regulowanej szerokości przenoszonego pasma. **BPF.dspproj** wykorzystuje przyciski do regulacji poziomu sygnału, wyciszania i ustalania szerokości pasma filtra. Schemat blokowy generatora przedstawia **rysunek 14**. Projekt wykorzystuje schemat hierarchiczny. Multiplexery sygnału oraz źródło szumu używane są podczas uruchamiania aplikacji.

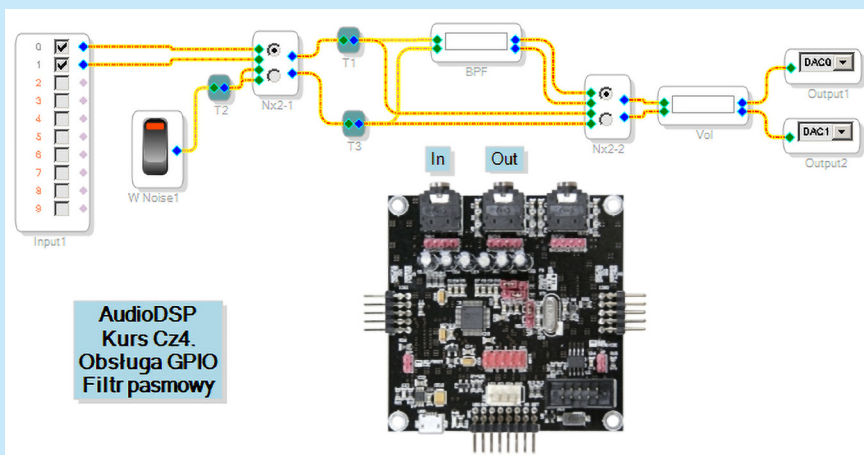
- Filtr wykorzystuje następujące bloki biblioteczne:
- z biblioteki **GPIO Conditioning\Volume Control\Push Button** - regulator głośności z funkcją wyciszania **PushButton Volume, Mute**,
 - z biblioteki **GPIO Conditioning\Push Button\Up & Down** - układ kontroli przycisku +/- **Up&Down Control, index output**,
 - z biblioteki **Filters\Second Order\Lookup\Double Precision** - konfigurowalny filtr sterowany indeksem **General 2nd Order w var Param\Lookup\Slew**,
 - z biblioteki **Sources\White Noise\Simple** - źródło szumu białego **White Noise**,
 - z biblioteki **Muxes\Demuxes\Multiplexers\Switch\Clickless SW Slew\1Ch**
 - multiplexer **Mono Switch Nx1**,
 - z biblioteki **IO\Output**:
 - wyjścia sygnału audio **Output**,
 - z biblioteki **IO\Input**
 - wejścia sygnału audio **Input**.

Sygnal z wejść analogowych podawany jest na multiplexer, gdzie możliwe jest przełączenie na sygnal testowy białego szumu. Z multiplexera podawany jest kolejno na blok filtra BPF **rysunek 15** i przez multiplexer umożliwiający pominięcie filtra na blok regulacji poziomu Vol (**rysunek 16**), i stąd doprowadzony jest do wyjścia.

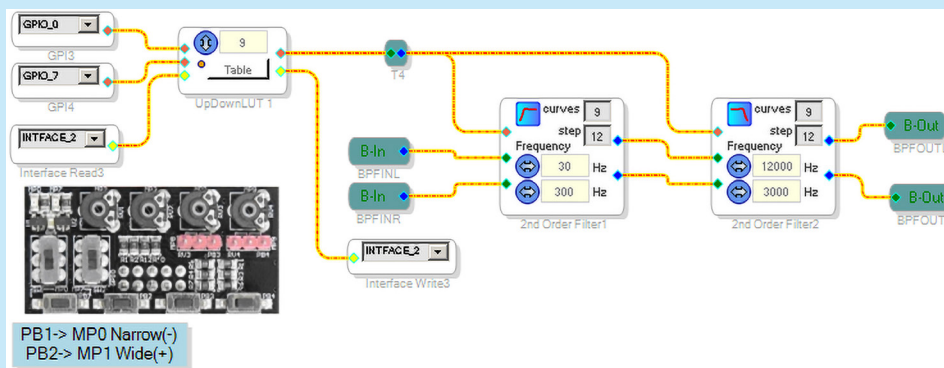
Filtr oparty jest na konfigurowalnym bloku **General 2nd Order w var Param\Lookup\Slew** umożliwiający realizację podstawowych



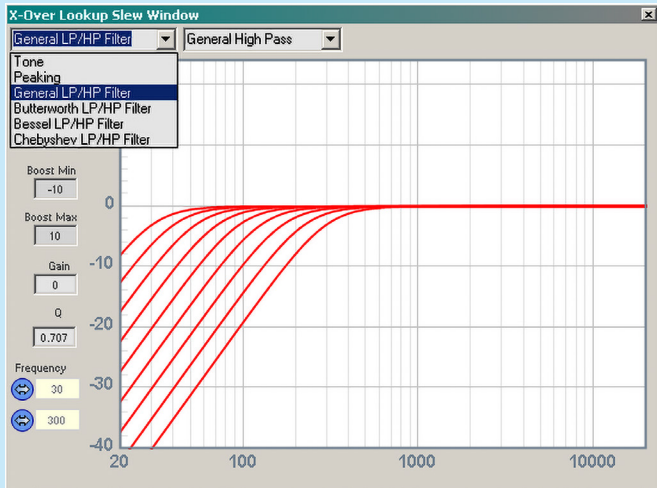
Rysunek 13. Sterowanie LED



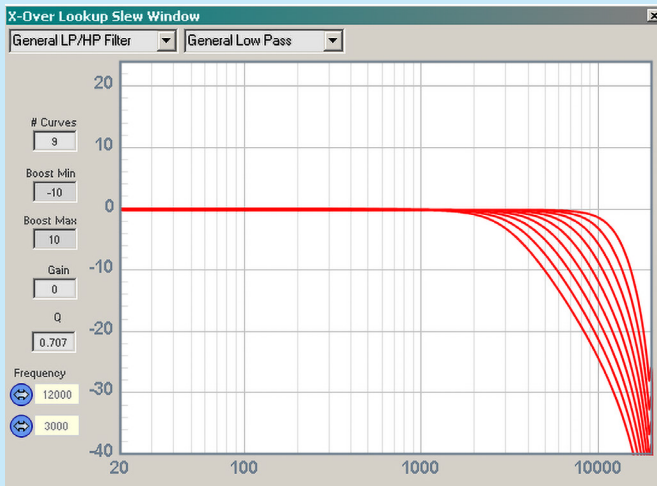
Rysunek 14. Filtr pasmowy BPF.dspproj



Rysunek 15. Schemat filtru pasmowego BPF



Rysunek 16. Konfiguracja filtra górnoprzepustowego

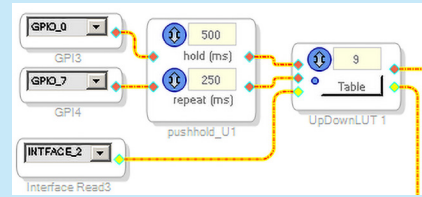


Rysunek 17. Konfiguracja filtra dolnoprzepustowego

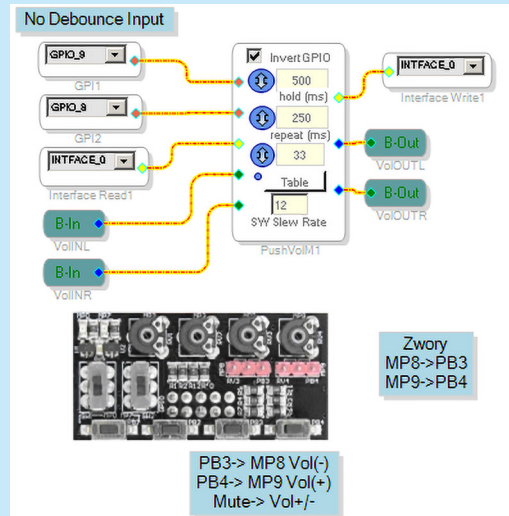
filtrów sygnału dolno-, górnoprzepustowych o różnych charakterystykach, filtrów pasmowych o dużej dobroci Peak oraz regulatorów barwy tonu. W modelu, dwa filtry, górno- (rysunek 17) i dolno-przepustowy (rysunek 17) połączone są w szereg, dla otrzymania wypadkowej charakterystyki pasmowej o regulowanej szerokości pasma przenoszenia. Sterowanie filtra odbywa się poprzez wejście indeksujące, wartość indeksu wybiera odpowiadający numer charakterystyki. Indeksy muszą mieścić się w zakresie zmniejszonym o jeden w stosunku do liczby ustalonych krzywych. Przekroczenie indeksu może spowodować nieprawidłowe działanie DSP. Do wyboru charakterystyki filtra użyte w projekcie są dwa przyciski PB1(Narrow) zmniejszający szerokość pasma, PB2(Wide) zwiększający szerokość pasma. Sygnały z GPIO doprowadzone są do bloku **Up&Down Control, index output**. Blok realizuje kilka funkcji związanych z obsługą GPIO. Zawiera skonfigurowany licznik rewersyjny sterowany przyciskami +/- (Up/Down), tablicę LUT umożliwiającą przeliczenie stanu licznika na wartość sterującą wraz z jej ograniczeniem i interfejs pozwalający zapamiętać stan indeksu w pamięci nieulotnej. Wartości zapisane w tablicy przez użytkownika zależą od rodzaju sterowanego bloku, dla filtra są to numery

Index	Value
1	0
2	1.19209E-07
3	2.38419E-07
4	3.57628E-07
5	4.76837E-07
6	5.96046E-07
7	7.15256E-07
8	8.34465E-07
9	9.53674E-07

Rysunek 18. Zawartość tablicy sterującej charakterystyką BPF (BPF.txt)



Rysunek 19. Podłączenie bloku Push And Hold



Rysunek 20. Regulator poziomu Vol

kolejnych charakterystyk, licząc od 0. Zawartość tablicy pokazuje rysunek 18. Oba filtry sterowane są jednym indeksem, odpowiednie odwrócenie charakterystyk filtra dolnoprzepustowego umożliwia jednocześnie sterowanie szerokością pasma. W projekcie założono 9 kroków regulacji, nie jest więc problemem wybór przez kilkakrotne naciśnięcie przycisku. Jeżeli liczba kroków musi być większa, np. przy regulacji barwy tonu lub głośności, warto zastosować blok **Push and Hold** z biblioteki **GPIO Conditioning\Push Button\Unmuted Output\2 In - 2 Out** (rozszerzając też oczywiście tablice wartości LUT). Realizuje on opcję automatycznego powtarzania naciśnięcia klawisza w ustawionym interwale czasowym (rysunek 19). Parametr **Hold** określa, jak długo przycisk musi być naciśnięty przed rozpoczęciem automatycznego powtarzania, parametr **Repeat** określa czas pomiędzy kolejnymi powtórzeniami.

Z bloku BPF sygnał doprowadzony jest do regulatora poziomu Vol (rysunek 20). Regulacja odbywa się poprzez sprawdzenie stanu przycisków PB3(-), PB4(+), jednocześnie naciśnięcie przycisków wycisza sygnał. Do realizacji funkcji jest użyty blok **PushButton Volume, Mute** integrujący w sobie układy

Index	Value
1	0
2	0.0313
3	0.0625
4	0.0938
5	0.125
6	0.1563
7	0.1875
8	0.2188
9	0.25
10	0.2813
11	0.3125
12	0.3438
13	0.375
14	0.4063
15	0.4375
16	0.4688
17	0.5
18	0.5313
19	0.5625
20	0.5938
21	0.625
22	0.6563
23	0.6875
24	0.7188
25	0.75
26	0.7813
27	0.8125
28	0.8438
29	0.875
30	0.9063
31	0.9375
32	0.9688
33	1

Rysunek 21. Tablica Vol (Vol.txt)

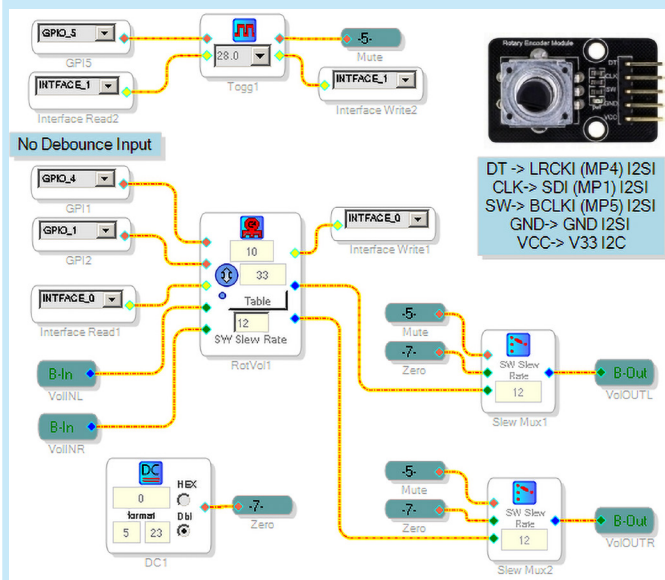
licznika dwukierunkowego, bloku Push & Hold, tablicy indeksów, regulator(y) poziomu oraz interfejs pozwalający zapamiętać stan indeksu w pamięci nieulotnej. Parametry **Hold** i **Repeat** definiowane są identycznie jak poprzednio, możliwe jest zanegowanie wejść sterujących, tablica określająca charakterystykę regulacji przedstawiona jest na rysunku 21.

Projekt po przetestowaniu, usunięciu elementów pomocniczych (źródło szumu i multiplexery) oraz zapisaniu do pamięci nieulotnej może służyć jako moduł rozszerzający funkcje amatorskiego odbiornika komunikacyjnego, kształtując charakterystykę odbiorczego toru audio w zależności od jakości odbioru.

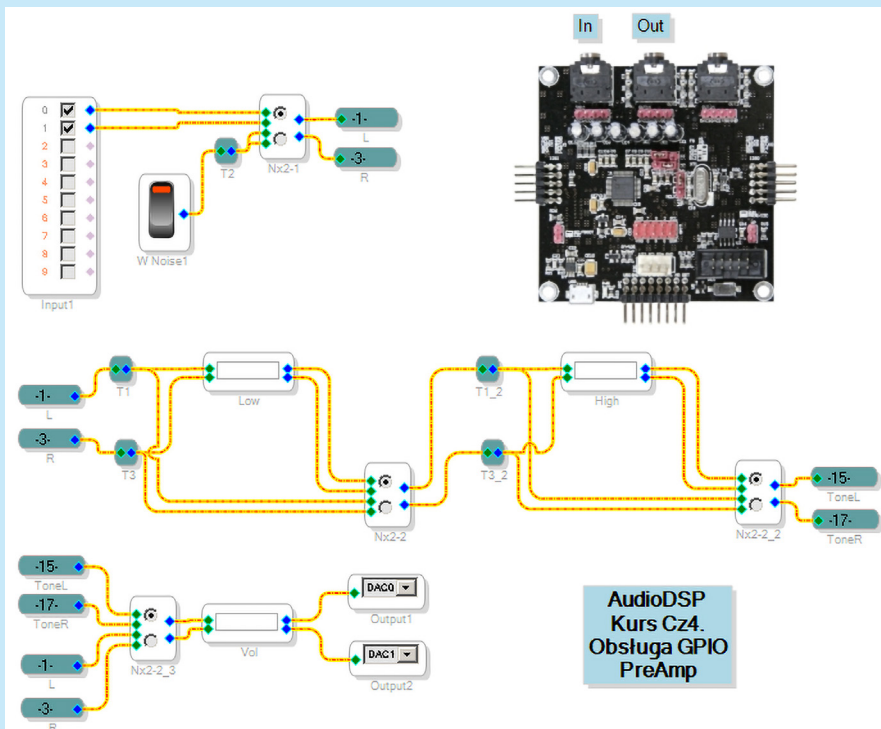
Trzecim projektem jest przedwzmacniacz **PreAmp.dspproj**, którego schemat blokowy przedstawia rysunek 22.

Projekt wykorzystuje enkoder obrotowy z wbudowanym przyciskiem do regulacji poziomu sygnału i wyciszania oraz przyciski chwilowe do regulacji barwy dźwięku. Układ składa się z trzech bloków funkcjonalnych, regulatora tonów niskich, wysokich oraz regulatora poziomu sygnału. Podobnie jak w poprzednim projekcie układ uzupełniają multiplexery i źródło szumu ułatwiające testowanie aplikacji. W aplikacji wykorzystane są bloki:

- z biblioteki GPIO **Conditioning\Volume Control\Rotary**
 - regulator poziomu sterowany enkderem **Rotary Volume**,
- z biblioteki GPIO **Conditioning\Push Button\Up & Down**
 - kontroler przycisku +/- z tablicą LUT **Up&Down Control w Lookup Table**,
- z biblioteki GPIO **Conditioning\Push Button\Toggle OnOff**
 - przerzutnik **Toggle OnOff**,
- z biblioteki **Filters\Second Order\Lookup\Double Precision**
 - konfigurowalny filtr sterowany indeksem **General 2nd Order w var Param\Lookup\Slew**,
- z biblioteki **Sources\White Noise\Simple**
 - źródło białego szumu **White Noise**,
- z biblioteki **Sources\DC**
 - stałą numeryczną **DC Input Entry**,



Rysunek 23. Regulator poziomu Vol



Rysunek 22. Schemat projektu przedwzmacniacza **PreAmp.dspproj**

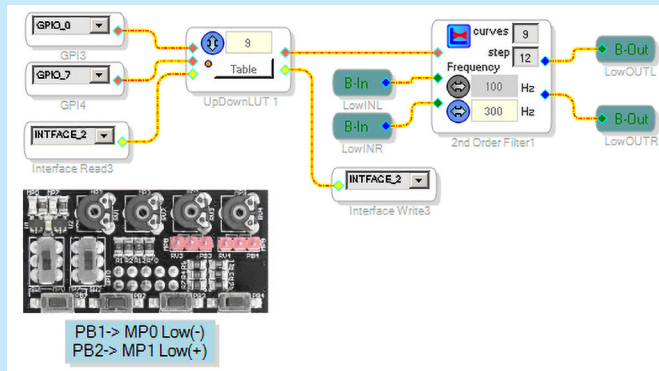
- z biblioteki **Muxes/Demuxes\Multiplexers\Switch**
 - Clickless SW Slew\1Ch**
 - multiplexer **Mono Switch Nx1**,
- z biblioteki **Muxes/Demuxes\Multiplexers\Index Selectable\Clickless SW Slew\Mono**
 - multiplexer sterowany indeksem **Index Selectable Slewling Mux**,
- z biblioteki **IO\Output**:
 - wyjścia sygnału audio **Output**,
- z biblioteki **IO\Input**:
 - wejścia sygnału audio **Input**.

Regulator poziomu Vol, którego schemat przedstawia rysunek 23, wykorzystuje gotowy blok funkcjonalny obsługujący enkoder obrotowy **Rotary Volume**. Parametryzacji podlegają: czas eliminacji drgań styków **Debounce** (10), liczba kroków tablicy regulacji (33), szybkość i płynność przełączenia pomiędzy krokami **SW Slew Rate** (12). Blok umożliwia zapis parametrów do pamięci nieulotnej. Zawartość tablicy prezentuje rysunek 24. Przy konfiguracji wejść GPIO podłączonych do bloku **Rotary Volume** należy wyłączyć domyślną eliminację drgań **Input No Debounce**, aby nie zakłócać działania jego funkcji.

Enkoder fizycznie podłączony jest do złącza I2SI, sygnał DT(A) do pinu MP4-LRCKI, CLK(B) do MP1-SDI, SW(PB) do MP5-BCLKI, GND do GND, a VCC do złącza I2C potencjał V33 (**pomyłkowe podłączenie do V50 uszkodzi DSP, trzeba więc uważać...**). Jeżeli regulacja

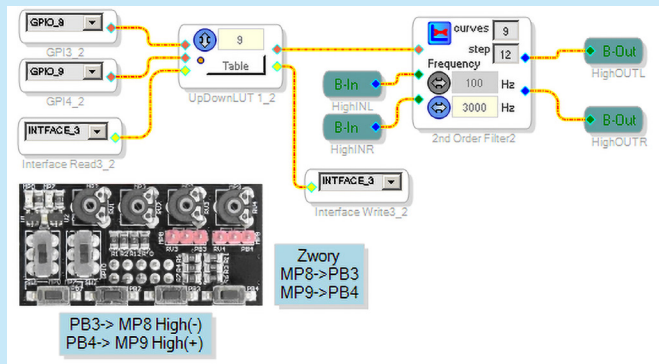
Table Editor -	
Save table values	Load table values
1	0
2	0.0313
3	0.0625
4	0.0938
5	0.125
6	0.1563
7	0.1875
8	0.2188
9	0.25
10	0.2813
11	0.3125
12	0.3438
13	0.375
14	0.4063
15	0.4375
16	0.4688
17	0.5
18	0.5313
19	0.5625
20	0.5938
21	0.625
22	0.6563
23	0.6875
24	0.7188
25	0.75
26	0.7813
27	0.8125
28	0.8438
29	0.875
30	0.9063
31	0.9375
32	0.9688
33	1

Rysunek 24. Tablica regulacji Vol (Enc.txt)



PB1-> MP0 Low(-)
PB2-> MP1 Low(+)

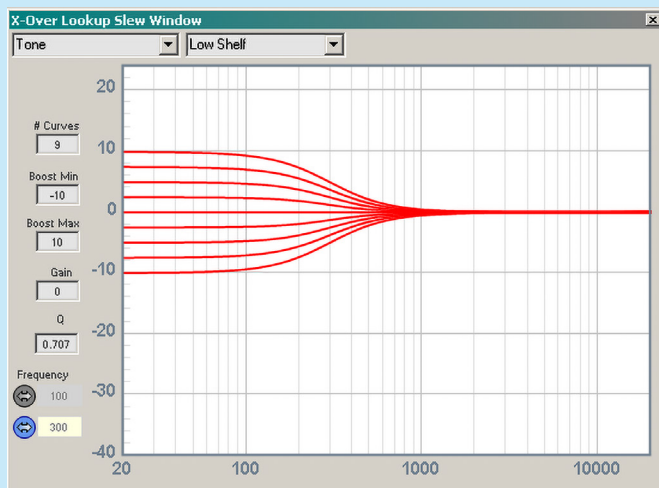
Rysunek 25. Regulator tonów niskich



Zwory
MP8->PB3
MP9->PB4

PB3-> MP8 High(-)
PB4-> MP9 High(+)

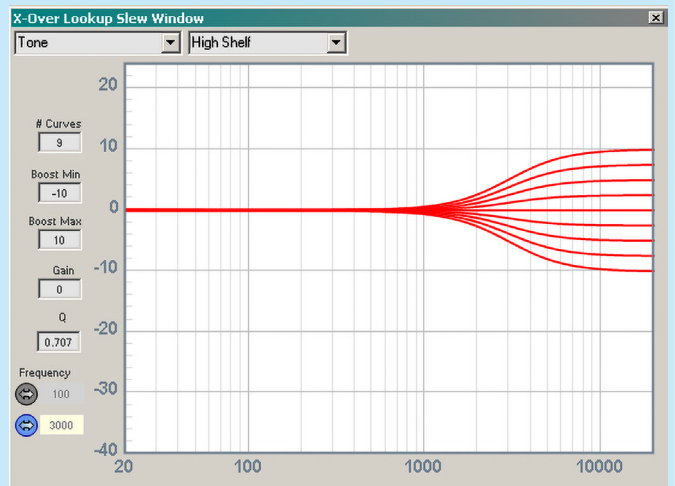
Rysunek 26. Regulator tonów wysokich



Rysunek 27. Charakterystyka: tony niskie

odbywa się w „złym” kierunku, należy zamienić wejścia A/B sygnału enkodera.

Blok **Rotary Volume** nie obsługuje funkcji wyciszania, którą należy zrealizować we własnym zakresie. Tym razem, aby było bardziej „dydaktycznie” nie powielię sprawdzonego wcześniej rozwiązania, polegającego na zmianie sygnału sterującego regulatorem poziomu (blok enkodera uniemożliwia jej wykorzystanie), a sygnał wyciszany będzie poprzez sterowanie multiplexerem. Przycisk SW(PB) enkodera doprowadzony jest do bloku Toogle, zmieniającego stan po każdym zboczu narastającym. Sygnał ten steruje przełączaniem wejść dwukanałowych multiplexerów **SlewMux**. Gdy wyciszanie



Rysunek 28. Charakterystyka: tony wysokie

Table Editor -	
Save table values	Load table values
1	0
2	1.19209E-07
3	2.38419E-07
4	3.57628E-07
5	4.76837E-07
6	5.96046E-07
7	7.15256E-07
8	8.34465E-07
9	9.53674E-07

Rysunek 29. Tablica regulacji tonów (Tone.txt)

jest aktywne, na wyjście regulatora podawane jest cyfrowe „zero” z bloku stałej DC1, gdy wyciszanie jest wyłączone, na wyjście podawany jest sygnał audio z bloku obsługującego enkoder. Ważny przy takim sposobie regulacji jest wybór elementów bibliotecznych z wbudowaną funkcją **Slew**, zapewniającą powolne narastanie sygnału i brak zakłóceń podczas zmiany stanu (niższa wartość Slew zapewnia szybszą reakcję na kolejne kroki regulacji).

Bloki regulacji barwy dźwięku działają na podobnej zasadzie jak w poprzednim projekcie, odmienna jest tylko konfiguracja filtra **rysunki 24...28**. Regulacja odbywa się poprzez sprawdzenie stanu przycisków PB1(Low-), PB2(Low+), PB3(High-), PB4(High+). Stan przycisków zapamiętywany jest w pamięci nieulotnej. Podczas uruchamiania aplikacji należy pamiętać o założeniu zwór aktywujących przyciski MP8, MP9 w pozycjach PB3, PB4. Tablicę regulacji przedstawia **rysunek 29**.

Podobnie jak we wcześniejszym przykładzie można w sterowaniu regulatorem barwy wykorzystać bloki **Push And Hold** do automatycznej zmiany indeksu przy przytrzymaniu przycisku sterującego. Tak przygotowany i sprawdzony projekt można po dostosowaniu krzywych regulacji do własnych upodobań wykorzystać w roli samodzielnego przedwzmacniacza audio z procesorem DSP.

W przedstawionych projektach w skrótowy sposób opisano użycie najczęściej stosowanych bloków obsługujących interfejs GPIO. W następnej części kursu zaprezentujemy zostanie sposób konfiguracji i wykorzystania szeregowego interfejsu cyfrowego. Tymczasem proszę załączone projekty modyfikować, przerabiać, psuć, naprawiać, aby nabrać wprawy przed własnymi samodzielnymi aplikacjami AudioDSP.

Zapraszam na kolejną część kursu...

Adam Tatus
adam.tatus@ep.com.pl

REKLAMA

www.facebook.com/ElektronikaPraktyczna