Teoria i przetwarzanie sygnałów

Projekt domofonu bezprzewodowego

Projekt wykonał:
Łukasz Uszko

Kierunek: Elektrotechnika

Spis treści:

1. **Wprowadzenie i historia pomysłu**
2. **Schemat ogólny urządzenia oraz krótki opis poszczególnych elementów użytych w układzie**
3. **Teoria związana z konstrukcją domofonu**
4. **Algorytm działania urządzenia**
5. **Problemy powstałe podczas konstrukcji urządzenia**
6. **Schemat ogólny urządzenia oraz płytki PCB**
7. **Wnioski końcowe**
8. **Literatura**

1. Wprowadzenie:

Tematem projektu było skonstruowanie domofonu (intercomu) bezprzewodowego. Pomysł budowy tego układu zrodził się w wyniku rosnącego zapotrzebowania na tego typu układy. Niestety dostępne na rynku urządzenia tego typu są w tej chwili drogie, więc nie każdy może sobie na nie pozwolić. Cena takich urządzeń zaczyna się w granicach 200zł. Moim celem było skonstruowanie domofonu którego wartość nie przekroczy 80zł. Jak wiadomo wraz z obniżeniem ceny, spada również jakość produktu. Dlatego połączenie odpowiedniej jakości z niską ceną okazało się rzeczą nie łatwą.

Urządzenie to będzie cały czas przeze mnie rozwijane w celu poprawy jakości, kolejnym krokiem jaki planuje jest przekształcenie urządzenia w wideodomofon (przesyłanie zarówno obrazu jak i dźwięku) a także budowa niewielkiej sieci by urządzenia mogły komunikować się ze sobą niezależnie na częstotliwościach pośrednich.

2. Schemat ogólny urządzenia oraz krótki opis poszczególnych elementów użytych w układzie

Jako że postawionym przeze mnie celem było zbudowanie urządzenia zarówno taniego jak i o dobrych parametrach technicznych dobranie odpowiednich komponentów nie było łatwe. Po prześledzeniu tego co na rynku dostępne postanowiłem swój układ oprzeć na następujących elementach:

a) Mikrokontroler STM32F051R8T6 z wydajnym 32-bitowym rdzeniem ARM Cortex-M0 wykonanym w technologii RISC, mogącym pracować z częstotliwością 48MHz.
Podstawowe cechy zastosowanego mikrokontrolera:
- 64kB pamięci FLASH i 8kB pamięci SRAM
- Standardowe interfejsy komunikacyjne: I^2C, SPI, I2S, USART, HDMI CEC
- 12-bitowy przetwornik ADC oraz 12-bitowy przetwornik DAC
- Pięć 16-bit timerów ogólnego przeznaczenia oraz jeden 32-bit timer z kontrolerem PWM
- 5-kanałowy kontroler DMA
- Wbudowany zegar RTC
- Wbudowany wewnętrzny oscylator 8MHz z możliwością mnożenia x6 za pomocą PLL

b) Moduł radiowy RFM 70
Jest to kompletny moduł radiowy zintegrowany z anteną, pracujący w paśmie 2,4 GHz, Moduły te mogą przesyłać dane z prędkością 1 lub 2 Mb/s, tor nadawczy wyposażono we wzmacniacz o programowalnej mocy wyjściowej w zakresie -40 dBm do 5 dBm. Wymiana danych pomiędzy modułem a otoczeniem odbywa się za pomocą 4-przewodowej magistrali SPI z maksymalną szybkością taktowania 8 MHz.

c) Wzmacniacz sygnału wyjściowego TDA2822
Jest to monolityczny wzmacniacz mocy mogący pracować przy napięciu już od 3V. Jego moc wyjściowa przy zasilaniu 5V i obciążeniu głośnikiem o impedancji 4Ω, wynosi 2W; Charakteryzuje się on stosunkowo niewielkim współczynnikiem THD przy zasilaniu niskim napięciem , a także niewielkimi całkowitymi stratami mocy.

d) Głośnik:
- moc 3W
- impedancja 4 om ± 15%
- częstotliwość przenoszenia 400 ~ 8000Hz
- wymiary magnesu Ø12 x 2mm (NdFeB)

- membrana papierowa

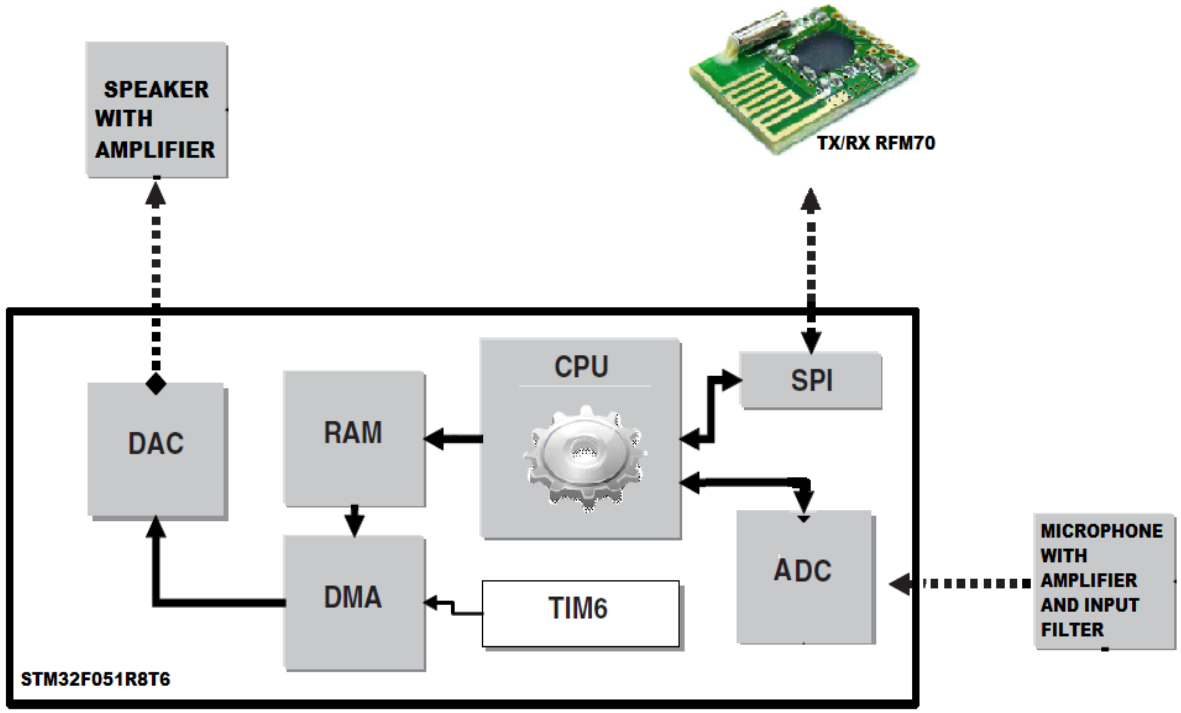
e) Mikrofon pojemnościowy:
   - impedancja 2.2k
   - napięcie 2V
   - wymiary: 6x5,2mm

f) Wzmacniacz wejściowy: układ zbudowany w oparciu o dwa wzmacniacze operacyjne mający na celu dopasowanie wartości sygnału z mikrofonu do odpowiedniej wartości podawanej na wejście przetwornika analogowo-cyfrowego.

g) Filtr wejściowy- antyaliasingowy:
   Filtr wejściowy ma zadanie eliminować częstotliwości nie występujące w widmie częstotliwości mowy ludzkiej. Jest to filtr dolno przepustowy 4-rzędu o ch-ce Bessela o fg=10kHz. Wzmocnienie filtra wynosi 0dB

h) Inne:
   - Rezystory ,
   - Kondensatory ,
   - Niskoszumny wzmacniacz operacyjny LM358
   - Stabilizator napięcia LM317T

Poniżej przedstawiony został schemat ogólny domofonu:



Rys. Schemat blokowy domofonu

Ogólna zasada działania urządzenia jest bardzo prosta. Po wciśnięciu przez użytkownika przycisku wywołania (zostaje wygenerowany sygnał dzwonkowy na odbiorniku) urządzenie uruchamia się. Mikrokontroler zostaje „wybudzony” ze stanu ograniczonego poboru mocy. Zostają uruchomione wszystkie moduły mikrokontrolera pokazane na rysunku powyżej oraz RFM70. Rozpoczyna się dwustronna transmisja dźwięku (mowy). Przetwornik ADC nadajnika pracujący z częstotliwością próbkowania 22kHz, zapełnia bufor wewnętrzny próbkami sygnału, Jeśli bufor jest zapełniony zostaje zgłoszone przerwanie obsługujące wysyłanie danych do odbiornika. Następnie podczas zapełniania kolejnego bufora próbkami , moduł radiowy zostaje przełączony w tryb odbioru, i nadajnik odbiera ramkę z próbkami od odbiornika , wysyłając poprzez kanał DMA natychmiast do przetwornika DAC.

Identyczna sytuacja ma miejsce po stronie odbiornika. Dzięki zastosowaniu kanału DMA odciążającego CPU mikrokontrolera transmisja przebiega płynnie i nie zaobserwowano dużego zniekształcenia sygnału spowodowanego opóźnieniami w przesyłaniu sygnału. Szczegóły w punkcie 4.

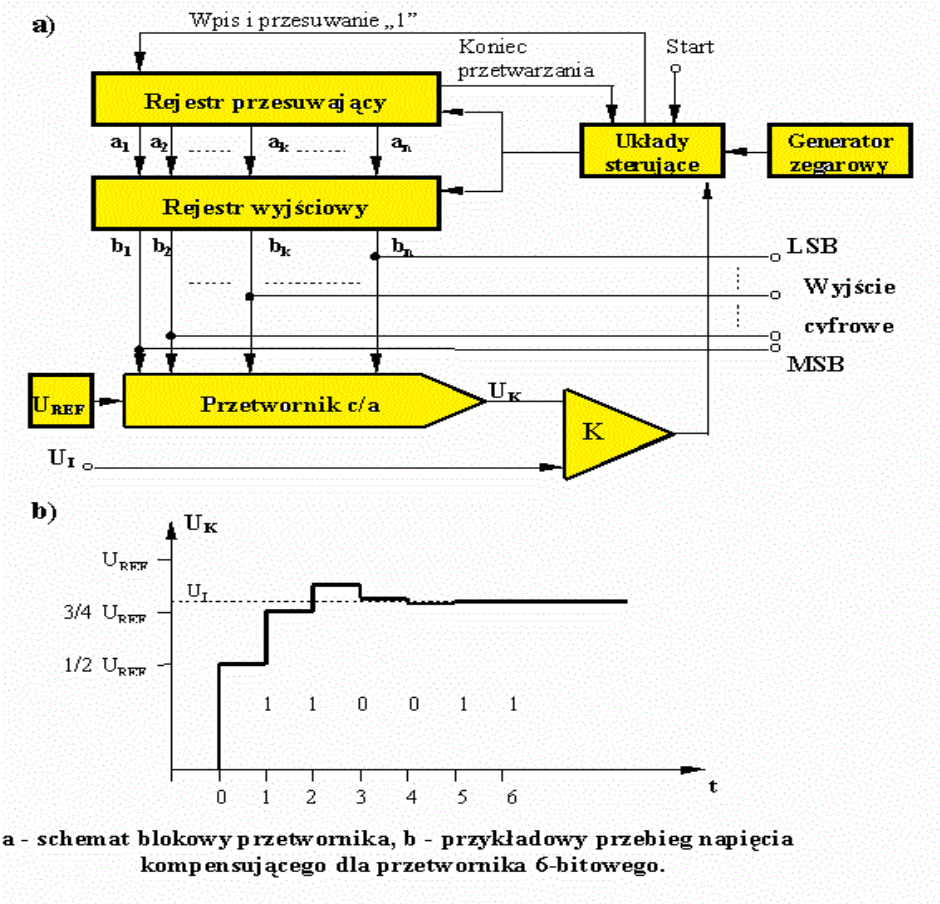
3. Teoria związana z konstrukcją domofonu

Poprawne działanie urządzenia związane jest z odpowiednim doborem układu pracy poszczególnych elementów. W punkcie tym zostaną szczegółowo opisane poszczególne podzespoły oraz dobór ich parametrów pracy, obliczenia z tym związane , a także analiza błędów układu .

a) Przetwornik Analogowo cyfrowy

Wykorzystany przeze mnie wbudowany przetwornik analogowo cyfrowy mikrokontrolera STM32F051R8T6 jest to 12- bitowy przetwornik typu SAR (tzw. sukcesywnej aproksymacji) Mogę śmiało stwierdzić iż przetwornik ADC jest bardzo ważnym elementem całego urządzenia. Od jego jakości , w dużym stopniu zależy jakość przesyłanego dźwięku.

- Opis, zasada działania oraz ogólna idea przetworników typu SAR (kompensacji wagowej).



a - schemat blokowy przetwornika, b - przykładowy przebieg napięcia kompensującego dla przetwornika 6-bitowego.

Typowy przetwornik ADC typu SAR składa się z 4 głównych pod obwodów:

- *Obwód wejściowy tzw. SAMPLE/HOLD* - służący do doprowadzenia analogowego napięcia mierzonego *Ui* do komparatora.
- *Analogowy komparator* - porównujący napięcie wejściowe *Ui* z napięciem wyjściowym z wbudowanego przetwornika DAC. Wynik porównania podawany jest do wejścia rejestru przesuwnego typu SAR.

- *Rejestr przesuwny sukcesywnej aproksymacji* - służący do wytworzenia danej w postaci cyfrowej dla wewnętrznego przetwornika DAC.
- *Wewnętrzny przetwornik DAC* - służący do wytworzenia napięcia dla komparatora, jako wartości odpowiadającej danej w postaci cyfrowej pobranej z wyjścia rejestru SAR.

Przetwarzanie polega na kolejnym porównywaniu napięcia przetwarzanego U_i z napięciem odniesienia U_r wytwarzanym w przetworniku c-a. W pierwszej kolejności następuje porównanie napięcia wejściowego z połową napięcia pełnego zakresu przetwarzania. Rezultat tego porównania ustala w rejestrze wartość cyfrową najstarszego bitu słowa wyjściowego oraz wartość najstarszego bitu wejścia przetwornika c-a. W przypadku przetwornika n- bitowego pełny cykl przetwarzania obejmuje n porównań. W każdym kolejnym kroku przetwarzania impuls z generatora zegarowego powoduje przesunięcie w rejestrze przesuwającym stanu logicznego „1" o jedno miejsce oraz w każdym kroku przetwarzania aktualna wartość napięcia kompensującego jest porównywana z napięciem wejściowym Uwe i w zależności od wyniku porównania zostaje ustalony stan przerzutnika w rejestrze wyjściowym. Stan ten wpisywany jest następnie do odpowiedniego przerzutnika rejestru wyjściowego, powodując ponowny przyrost napięcia kompensującego na wyjściu przetwornika c/a o wartość ΔUk. W ten sposób po n krokach przetwarzania cyfrowa zawartość rejestru wyjściowego jest równoważnikiem analogowej wartości napięcia Uwe i może zostać przekazana do cyfrowych urządzeń zewnętrznych.

Zaletą tego typu przetworników jest możliwość budowania przetworników wielobitowych o dużej szybkości przetwarzania, gdyż czas przetwarzania jest równy nT, gdzie T jest czasem jednego cyklu porównania.

Wadą natomiast dość duża nieliniowość różniczkowa, spowodowana właściwościami samego przetwornika. Brak monotoniczności w charakterystyce przetwornika może być przyczyną brakujących kodów, czyli dziur kodowych w charakterystyce całego przetwornika.

- <u>Dobór parametrów pracy przetwornika analogowo-cyfrowego.</u>

  Jako napięcie referencyjne dla przetwornika ADC , wybrałem 3.3 V czyli napięcie Vdd zasilające mikrokontroler. Napięcie to w celu uniknięcia zakłóceń zostało odfiltrowane za pomocą kondensatorów oraz wejściowego dławika. Nie zaobserwowano odkształceń napięcia w trakcie działania urządzenia.

  W celu poprawienia szybkości transmisji, wykorzystuję rozdzielczość 8 bitową przetwornika. Mimo iż rozdzielczość przetwornika wbudowanego w mikrokontroler może wynosić nawet 12 bitów (rozdzielczość można wybierać programowo za pomocą odpowiednich rejestrów- możliwe tryby: 6, 8 ,10 ,12 bit). Zaletą tego rozwiązania jest fakt iż czas pomiaru ADC każdej próbki jest krótszy o połowę niż dla rozdzielczości 12 bit i wynosi tylko 1us, oraz to iż bufor danych wysyłany do drugiego urządzenia pomieści 32 próbki, natomiast w przypadku zastosowania rozdzielczości 12bit , bufor z próbkami do wysłania mógłby pomieścić tylko 16 takich próbek. Z przeprowadzonych badań zaobserwowałem że różnica w jakości dźwięku dla rozdzielczości 8 bit a 12 bit nie jest aż tak odczuwalna.

Spowodowane jest to zapewne opóźnieniami związanymi z tym iż dla 12 bit maksymalna liczba próbek jaką za jednym razem mogę wysłać wyniosi właśnie tylko 16 , czyli połowę mniej jak dla 8 bit .

Całkowity czas konwersji jednej próbki zgodnie z notą wynosi:

tCONV = Sampling time + 12.5 x ADC clock cycles

gdzie : ADC clock cycle wynosi u mnie 1/(14*10^6), Sampling time=1.5 *ADC clock cycle,

czyli łącznie całkowity czas pomiaru jednej próbki wynosi **tCONV= 1[us]**

Zważając na fakt iż wybrana przeze mnie częstotliwość próbkowania wynosi dokładnie 10kHz (czyli co 100[us]). Czas pomiaru 1[us] jest wręcz optymalny. Widmo mowy obejmuje częstotliwości od 100 Hz do około 4kHz, przy czym największa gęstość widmowa (energia) przypada w okolicy 500 Hz i sukcesywnie maleje ze wzrostem częstotliwości. Dla dobrego zrozumienia mowy i rozpoznania osoby mówiącej wystarczy pasmo, w którym jest zawarta główna część energii, to znaczy w zakresie od 300 Hz do 3400 Hz. Dlatego częstotliwość próbkowania 10kHz w zupełności wystarczy celu spełnienia warunku twierdzenia Kotielnikowa-Shannona.

- <u>Wykorzystanie przetwornika ADC w układzie</u>

  Działanie przetwornika ADC wbudowanego w mikrokontroler jest ściśle powiązane z innymi układami wewnętrznymi mikrokontrolera , takimi jak blok Timer, oraz układ DMA. Szczegóły dotyczące współpracy poszczególnych elementów ze sobą omówione zostaną w punkcie [4].

b) Przetwornik DAC oraz kontroler DMA

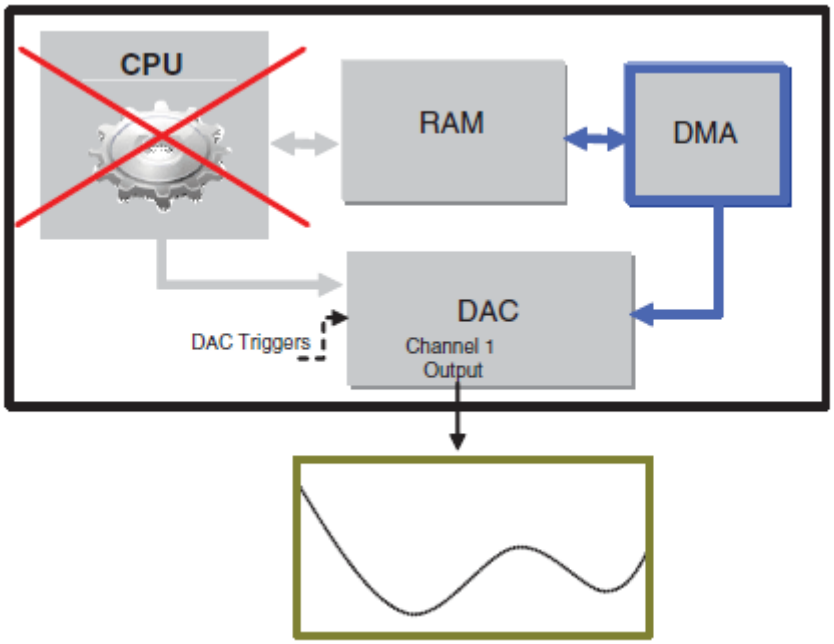
Wykorzystany przetwornik DAC jest 12 bitowym przetwornikiem z możliwością bezpośredniej współpracy z układem DMA mikrokontrolera STM32F051R8T6.
Jako że przesyłane przeze mnie próbki są długości 8 bitów, DAC również pracuje jako 8 bitowy co daje możliwość 256 różnych wartości danej próbki.
Napięcie wyjściowe przetwornika oblicza się z następującego wzoru:

$$DAC_{output} = V_{DD} \times \frac{DOR}{256}$$

gdzie:

- ✓ V_{DD} – jest to wartość napięcia odniesienia dla przetwornika DAC – w naszym przypadku jest ona równa napięciu zasilania całego mikrokontrolera czyli dokładnie 3.3 [V]
- ✓ **DOR** – zawartość rejestru danych wyjściowych dla przetwornika DAC. Jest to liczba 8 bitowa reprezentująca cyfrową wartość napięcia na wyjściu przetwornika DAC w danej chwili.

Poniżej pokazany jest poglądowy schemat współpracy przetwornika DAC z układem bezpośredniego dostępu do pamięci tak zwanym - **kontrolerem DMA** (z ang. Direct Memory Access).



Co to jest kontroler DMA i do czego służy ?

Wiele zadań w systemach mikroprocesorowych polega na przesyłaniu danych z jednego miejsca w pamięci do drugiego. Jak wiadomo w podczas takich działań nie wykonywane są żadne operacje arytmetyczno logiczne przez CPU czyli ściślej mówiąc nie ma potrzeby wykorzystywania mocy obliczeniowej i rejestrów CPU. I właśnie dla możliwości odciążenia rejestrów CPU zrodziła się idea budowy układu służącego do transmisji bloków danych w pamięci. Układy z tej grupy noszą nazwę kontrolerów DMA (*Direct Memory Access*). Zajmują się całą pracą związaną z kopiowaniem i przenoszeniem dużych bloków danych, zwalniając tym samym z tego obowiązku CPU.
Zastosowany przeze mnie mikrokontroler STM32F051R8T6 jest wyposażony w 5-kanałowy kontroler DMA o ustawianej na 8, 16 lub 32 bity długości słowa danych. Każdemu z kanałów można przypisać określony priorytet. Możliwa jest transmisja pomiędzy dwoma układami peryferyjnymi, z układu peryferyjnego do pamięci, z pamięci do układu peryferyjnego oraz z pamięci do pamięci. Dane można pojedynczo lub w postaci całych bloków danych, przy czym maksymalny rozmiar takiego bloku może wynosić 65535. Kontroler DMA, wbudowany w mikrokontrolery STM32 rozróżniam cztery programowo ustalane priorytety:

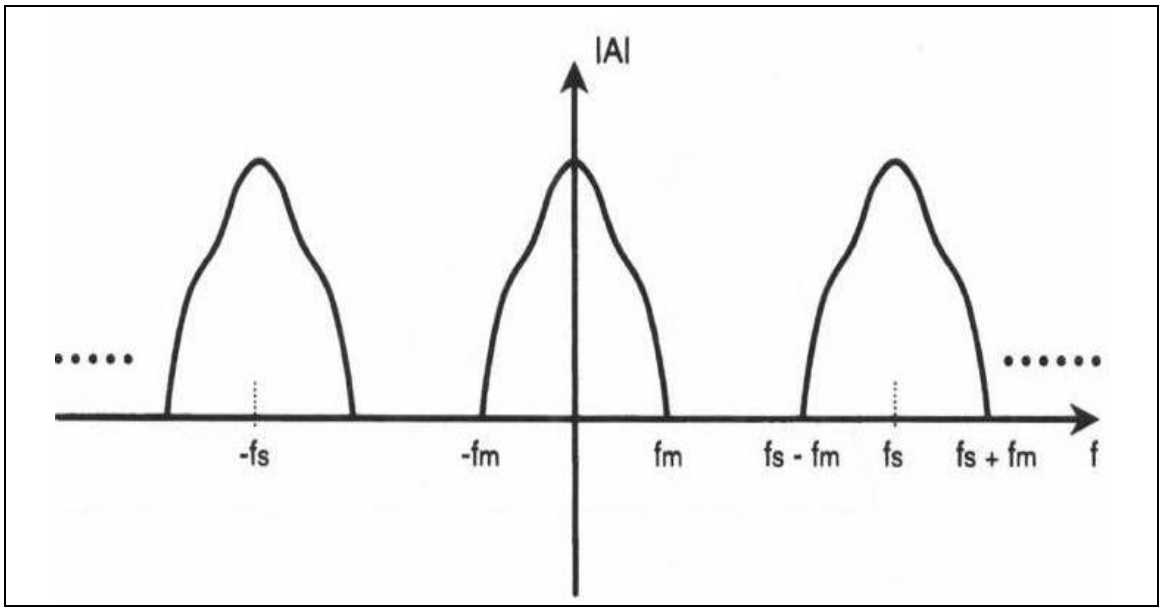
- ✓ najwyższy (*very high priority*),
- ✓ wysoki (*high priority*),
- ✓ średni (*medium priority*),
- ✓ niski (*low priority*).

Jako że w naszym układzie wykorzystywane są 2 kanały DMA (transmisja: przetwornik ADC ->pamięć RAM, oraz transmisja: pamięć RAM ->przetwornik DAC) priorytet najwyższy ustawiony jest dla kanału 1 czyli kierunek ADC -> RAM. Jak wiadomo pobranie jak największej ilości próbek jest sprawą kluczową, natomiast odtworzenie już odebranych danych ma mniej priorytetowe znaczenie (oczywiście jeśli chodzi o czas kiedy to odtworzenie nastąpi). Dlatego też dla 3 kanału kontrolera DMA priorytet wywołania ustawiony został jako wysoki.

c) Wejściowy filtr antyaliasingowy

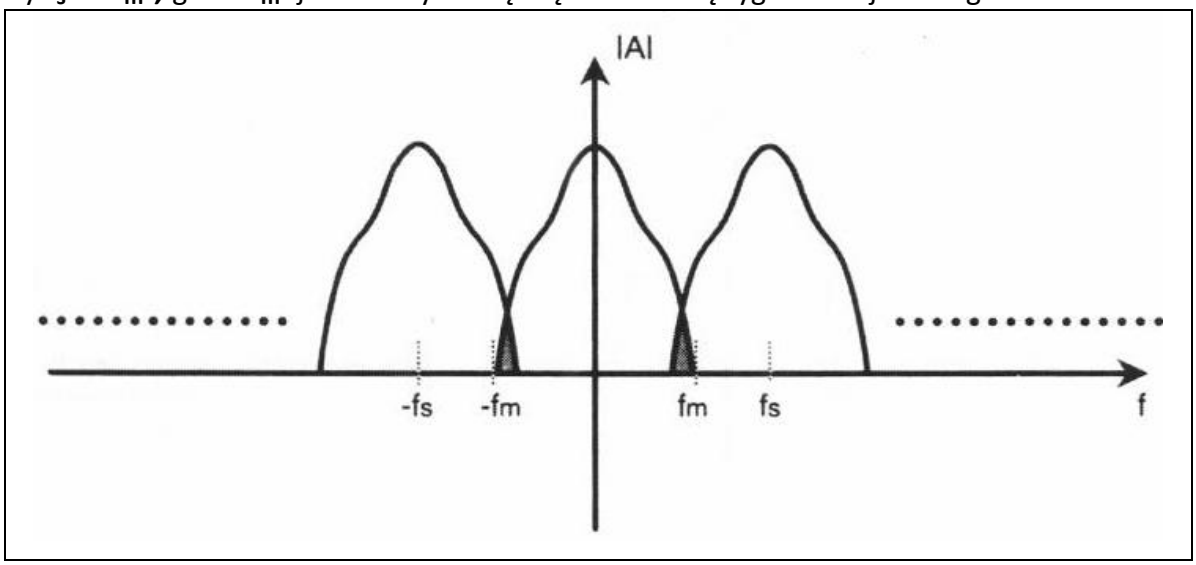
Co to jest zjawisko aliasingu oraz do czego służy filtr antyaliasingowy ?

Aliasing - to nieodwracalne zniekształcenie sygnału w procesie próbkowania wynikające z niespełnienia założeń twierdzenia Kotielnikowa - Shannona. Zniekształcenie to objawia się obecnością w sygnale składowych o błędnych częstotliwościach (aliasów).
Rezultat próbkowania jest podobny do mnożenia sygnału przez szereg przebiegów sinusoidalnych o częstotliwościach będących wielokrotnościami częstotliwości próbkowania f_s. Oznacza to że widmo wejściowe powtarzane jest co f_s.



Rys. Widmo próbkowane $f_s > f_m$ - widmo powtarza się co każde f_s

Gdy $f_s < f_m$, gdzie f_m jest maksymalną częstotliwością sygnału wejściowego :



Rys. Pasma częstotliwościowe nałożą się na siebie, co spowoduje interferencję sygnału wejściowego -> efekt ten właśnie nazywany jest **aliasingiem.**

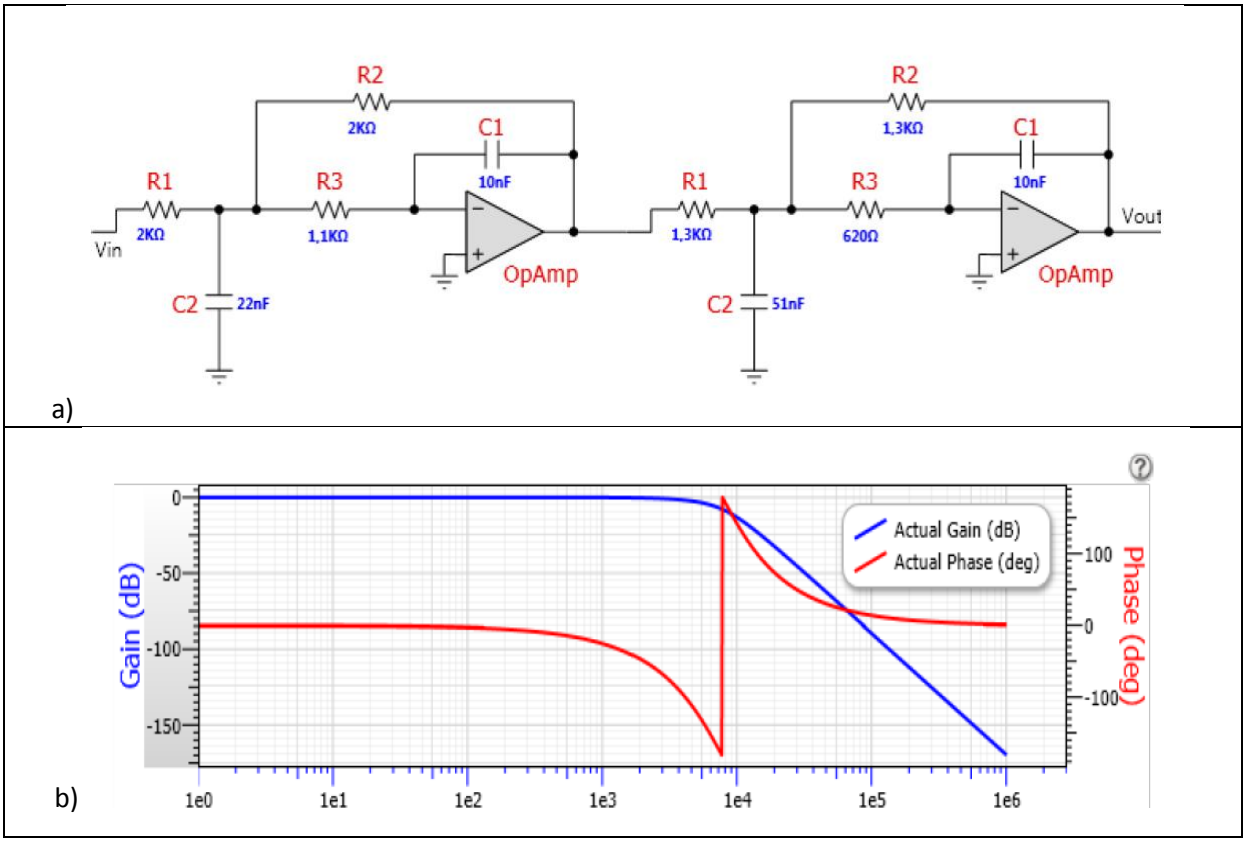
Jak widać przebieg sinusoidalny o wyższej częstotliwości ma teraz swój „alias" (element z nim utożsamiany) w przebiegu o niższej częstotliwości, będący efektem dokonanego

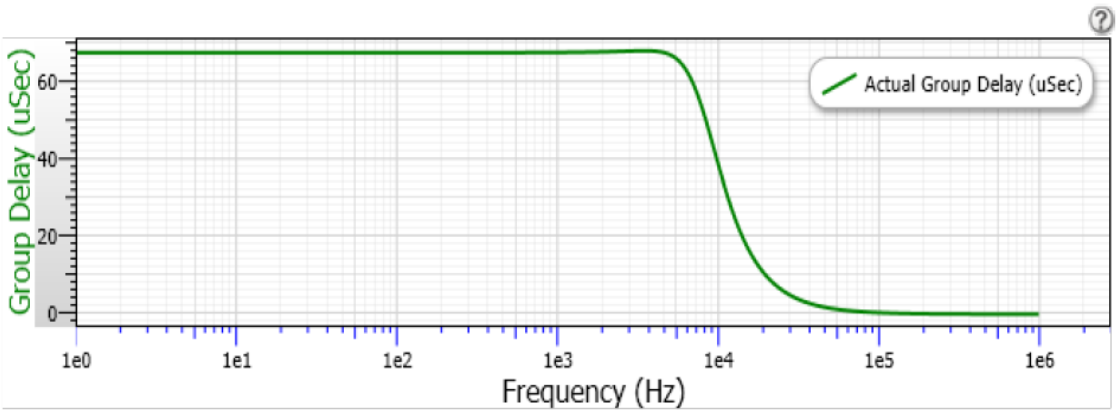
próbkowania. Tak więc próbki nie opisują sygnału wejściowego i w konsekwencji dalsze ich przetwarzanie cyfrowe będzie błędne.

Filtr antyaliasingowy to zwykły filtr dolnoprzepustowy umieszczony przed przetwornikiem A/D o częstotliwości granicznej równej częstotliwości nyquista próbkowanego sygnału.

Poniżej przedstawiono zaprojektowany filtr antyaliasingowy zaprojektowany za pomocą programu firmy Texas Instruments **FilterPro.** Zastosowanie programu pozwoliło na znaczne ułatwienie projektowania filtru. Pozwoliło pozbyć się żmudnych obliczeń a skupić się na doborze odpowiedniego typu i parametrów filtra.

Założenia projektowe filtru:
- dolnoprzepustowy
- charakterystyka Bessela
- częstotliwość graniczna – 5kHz
- tłumienie w paśmie zaporowym : -80 dB





c)

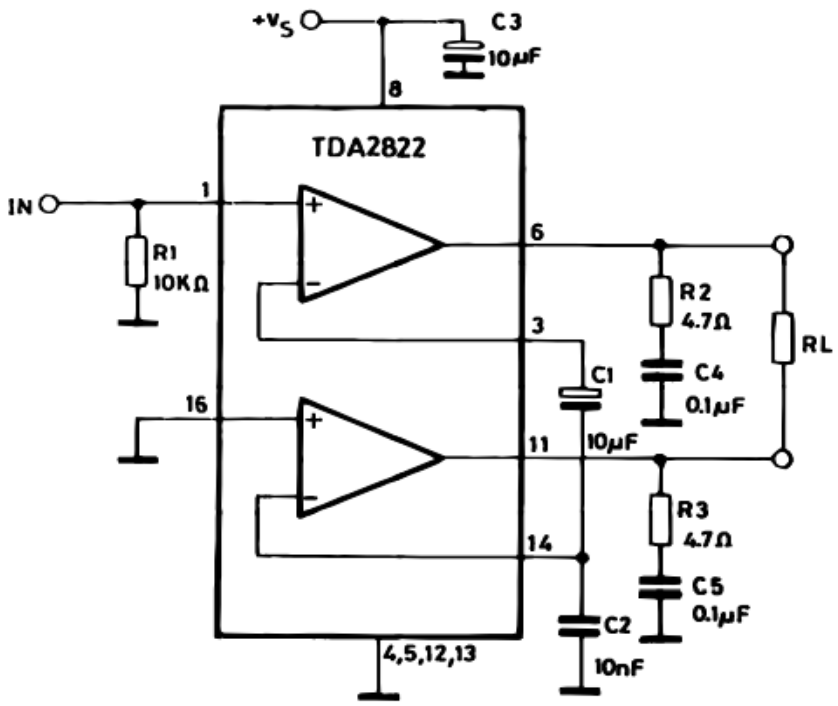
Rys. Schemat a) oraz charakterystyki opisujące filtr: b)amplitudowa i fazowa c)opóźnienia grupowego

Filtr Bessela zapewnia liniowy przebieg charakterystyki fazowej i stały poziom opóźnienia grupowego w paśmie przepustowym. Z tych względów filtr o takiej charakterystyce najczęściej stosuje sie w aplikacjach audio. Duże nachylenie charakterystyki filtra wymusza selekcje wzmacniacza operacyjnego o jak najniższym poziomie szumów i jak najniższych zniekształceniach. Zastosowano rezystory i kondensatory o tolerancji 5 % Dla jak najlepszych rezultatów należy wykorzystać
kondensatory o dużej stabilności temperaturowej. Użycie rezystorów o wartości mniejszej nie większych niż 2 k zminimalizowało wpływ szumów termicznych tych elementów. Jako wzmacniacze zastosowano popularne a o nie najgorszych parametrach NE5532.

d) Wzmacniacz sygnału wyjściowego:

Wzmacniacz sygnału wyjściowego TDA2822
Na wyjściu przetwornika w celu wzmocnienia amplitudy sygnału wyjściowego zastosowałem układ scalony firmy ST microelectronics. Jest to monolityczny wzmacniacz mocy mogący pracować przy napięciu już od 3V.
Jego moc wyjściowa przy zasilaniu 5V i obciążeniu głośnikiem o impedancji 4Ω, wynosi 2W; Charakteryzuje się on stosunkowo niewielkim współczynnikiem THD przy zasilaniu niskim napięciem , a także niewielkimi całkowitymi stratami mocy.

Układ TDA2822 może pracować w trybie stereo (dwukanałowym) lub w trybie mono w tak zwanym układzie mostkowym - bridge (jednokanałowy).
Jako że nie wykorzystuję trybu stereo w moim domofonie, układ wzmacniacza pracuje w konfiguracji mostkowej. Zabieg ten pozwala również na zwiększenie mocy wyjściowej wzmacniacza.
Poniżej schemat aplikacyjny układu TDA 2822 w konfiguracji mostkowej:



Rys. schemat układu TDA2822 w konfiguracji mostkowej

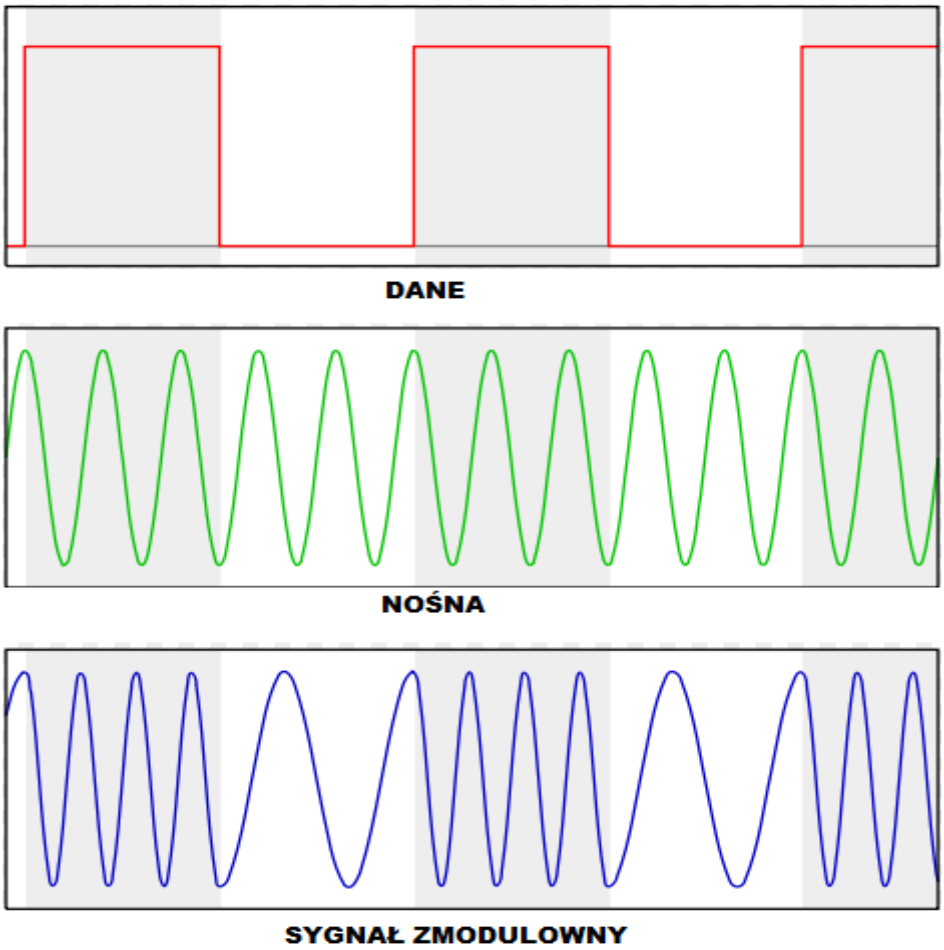
e) **Transciever RFM 70**

Jako że konstruowany domofon jest bezprzewodowy , kluczowym elementem w doborze był obwód radiowy. Zdecydowałem się na układ RFM70 firmy Hoperf. Podstawowym kryterium wyboru była cena oraz oczywiście możliwości sprzętowe modułu.

Układ RFM70 jest to kompletny moduł radiowy zintegrowany z anteną, pracujący w nielicencjonowanym paśmie ISM - 2,4 GHz. Rodzjem modulacji sygnału wykorzystywanym w tym układzie jest tak zwana modulacja typu GFSK (odmiana modulacji FSK).

Pokrótce opiszę ideę działania modulacji FSK; FSK (ang. Frequency - Shift Keying) - modulacja częstotliwości dla sygnałów cyfrowych, czyli kluczowanie z przesuwem częstotliwości. Przy stałej amplitudzie harmonicznego sygnału nośnego następuje zmiana częstotliwości: niższej dla symbolu "zera logicznego" i wyższej dla "jedynki logicznej" informacji binarnej, czasami przypisanie częstotliwości może być odwrotne.

Poniżej wykresy ilustrujące zasadę działania modulacji FSK:

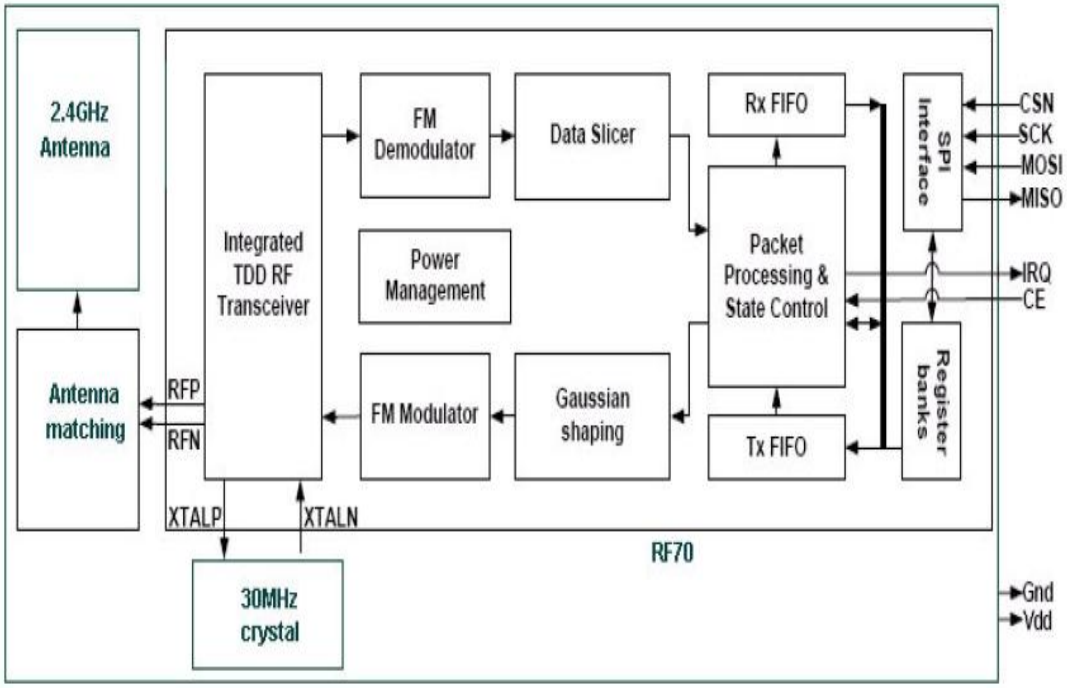


Rys. Zasada działania modulacji FSK

Natomiast zastosowany w układzie rodzaj modulacji GFSK różni się tylko tym od typowej modulacji FSK że fale sygnału przyjmują kształt krzywych Gaussa. Wygładzenie zboczy impulsów odbywa się przy pomocy filtru Gaussowskiego, kolejnym etapem jest modulacja FSK. Efektem zastosowania filtru Gaussowskiego jest zmniejszenie szerokość widma sygnału.

Moduły te mogą przesyłać dane z prędkością 1 lub 2 Mb/s.

Tor nadawczy wyposażono we wzmacniacz o programowalnej mocy wyjściowej w zakresie -40 dBm do 5 dBm. Wymiana danych pomiędzy modułem a otoczeniem odbywa się za pomocą 4-przewodowej magistrali SPI z maksymalną dostępną prędkością taktowania 8 MHz. Oprócz tego układ ten posiada tak zwany „głęboki tryb uśpienia” co jest bardzo przydatną cechą w układach zasilanych bateryjnie. Działanie tego trybu polega na tym iż w czasie oczekiwania na odbiór danych, układ przechodzi w stan obniżonego poboru mocy , wyłączając automatycznie zbędne obwody wewnętrzne. Po odebraniu danych zgłaszane jest przerwanie które automatycznie „wybudza układ” – pozostałe obwody zostają uruchomione.



Rys. Schemat ideowy układu RFM70

4. Algorytm działania urządzenia

W tym punkcie opiszę kluczowe zagadnienie dotyczące poprawnego działania całego urządzenia – oprogramowanie.
Program został całkowicie napisany w języku C, z wykorzystaniem darmowego środowiska CooCox , oraz kompilatora ARM GCC.
W projekcie wykorzystałem też niektóre funkcje z biblioteki SPL (Standard Peripheral Library) dostarczanej przez firmę ST Microelectronics.
Projekt składa się z 5 plików:

- ***main.c*** -> główny plik programu;
- ***rfm70.h i rfm70.c*** -> pliki zawierające napisaną bibliotekę do obsługi układów RFM 70;
- ***adc_tim_dac.h i adc_tim_dac.c*** -> pliki zawierające kod obsługi przetworników ADC, DAC, DMA, układów zegara, oraz funkcja do wykrywania ciszy w sygnale.

Opis algorytmu:

Opisywany przeze mnie algorytm działania urządzenia będzie dotyczył układu nadajnika. Kod odbiornika różni się tylko nieznacznie od kodu nadajnika. Używana przeze mnie nazwa nadajnik czy odbiornik może być myląca, ze względu na to układu nadajnika pracuje również jako odbiornik i odwrotnie. Mówiąc nadajnik mam na myśli układ znajdujący się przy wejściu do domu z przyciskiem wywołania.

Po włączeniu zasilania układ znajduje się w trybie całkowitego uśpienia z poborem prądu rzędu 13 [uA] , z możliwością wybudzenia tylko za pomocą sygnału zewnętrznego. Po wciśnięciu przycisku wywołania mikrokontroler wykonuje inicjalizację poszczególny bloków

wykorzystywanych w transmisji (bloki ADC, DAC, kanły DMA, magistrala SPI, układy licznikowe). Następnie wysyłana jest ramka danych z odpowiednim kodem , informując odbiornik by ten wygenerował zapisany w pamięci FLASH sygnał syreny. Następnie mikroprocesor przechodzi w ciągły cykl pracy składający się z następujących stanów :

- ✓ Próbkowanie sygnału z mikrofonu i przechowywanie , zebranych próbek w 3 przełączanych 32 bajtowych buforach;
- ✓ Wysyłanie zawartości buforów do układu odbiornika;
- ✓ Po wysłaniu ramki danych , przełączenie w tryb odbioru i natychmiastowy odbiór danych z odbiornika;
- ✓ Ponowne przełączenie w tryb nadawania i wysłanie ramki do odbiornika;

Płynność działania urządzenia oraz brak gubienia danych , udało się osiągnąć dzięki zastosowaniu kontrolerów DMA, dzięki którym pobieranie próbek z ADC, wysyłanie do DAC, a także obsługa RFM70 (magistrali SPI) może odbywać się równocześnie. Dodatkowo po 30 sekundach od naciśnięcia przycisku wywołania program analizuje pobierane próbki z ADC. Analiza ta ma na celu pomiar amplitudy napięcia wejściowego , i zbadanie czy na wejściu mamy do czynienia z brakiem mowy ludzkiej. Analiza ta trwa 5 sekund. Jeśli układ wykryje na wejściu ciszę, mikrokontroler przechodzi w stan uśpienia , wcześniej wysyłając sygnał do odbiornika z odpowiednim kodem informującym go o przejściu w stan uśpienia.

Badanie wystąpienia ciszy polega na obliczeniu średniej arytmetycznej z 50000 próbek:

$$\overline{x} = \frac{x_1 + x_2 + \ldots + x_N}{N} = \frac{\sum_{i=1}^{N} x_i}{N}$$

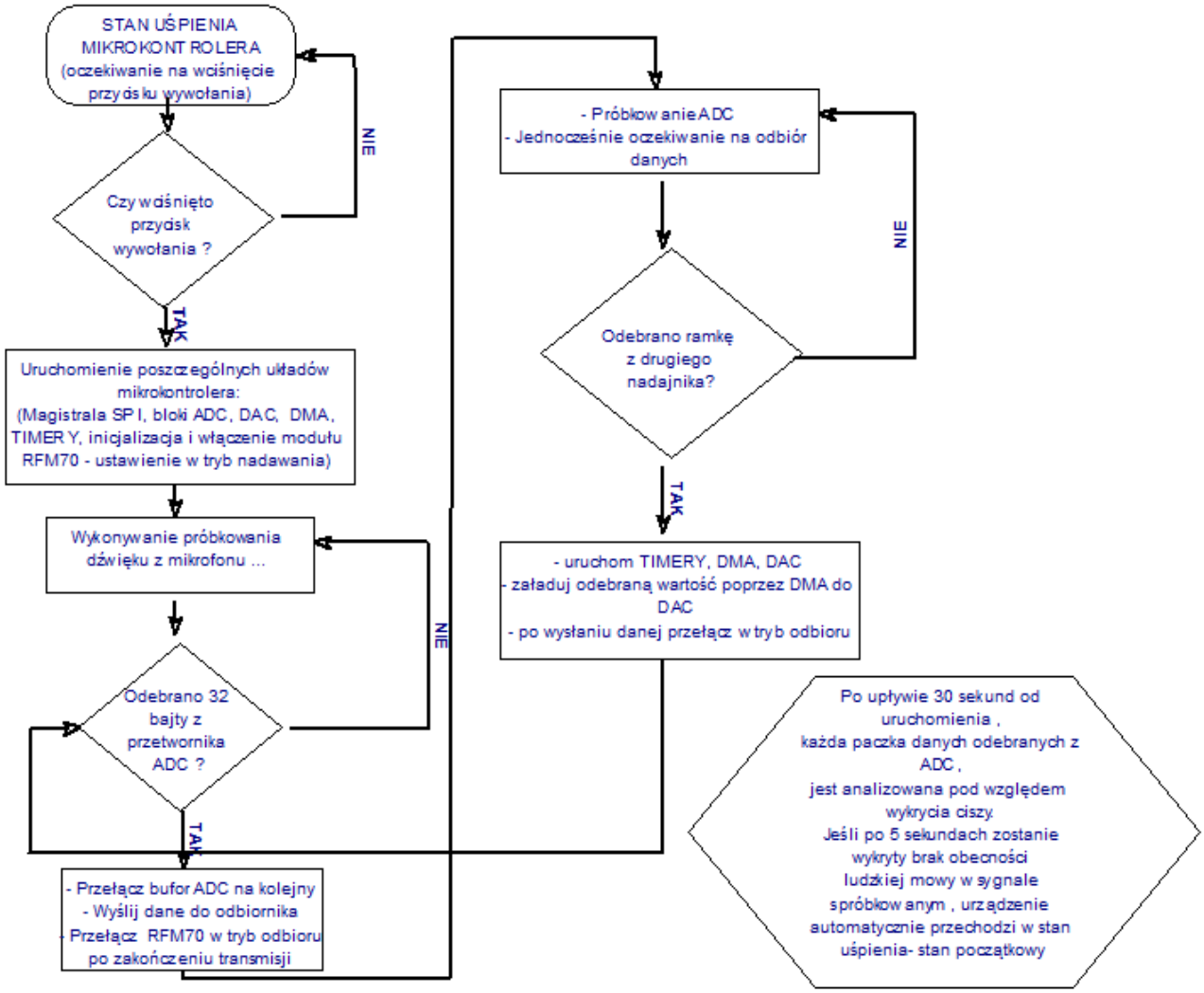
$\overline{x}$ - symbol średniej arytmetycznej,

x_N - wartość badanej próbki w postaci cyfrowej

N - liczba próbek w buforze (tablicy)

Tyle próbek zostaje pobranych w ciągu 5 sekund. Średnia arytmetyczna jest liczona z próbek pobranych do każdego bufora (32 próbki w buforze) , a następnie ustawiana flaga czy wartość przekroczyła dobraną eksperymentalnie wartość w postaci cyfrowej uznawaną za ciszę ($500_{(10)}$ zakres 0-$4095_{(10)}$). Stany flag przechowywane są w tablicy bool - 156 elementowej. Kolejna tablica 10 elementowa analizuje stan z 10 ostatnio zapełnionych tablic 156 elementowych. Jeśli średnia boolowskich jedynek jest większa niż 0 , znaczy to że mamy do czynienia z ciszą. Układ zostaje wyłączony.

Rys. 4.2 Schemat blokowy algorytmu działania urządzenia



Rys. Algorytm działania urządzenia

```
/*
 * domofon bezprzewodowy stm32f051r8 , zegar HSI 48MHz
 */

#include "rfm70.h"
#include "adc_tim.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "stm32f0xx_dma.h"
#include <stdbool.h>
void my_receive_packet(void * pBuf, uint8_t len); // deklaracja funkcji do odbioru ramek
volatile uint8_t full_frameADC_received;
volatile uint8_t num_buf;
volatile bool TX_RX;
int main(void) {

	RFM70_Init();
	ADC1_DMA_Config();
	DAC_DMA_Config();
```

```
        SetChannelNum(46);  //  wybor kanalu czsstotliwosc 2400 - 2483.5 GHz
        register_rfm70_rx_event_callback(my_receive_packet); //rejestracja zdarzenia rx_event

        while (1) {
                switch (TX_RX) {
                case 0: {
                RFM70_RX_EVENT(); // zdarzenie odbiorcze z RFM70
                }

                case 1: {

                if (full_frameADC_received == 1) {
                        Send_Packet(W_TX_PAYLOAD_NOACK_CMD, &buf_ADC[num_buf ^ 1][0],
                                                SIZE_BUF_ADC);
                                full_frameADC_received = 0;
                                TX_RX ^= 1;       // przelaczam na wysylanie
                            }
                        }
                 }
        }
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////
// funkcja do analizy odebranych danych z rfm
void my_receive_packet(void * pBuf, uint8_t len) {

        memcpy(&buf_DAC[num_bufDAC][0], pBuf, len); // kopiowanie do aktualnie pustego bufora
        DMA_Cmd(DMA1_Channel3, ENABLE);
        num_bufDAC ^= 1;
        TX_RX ^= 1;

}
```

Listing 1 : kod pliku głównego

5. Problemy powstałe podczas konstrukcji urządzenia

Jest jasne że podczas konstrukcji jakiegokolwiek urządzenia pojawiają się problemy natury zarówno sprzętowej , programistycznej , lub po prostu powstałe w wyniku winy konstruktora. Nie inaczej było w moim przypadku. Chciałbym się podzielić w tym punkcie problemami z jakimi borykałem się podczas konstruowania domofonu.

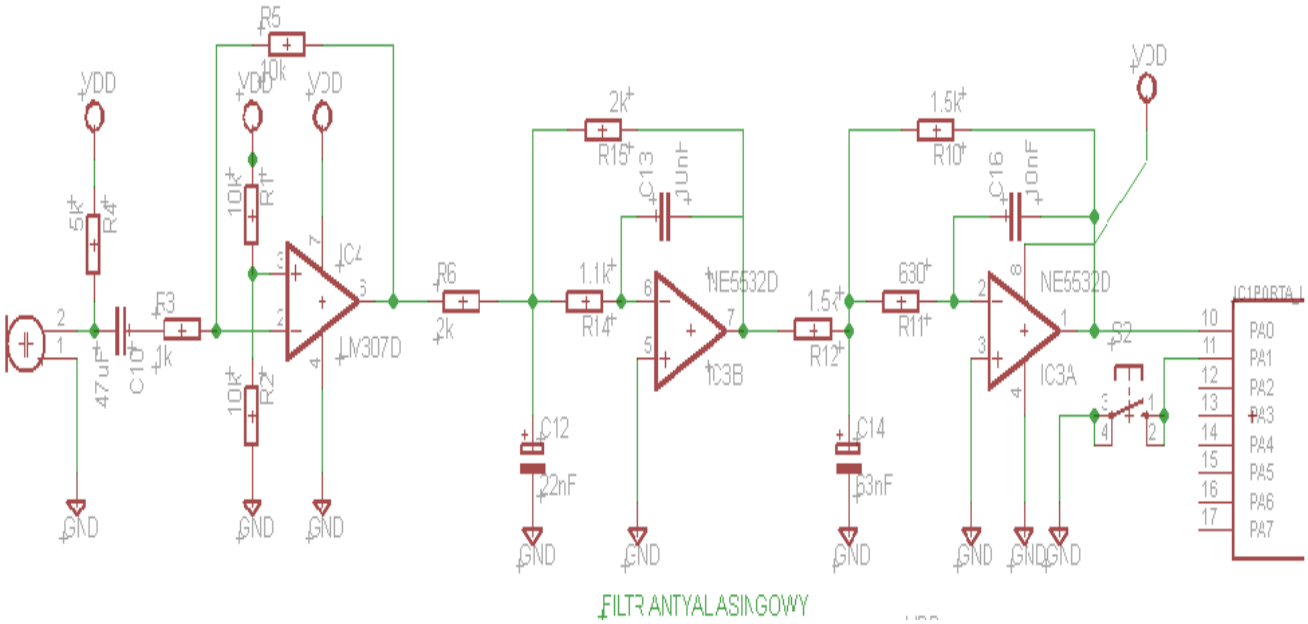
1. Skomplikowana obsługa modułu transcievera RFM 70
   Jak się okazało, napisanie kodu biblioteki do układów RFM 70 nie było prostą sprawą. Układ ten jest bardzo rozbudowany funkcjonalnie , a udostępniona przez producenta dokumentacja techniczna pozostawia wiele do życzenia. Uruchomienie układów, a ściślej dwu kierunkowej transmisji między nimi zajęło mi bardzo dużo czasu.

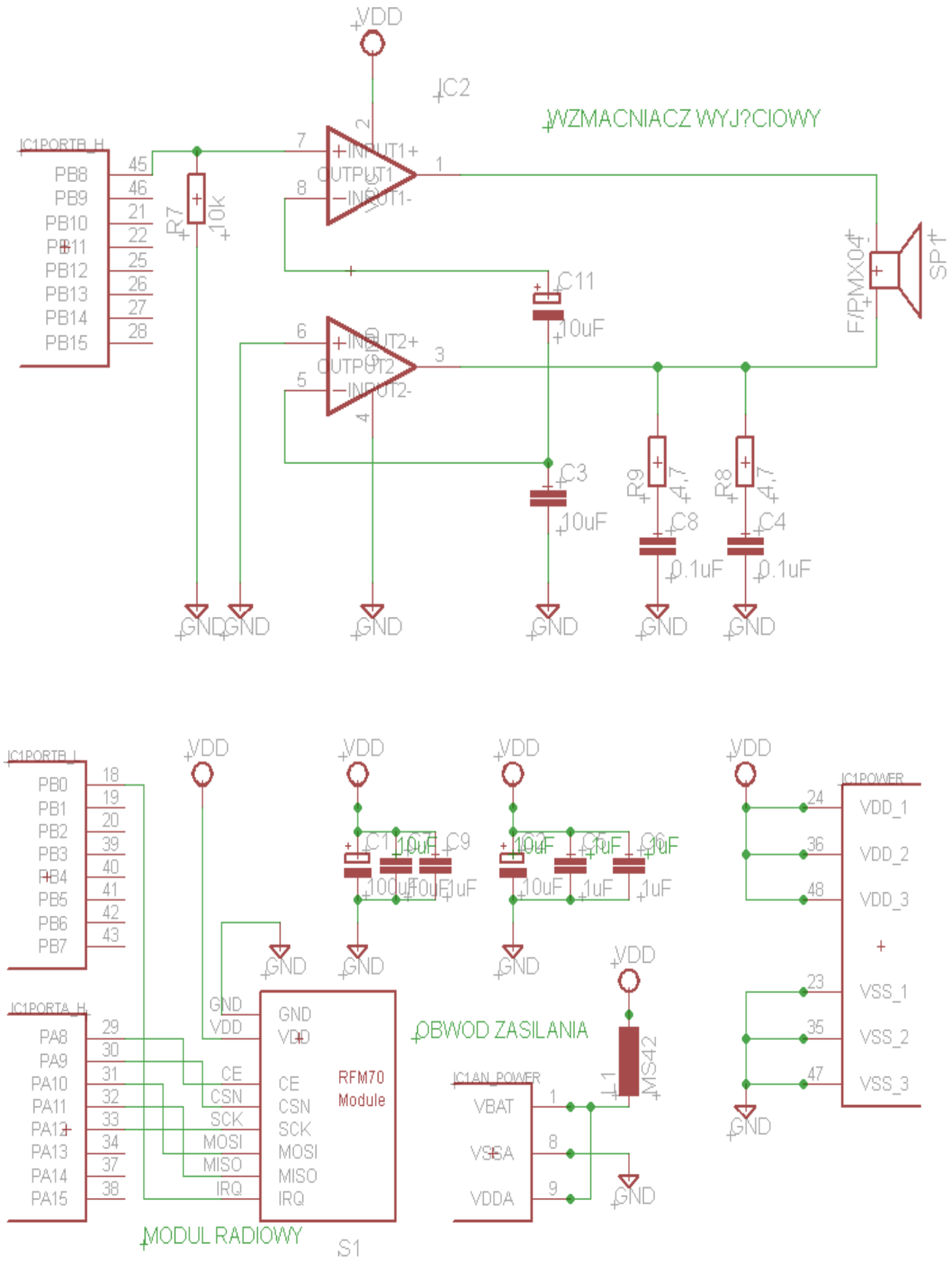
2. Problem z dużym szumem w sygnale wyjściowym.
   Rozwiązaniem tego problemu okazało się zastosowanie filtra antyaliasingowego na wejściu przetwornika ADC.

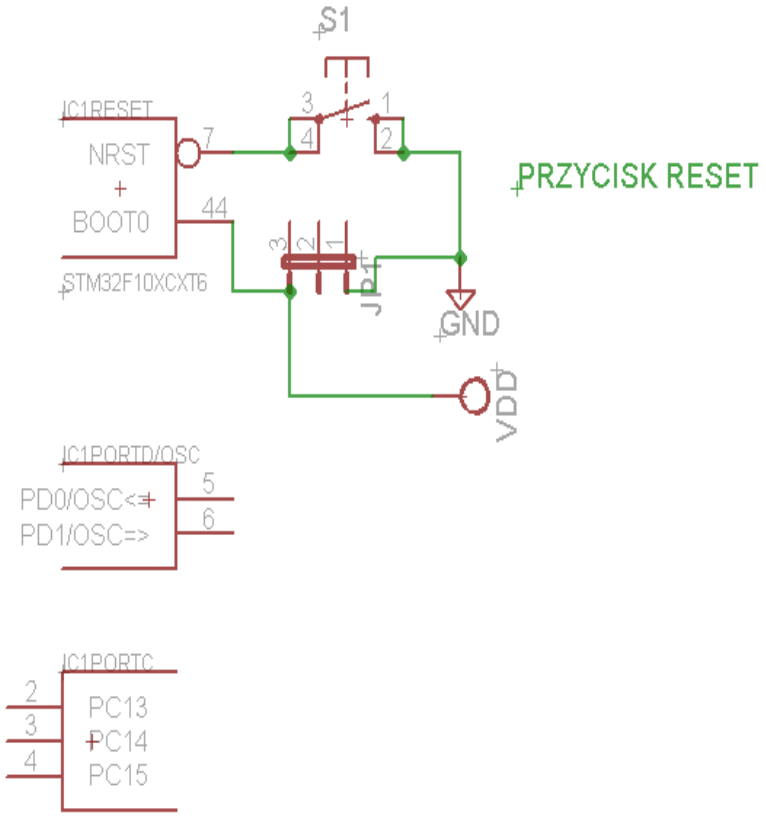
3. Lekkie zniekształcenia sygnału spowodowane gubieniem poszczególnych próbek sygnału. Problem polegał na tym iż w układzie powstawały opóźnienia związane z obsługą naprzemiennego wysyłania i odbierania danych poprzez układy RFM. Rozwiązaniem które zastosowałem było wprowadzenie 3 buforów zarówno dla danych odbieranych jak i wysyłanych . Dźwięk jest opóźniony lekko do rzeczywistego ale nie słychać dzięki temu dużych trzasków w sygnale wyjściowym. Niestety zastosowany procesor okazał się trochę zbyt wolny i całkowicie nie udało się wyeliminować tego problemu.
   Optymalnym rozwiązaniem jednak byłoby zastosowanie osobnych układów RFM 70 pracujących osobno jako odbiornik i nadajnik.

6. Schemat ogólny urządzenia oraz płytki PCB

Przedstawiony poniżej schemat elektryczny urządzenia oraz jego płytka PCB która do momentu pisania tego tekstu nie została ukończona (wyjaśnienie dalej), zostały zaprojektowane w programie EAGLE 5.11 wersja Lite. Jest to program komputerowy służący do wspomagania projektowania obwodów elektrycznych niemieckiej firmy CadSoft. Zawiera on edytor schematów oraz płytek drukowanych.




VDD
IC2
WZMACNIACZ WYJ?CIOWY
IC1PORTB_H
PB8
PB9
PB10
PB11
PB12
PB13
PB14
PB15
45
46
21
22
25
26
27
28
R7
10k
INPUT1+
OUTPUT1
INPUT1-
7
8
2
1
C11
10uF
F/PMX04
SP1
INPUT2+
OUTPUT2
INPUT2-
6
5
4
3
C3
10uF
R9
4,7
R8
4,7
C8
0.1uF
C4
0.1uF
GND
IC1PORTB_L
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
18
19
20
39
40
41
42
43
C1
10uF
C9
100uF
10uF
1uF
C7
10uF
C6
1uF
1uF
IC1POWER
VDD_1
VDD_2
VDD_3
VSS_1
VSS_2
VSS_3
24
36
48
23
35
47
IC1PORTA_H
PA8
PA9
PA10
PA11
PA12
PA13
PA14
PA15
29
30
31
32
33
34
37
38
GND
VDD
CE
CSN
SCK
MOSI
MISO
IRQ
RFM70
Module
MODUL RADIOWY
S1
OBWOD ZASILANIA
IC1AN_POWER
VBAT
VSSA
VDDA
1
8
9
L1
MS42



Rys. Schemat poszczególnych bloków urządzenia

7. Wnioski

Mogę śmiało stwierdzić iż projekt nie jest jeszcze w pełni ukończony. Głównym problemem okazała się zbyt wolna obsługa magistrali SPI przez mikrokontroler co ma odbicie w jakości dźwięku. Przesyłany dźwięk mowy jest w pełni zrozumiały jednak zdarza się czasem gubić niektóre próbki. Cały czas pracuję nad rozwojem oprogramowania . Moim zadaniem jest też implentecja filtru typu FIR który pozwoli wyeliminować resztę zakłóceń. Płytka PCB musi zostać zrobiona w profesjonalnych warunkach, gdyż wytrawiona przeze mnie w domowych warunkach, nie spełniła swojego zadania- układ nie działał poprawnie. Mimo wszystko jestem zadowolony z otrzymanych wyników.

8. Literatura

1. Tomasz P. Zieliński - Cyfrowe przetwarzanie sygnałów od teorii do zastosowań
2. Górecki Piotr - Wzmacniacze operacyjne
3. Paul Horowitz - Sztuka elektroniki
4. Reference manual RM0091: STM32F05xxx advanced ARM-based 32-bit MCUs
5. Application note AN4058: Audio and waveform generation using the DAC in STM32F0xx microcontroller families
6. Reference manual RFM70 v1.0: Low Power High Performance 2.4 GHz GFSK Transceiver Module
7. Design Methodology for MFB Filters in ADC Interface Applications: http://www.ti.com/lit/an/sboa114/sboa114.pdf

8. Sallen-Key Low Pass Filter Design Equation: http://www.daycounter.com/Filters/SallenKeyLP/Sallen-Key-LP-Filter-DesignEquations.phtml
9. FilterPro User's Guide http://www.ti.com/lit/an/sbfa001c/sbfa001c.pdf
10. Precision Analog Applications Seminar Texas Instruments : “Input Drive Circuitry for SAR ADCs”