

Politechnika Warszawska

W Y D Z I A Ł M E C H A T R O N I K I



Praca dyplomowa inżynierska

na kierunku Inżynieria Biomedyczna
w specjalności Aparatura Medyczna

Urządzenie do bezprzewodowej akwizycji sygnałów
elektriomiotograficznych

numer pracy według wydziałowej ewidencji prac: 114D-ISP-IB/305386/1200285

Szymon Zelkowski

numer albumu 305386

promotor

dr inż. Krzysztof Wildner

konsultacje

—

WARSZAWA 2023

PRACA DYPLOMOWA inżynierska

Specjalność: Aparatura Medyczna

Instytut prowadzący specjalność: Instytut Metrologii i Inżynierii Biomedycznej

Instytut prowadzący pracę: Instytut Metrologii i Inżynierii Biomedycznej

Temat pracy: Urządzenie do bezprzewodowej akwizycji sygnałów elektromiograficznych.

Temat pracy (w jęz. ang.): The device for wireless electromyographic signal acquisition.

Zakres pracy:

1. Zaprojektowanie urządzenia i wykonanie jego wersji prototypowej
2. Przeprowadzenie testów potwierdzających funkcjonowanie opracowanego urządzenia
3. Opracowanie dokumentacji technicznej urządzenia

Podstawowe wymagania:

1. Urządzenie powinno być zasilane baterią
2. Bezprzewodowa komunikacja powinna zostać zrealizowana w oparciu o standard Bluetooth
3. Urządzenie powinno zapewnić rozdzielcość sygnału wyjściowego na poziomie min. 10bit.
4. Urządzenie powinno być przystosowane do pracy w sieci minimum 2 tego rodzaju urządzeń.
5. Powinieneń zostać zdefiniowany protokół komunikacji, służący do transmisji komend sterujących do urządzenia oraz danych wyjściowych z urządzenia, jak również format danych wyjściowych.
6. Urządzenie powinno posiadać obudowę.
7. Dokumentacja techniczna powinna obejmować: dokumentację obwodu drukowanego, dokumentację projektu obudowy, dokumentację protokołu komunikacji oraz krótką instrukcję obsługi.

Literatura:

1. E. Merletti, P.J.Parker: "Electromyography: Physiology, Engineering, and Non-Invasive Applications", Wiley-IEEE Press, 2004

Słowa kluczowe:

Praca dyplomowa jest realizowana we współpracy z przemysłem

Tak/Nie,

Nazwa firmy:

Imię i nazwisko dyplomanta:

Szymon Zelkowski

Imię i nazwisko promotora:

Dr inż. Krzysztof Wildner

Imię i nazwisko konsultanta:

Temat wydano dnia:

21.01.2022

Termin ukończenia pracy:

Miejsce wykonywania praktyki przeddyplomowej: Uośrodkowy Instytut Medyczny ul. Sosnowa 5-129

Zatwierdzenie tematu

Opiekun specjalności

Z-ca Dyrektora Instytutu

Streszczenie pracy

Niniejsza praca dyplomowa przedstawia proces projektowania, tworzenia prototypu urządzenia, badania symulacyjne oraz testowanie funkcjonalności bezprzewodowego urządzenia do akwizycji sygnału elektromiograficznego. Zbudowane urządzenie pozwala w sposób nieinwazyjny na akwizycję sygnału bioelektrycznego z różnych partii mięśni, a następnie umożliwia transmisję danych przy wykorzystaniu Bluetooth Low Energy. Rozmiary urządzenia i zastosowanie zasilania baterijnego pozwala zwiększyć komfort, a przede wszystkim bezpieczeństwo użytkowania.

W pierwszej części pracy zostały przedstawione podstawowe pojęcia i terminy związane z tematyką pracy. Zostały określone wymagania i sfomułowano założenia projektowe dla wzmacniacza, a także zdefiniowano budżet na jego wykonanie. Następnie po opisie działania urządzenia dokonano analizy komponentów, której celem było dobranie właściwych elementów zapewniających poprawne funkcjonowanie. Ponadto zaprezentowano dobór sposób przesyłu sygnału przy użyciu BLE.

W kolejnym rozdziale zaprezentowano projekt urządzenia zrealizowany w oparciu o wcześniej zdefiniowane wymagania i założenia projektowe. Opracowano schematy poszczególnych bloków, dokonano charakterystyki bloków funkcjonalnych toru pomiarowego, a także przeprowadzono proces ich doboru uwzględniając najważniejsze parametry.

Zaprojektowany tor pomiarowy został następnie poddany symulacji, której celem było sprawdzenie poprawności przyjętych rozwiązań. Następnie opracowano niezbędne narzędzia informatyczne w postaci aplikacji komputerowej w język C++, oprogramowanie mikrokontrolera w języku C oraz skrypty do przetwarzania plików w programie Matlab.

W następnej części pracy scharakteryzowano proces fizycznej realizacji prototypu wzmacniacza bioelektrycznego. Opisano proces tworzenia obwodu drukowanego PCB, montaż elementów składowych oraz zaprezentowano sposób realizacji obudowy z wykorzystaniem wydruku 3D. Dodatkowo zostały przedstawione problemy, które pojawiły się podczas realizacji fizycznej urządzenia. Ponadto zamieszczono sumaryczny kosztorys.

Po uruchomieniu wzmacniacza przeprowadzono szereg testów dotyczących poprawności funkcjonowania układu, opracowanego oprogramowania. Następnie zaprezentowano, przy użyciu opracowanej aplikacji, akwizycja sygnału EMG pochodzącego z wybranej partii mięśni.

W podsumowaniu przedstawiono istotne informacje, które dotyczą problemów napotkanych w trakcie projektowania wzmacniacza bioelektrycznego.

W końcowej części pracy zamieszczono bibliografię, spis rysunków, tabel oraz załączników.

Abstract of the thesis

This diploma thesis presents the process of designing, creating a prototype, conducting simulation tests, and functional testing of a wireless device for electromyographic signal acquisition. The constructed device allows for non-invasive acquisition of bioelectrical signals from various muscle groups, and then enables data transmission using Bluetooth Low Energy. The device's size and battery power supply increase comfort and, above all, user safety.

In the first part of the thesis, basic concepts and terms related to the subject matter of the thesis are presented. Requirements and design assumptions for the amplifier were determined, and a budget for its implementation was defined. After describing the operation of the device, a component analysis was performed, aiming to select the appropriate elements ensuring correct functioning. Additionally, the choice of signal transmission method using BLE was presented.

In the next chapter, the device project was presented based on the previously defined requirements and design assumptions. Schematics of individual blocks were developed, functional block characteristics of the measurement path were performed, and the selection process was carried out, taking into account the most important parameters.

The designed measurement path was then subjected to simulation to verify the correctness of the adopted solutions. Subsequently, the necessary computer tools were developed in the form of a C++ computer application, microcontroller software in C, and scripts for processing files in Matlab.

In the next part of the thesis, the process of physical implementation of the bioelectrical amplifier prototype was characterized. The process of creating the PCB printed circuit, assembling the components, and presenting the implementation of the enclosure using 3D printing was described. Additionally, problems that arose during the physical implementation of the device were discussed. Moreover, a summary of the total cost estimate was included.

After starting the amplifier, a series of tests were conducted regarding the proper functioning of the system and the developed software. Then, using the developed application, the acquisition of the EMG signal from a selected muscle group was presented.

In the summary, essential information concerning the problems encountered during the design of the bioelectrical amplifier was presented.

In the final part of the thesis, a bibliography, list of figures, tables, and attachments were included.

Spis treści

Wstęp.....	7
1. Wstęp teoretyczny	8
1.1. Identyfikacja podstawowych pojęć i terminów wykorzystywanych w pracy	8
1.2. Charakterystyka sygnału elektromiograficznego	9
1.2.1. Wzmacniacz biologiczny	9
2. Analiza rozwiązań konstrukcyjnych	12
2.1. Ogólny opis działania urządzenia.....	13
2.2. Elektrody.....	14
2.3. Wybór wersji standardu Bluetooth.....	14
2.4. Sposób przesyłu sygnału za pomocą standardu Bluetooth	15
2.4.1. Rozdzielcość sygnału	16
2.5. Filtracja	16
2.6. Sekcja zasilania	17
2.6.1. Poziom napięcia zasilania	18
3. Projekt urządzenia.....	19
3.1. Wstępny podział budżetu przeznaczonego na prototyp.....	19
3.2. Blok przedwzmacniacza.....	19
3.2.1. Projekt bloku przedwzmacniacza.....	22
3.3. Blok filtracji.....	23
3.3.1 Schemat bloku filtracji (filtr górnoprzepustowy, filtr dolnoprzepustowy).....	23
3.4. Blok II stopnia wzmacnienia	25
3.4.1. Schemat bloku II stopnia wzmacnienia	27
3.5. Blok mikrokontrolera	29
3.5.1. Projekt bloku mikrokontrolera	31
3.6. Blok zasilania.....	33
3.6.1. Projekt bloku zasilania.....	36
4. Badanie symulacyjne układu	38
4.1. Symulacja bloku zasilania	38
4.2. Symulacja bloku przedwzmacniacza.....	39
4.3. Charakterystyka amplitudowo-częstotliwościowa	41
4.4. Symulacja wzmacnienia	42
4.5. Symulacja CMRR.....	45
5. Oprogramowanie.....	47
5.1. Opis protokołu komunikacji	47
5.1.1. Protokół komunikacji pomiędzy aplikacją a urządzeniem.....	47
5.1.2. Aplikacja komputerowa.....	48
5.1.3. Ostateczna wersja aplikacji	50
5.2. Oprogramowanie mikrokontrolera	52
5.2.1. Odbiór danych wejściowych	52
5.2.2. Algorytm przesyłu danych.....	53
5.3. Działanie urządzenia w sieci MESH	53
5.4. Komunikacja z potencjometrem cyfrowym MCP40D18T-104E/LT	55
6. Prototyp urządzenia	57
6.1. Stworzenie projektu obwodu drukowanego	57
6.2. Montaż komponentów	58
6.2.1. Napotkane problemy w realizacji fizycznej.....	60

6.3. Kosztorys poszczególnych bloków	60
6.4. Projekt obudowy urządzenia	62
6.5. Pierwsze uruchomienie układu.....	64
7. Testy urządzenia	66
7.1. Test pomiaru napięcia zasilania	66
7.2. Pomiar charakterystyki amplitudowo-częstotliwościowej	66
7.2.1. Filtr górnoprzepustowy	66
7.2.2. Filtr dolnoprzepustowy.....	68
7.3. Badanie CMRR w układzie.....	70
7.4. Pomiar szumów własnych w układzie	70
7.5. Test zasięgu transmisji układu	72
7.6. Pomiar sygnału elektromiograficznego.....	72
7.7. Testy oprogramowania.....	74
7.7.1. Test poprawności zapisu danych oraz odczytu przez aplikację.....	74
7.7.2. Test kolejności odbioru ramek notyfikacyjnych przez urządzenie	75
7.7.3. Test poprawności przesyłu oraz zapisu danych do pliku.....	77
Podsumowanie	78
Bibliografia	80
Spis rysunków	82
Spis tabel	84
Spis załączników	85

Wstęp

Wzmacniacze sygnałów bioelektrycznych należą do jednych z najczęściej używanych rodzajów aparatury medycznej. Dane otrzymane za ich pomocą znajdują wykorzystywanie w różnych formach diagnostyki medycznej. Na ich podstawie często jesteśmy w stanie stwierdzić potencjalne zagrożenia dla zdrowia bądź życia. Urządzenia te zazwyczaj cechują się wysoką dokładnością pomiaru, jak również bezpieczeństwem użytkowania. Wzmacniacze sygnałów bioelektrycznych, również często wykorzystywane w sporcie, a także w preżnie rozwijającej się branży fitness są smartwatche, smartbandy czy trening trackery to tylko niektóre z urządzeń bazujących na pomiarze parametrów biologicznych. Wymienione rozwiązania umożliwiają znaczne zwiększenie efektywności treningu oraz ułatwiają codzienną profilaktykę zdrowotną. Oprócz informacyjnej roli (przekazywanie czytelnych informacji biomedycznych, dotyczących sygnałów bioelektrycznych, pomagających w zrozumieniu procesów biologicznych), wzmacniacze sygnałów bioelektrycznych używane są również w urządzeniach z obszaru biometrii czy robotyki, gdzie dane zebrane za ich pomocą umożliwiają rozszerzenie funkcjonalności ich działania.

Praca dyplomowa składa się ze wstępu, części głównej złożonej z 7 rozdziałów, podsumowania, bibliografii oraz spisu rysunków, tabel i załączników rozdziale pierwszym dokonano identyfikacji pojęć oraz terminów wykorzystywanych w pracy, jak również przedstawiono parametry badanego sygnału elektromiograficznego oraz podstawowe informacje dotyczące wzmacniaczy biologicznych rozdziale drugim przeprowadzono analizę rozwiązań konstrukcyjnych rozdziale trzecim przeprowadzono szczegółową analizę konstrukcyjną dotyczącą bloku przedwzmacniacza, mikrokontrolera, filtracji, II stopnia wzmacnienia oraz zasilania rozdziale czwartym dokonano symulacji komputerowej projektowanego wzmacniacza w programie Lt-Spiece. W rozdziale piątym przedstawiono oprogramowanie mikrokontrolera, opracowaną aplikację do komunikacji z urządzeniami obsługującymi protokół komunikacji BLE oraz pokazano możliwość pracy urządzenia w sieci BueNRG-MESH.W rozdziale szóstym przedstawiono proces realizacji prototypu urządzenia wraz z całkowitym kosztorysem urządzenia rozdziale siódmym przedstawiono testy układu oraz akwizycje sygnału elektromiograficznego za pomocą zaprojektowanego wzmacniacza. Przeprowadzono również testów oprogramowania.W końcowej części pracy zamieszczone są podsumowanie i wnioski wynikające z realizacji pracy oraz perspektywy dalszego rozwoju urządzenia.

1. Wstęp teoretyczny

1.1. Identyfikacja podstawowych pojęć i terminów wykorzystywanych w pracy

W niniejszym rozdziale w sposób syntetyczny przedstawiono najważniejsze pojęcia, które związane są z tematyką pracy.

Bill of Materials (zestawienie materiałów). Jest to dokument zawierający informacje dotyczące komponentów potrzebnych do zrealizowania projektu. Jego prawidłowe wykonanie znacznie zmniejsza ryzyko uchybień w procesie realizacji fizycznej układu. W zależności od istoty oraz zakresu projektu rodzaje informacji w nim zawarte mogą się różnić. W zestawieniach materiałów w projektach w dziedzinie elektroniki zazwyczaj umieszcza się takie informacje dotyczące komponentów jak: liczba, nazwa, oznaczenie w układzie, footprint, nazwa producenta, cena, dostępność oraz dystrybutor [1, 2].

Bluetooth – to standard bezprzewodowej komunikacji o krótkim zasięgu. Korzysta on z fal radiowych o częstotliwości 2.4 GHz. Umożliwia dostosowanie wielu parametrów transmisji, dzięki czemu jest bardzo często wykorzystywany w projektach z różnych dziedzin. Od wersji 4.0 dzieli się na wersje [3,4]:

- BR - Basic Rate, czyli nominalną wersję bluetooth,
- LE - Low Energy, czyli wersję, która oferuje niski pobór mocy, przy odpowiednio zredukowanym zasięgu działania.

BIN-File – format reprezentacji kodu maszynowego, zawierający informacje dotyczące obszaru pamięci oraz gdzie powinien zostać wgrany kod do pamięci mikrokontrolera. Ponadto cechuje się on niewielkim rozmiarem [5].

HEX-File – Format wprowadzony przez firmę Intel, który służy do przechowywania w formie szesnastkowej kodu maszynowego. Kod w takiej postaci może być wgrany do pamięci EEPROM oraz ROM mikrokontrolera dzięki zastosowaniu programatora [6].

BlueNRG-1 ST-LINK Utility – jest to program umożliwiający obsługę pamięci mikrokontrolerów z rodziny BlueNRG. Za jego pomocą, można również porównywać ze sobą pliki typu HEX oraz BIN z zawartością pamięci mikrokontrolera, a także uruchamiać programy w celu sprawdzania wartości poszczególnych rejestrów [7].

Wise-Studio (STMicroelectronics) – środowisko bazujące na Eclipse IDE dedykowane dla mikrokontrolerów z serii BlueNRG. Umożliwia ono debugowanie kodu w czasie rzeczywistym, sprawdzenie zawartości rejestrów układu oraz wygenerowanie pików HEX i BIN, które pozwalają na wgranie programów do pamięci mikrokontrolera [8].

Programator ST-LINKV2 – to połączenie programatora oraz debuggera przeznaczone do komunikacji z mikrokontrolerami STM8, STM32 przy pomocy interfejsów SWIM oraz SWD [9]. Na rysunku 1.1. przedstawiono zdjęcie programator ST-Link V2.



Rys. 1.1. Programator ST-Link V2

I²C – jest to interfejs umożliwiający komunikację pomiędzy urządzeniem kontrolującym Master a urządzeniem odbierającym Slave. Składa się z dwóch linii:

- SCL-Serial Clock,
- SDA-Serial Data.

Obie linie należy ustawić w stanie wysokim za pomocą rezystora podciągającego (z ang. Pull up Resistor) [10].

RSSI (Recived Signal Strenght Indicator) – jest to parametr wskazujący jakość sygnału radiowego otrzymywanego przez urządzenie. Wartość tą podaje się w jednostkach dBm. Im wartość jest bliższa zeru, tym jakość sygnału jest lepsza. Poniżej przedstawiono poglądowe wartości RSSI:

- -30 dBm – bardzo dobra jakość sygnału, zazwyczaj osiągalna jedynie w idealnych warunkach laboratoryjnych,
- -67dBm – dobra jakość sygnału, pozwalająca na stabilną komunikację,
- -80dBm – zła, jakość sygnału może zachodzić problem z przesyłaniem danych,
- -90dBm – bardzo zła jakość sygnału, posiadająca wysoki poziom szumów [11].

BlueNRG-MESH – jest to system umożliwiający łączenie licznych urządzeń BLE (Bluetooth Low Energy) w strukturę Mesh dla rozwiązań IoT (Internet of Things). Pozwala na przekształcenie urządzeń wyposażonych w protokół komunikacji Bluetooth w wersji BLE w zintegrowaną sieć Mesh o rozszerzonym zasięgu, jak również komunikacji dwukierunkowej. Dostępny jest on dla produktów z rodziny BlueNRG [12,13,14].

1.2. Charakterystyka sygnału elektromiograficznego

Wartości amplitudy sygnału EMG mogą wynosić między $\pm 5000 \mu\text{V}$ [15, s. 10]. Rejestrowane wartości sygnału mogą różnić się zależności od typu badania, miejsca akwizycji sygnału czy też aktywności pacjenta. Dla sportowców otrzymywane są większe wartości sygnału EMG. Zwykle wartości sygnału mieszczą się w przedziale od 6 Hz do 500 Hz, natomiast większość mocy sygnału przypada na przedział pomiędzy 20 Hz a 150 Hz [15].

1.2.1. Wzmacniacz biologiczny

Wzmacniacz biologiczny to urządzenie umożliwiające akwizycję słabych, wzmacnienie oraz odpowiednią modyfikację słabych sygnałów biologicznych. Do najważniejszych cech takich urządzeń należą [16, 17, 18]:

- wysoka czułość,
- bezpieczeństwo użytkowania,

- prostota budowy umożliwiająca zminimalizowanie ryzyka wystąpienie błędu ludzkiego w procesie projektowania oraz użytkowania a także nieprawidłowości w funkcjonowaniu komponentów,

Cel pracy

Celem niniejszej pracy jest zaprojektowanie oraz wykonanie układu do bezprzewodowej akwizycji sygnałów elektromiograficznych.

2. Analiza rozwiązań konstrukcyjnych

W poniższym rozdziale przedstawiono wymagania oraz założenia projektowe dotyczące projektowanego wzmacniacza, opis funkcjonowania urządzenia, jego schemat blokowy oraz analizę rozwiązań konstrukcyjnych dotyczących całego układu.

Wobec projektowanego urządzenia przedstawione zostały następujące wymagania podstawowe:

- 1) Urządzenie powinno być zasilane baterijnie.
- 2) Bezprzewodowa komunikacja powinna zostać zrealizowana w oparciu o standard Bluetooth.
- 3) Urządzenie powinno zapewnić rozdzielcość sygnału wejściowego na poziomie min. 10 bit.
- 4) Powinien zostać zdefiniowany protokół komunikacji służący do transmisji komend sterujących do urządzenia oraz danych wyjściowych z urządzenia, jak również format danych wyjściowych.
- 5) Urządzenie powinno być przystosowane do pracy w sieci minimum dwóch tego rodzaju urządzeń.
- 6) Urządzenie powinno posiadać obudowę.
- 7) Dokumentacja techniczna powinna obejmować dokumentację obwodu drukowanego, dokumentację projektu obudowy, dokumentację protokołu komunikacji oraz krótką instrukcję obsługi.

Poniżej przedstawiono sformułowane założenia projektowe:

- 1) Maksymalny koszttworzenia prototypu nie powinien przekroczyć 110 zł, nie wliczając w to kosztów związanych z przesyłką poszczególnych elementów układu drukowanego PCB.
- 2) Szumy własne wzmacniacza w stosunku do sygnału wejściowego $<5 \mu\text{V}$.
- 3) Docelowy rozmiar układu nie powinien przekraczać wymiarów 5 cm x 5 cm. Urządzenie to ma umożliwiać akwizycję sygnału pochodzącego z różnych mięśni pacjenta, stąd narzucono określone wymiary, tak aby móc w komfortowy sposób je użytkować. Ponadto sam koszt tworzenia układu PCB zazwyczaj zależy od jego rozmiaru
- 4) Wartość CMRR $> 80 \text{ dB}$ dla projektowanego wzmacniacza (wartości rekomendowana przez SENIAM (Surface ElectroMyoGraphy for the Non-Invasive Assessment of Muscles) wynosi $> 95 \text{ dB}$, niemniej jednak wartość 80 dB przyjmuje się często za minimalną akceptowalną wartość).
- 5) Wzmocnienie regulowane w zakresie od 650 - 7500 z dokładnością do 1%. I stopień wzmocnienia 25 V/V, II stopień wzmocnienia 25V/V - 300V/V.
- 6) Zakres amplitud wejściowych sygnału od $10\mu\text{V}$ do 5mV .
- 7) Zakres częstotliwości analizowanego sygnału elektromiograficznego ustalono na przedział $5 \text{ Hz} - 255 \text{ Hz} \pm 20 \%$.

Ogólnym założeniem dotyczącym konstruowanego urządzenia było wykonanie taniego, przenośnego wzmacniacza o niewielkich rozmiarach służącego do doraźnej akwizycji sygnału z różnych miejsc ciała pacjenta.

Na wstępie wyróżniono dwa podejścia do procesu projektowania, które determinują sposób pracy nad układem, tj. podejście holistyczne i podejście blokowe [22]. W tabeli 2.1. przedstawiono porównanie dwóch podejść do procesu projektowego.

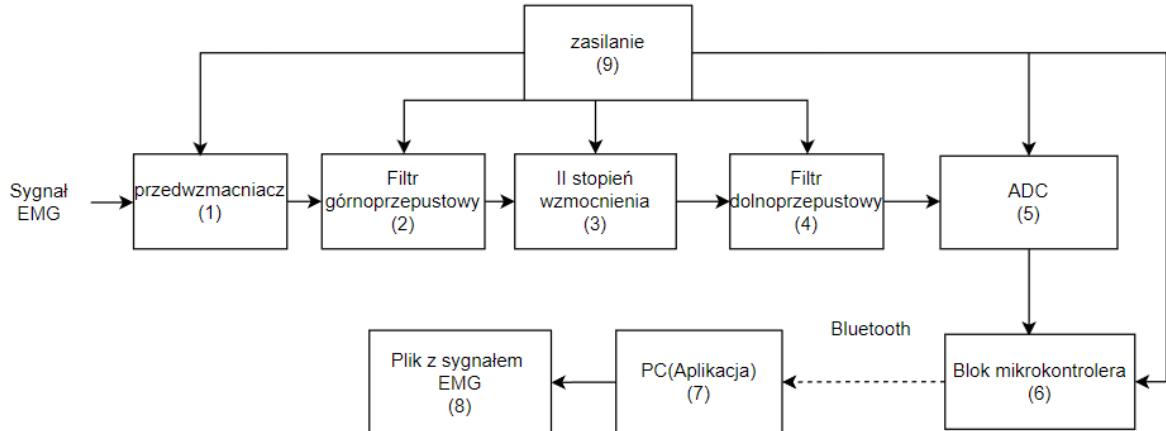
Tab. 2.1. Porównanie podejść do procesu projektowego

Kryteria porównania	Podejście holistyczne	Podejście blokowe
Realizacja projektu	Realizacja projektu odbywa się w sposób całościowy bez podziału na mniejsze części.	Projekt realizuje się częściowo poprzez podział na mniejsze części w postaci bloków, które następnie łączy się ze sobą.
Wymagana wiedza	Od projektanta wymagana jest fachowa wiedza nabyta poprzez wielokrotne realizowanie projektów w danej dziedzinie.	Wymagana wiedza może być zazwyczaj zdobyta w trakcie studiów. Pozwala na dalszą naukę oraz realizację praktyczną nabytej wiedzy teoretycznej.
Czas wymagany do realizacji projektu	Krótszy ze względu na możliwość wdrożenia wielokrotnie przetestowanych rozwiązań	Dłuższy ze względu na konieczność zaprojektowania oraz weryfikacji poprawności działania każdego z bloków.
Koszt realizacji	Niższy	Wyższy

Analizując dane zawarte w tabeli 2.1. dotyczące dwóch podejść do procesu projektowania zdecydowano się na wybór blokowego procesu projektowania wzmacniacza. Wybór ten został dokonany ze względu na niewielkie doświadczenie w zakresie projektowania tego typu urządzeń.

2.1. Ogólny opis działania urządzenia

Na rysunku 2.1. przedstawiono schemat blokowy projektowanego systemu do akwizycji sygnału elektromiograficznego.



Rys. 2.1. Schemat projektowanego systemu do akwizycji sygnału EMG

Całość urządzenia zasilana będzie w sposób baterijny (9). Przewidywany czas ciągłej pracy urządzenia wynosił będzie powyżej 6 h. Sygnał EMG podlega najpierw wzmacnieniu przez przedwzmacniacz (1). Blok przedwzmacniacz charakteryzuje się wysokim CMRR oraz wysoką impedancją wejściową umożliwiającą mu bezpośrednią akwizycję sygnału z elektrod.

Wstępnie wzmacniony sygnał poddawany jest filtracji górnoprzepustowej (2) a następnie jest wzmacniany przez układ wzmacniacza II stopnia (3). W dalszej kolejności sygnał poddawany jest filtracji dolnoprzepustowej (4) oraz konwersji analogowo-cyfrowej przez ADC

(5). Aplikacja desktopowa (7) służy do komunikacji z urządzeniami BLE. Umożliwia zapis odebranego sygnału wysyłanego przez mikrokontroler (6) do pliku (8).

2.2. Elektrody

Rozważono jedynie elektrody stosowane do powierzchniowej akwizycji sygnałów bioelektrycznych (tabela 2.2.).

Tab. 2.2. Porównanie dwóch rodzajów elektrod rozważanych do zastosowania w powierzchniowej akwizycji sygnału EMG w projektowanym wzmacniaczu

Kryteria porównania	Elektroda sucha	Elektroda mokra
Obecność żelu	Brak	Obecny
Dyskomfort badania	Brak większych czynników mogących powodować uczucie dyskomfortu u osoby badanej	Obecność żelu może powodować uczucie dyskomfortu u pacjentów
Impedancja styku	Wysoka	Niska
Jakość odbieranego sygnału	Mogliwa gorsza jakość sygnału ze względu na wysoką impedancję elektroda-tkanka	Wysoka, ze względu na niską impedancję elektroda-tkanka
Koszt	Wyższy niż w przypadku elektrod mokrych	Niski
Przygotowanie obszaru skóry do badania	Minimalna konieczność przygotowania	Mogliwa konieczność dokładniejszego przygotowania obszaru skóry poprzez m.in. usunięcie owłosienia
Podatność na artefakty ruchowe	Mogliwa do ograniczenia poprzez zastosowanie m.in. zewnętrznych powierzchni kleistych dociskających i zapobiegających potencjalnym przemieszczaniu się elektrod.	Niska, ograniczona dodatkowo poprzez zastosowane powierzchnie kleiste.

Na podstawie danych przedstawionych w tabeli 2.2. zdecydowano się na wykorzystanie elektrod mokrych. Cechują się niższą impedancją na powierzchni elektroda-tkanka ze względu na obecność żelu przewodzącego, dzięki czemu jakość sygnału odbieranego za ich pomocą jest zazwyczaj wyższa niż w przypadku elektrod suchych. Ponadto niska podatność na artefakty ruchowe może być istotna ze względu na mobilny charakter urządzenia.

Natomiast zachodzi konieczność uważanego monitorowania warstwy żelu ze względu na jego wchłanianie, która może doprowadzić do zmiany impedancji sygnału na powierzchni elektroda-tkanka, co z kolei może prowadzić do pogorszenia się jakości sygnału.

Rozważono również połączenie elektrod z układem. Zdecydowano się na zastosowanie krótkiego nie przekraczającego 20 cm przewodu ekranowanego.

2.3. Wybór wersji standardu Bluetooth

Jednym z podstawowych wymagań postawionych przed urządzeniem było zapewnienie bezprzewodowej komunikacji która powinna być oparta o standard Bluetooth. W tabeli 2.3. porównano dwa standardy protokołu Bluetooth.

Tab. 2.3. Porównanie dwóch standardów protokołu Bluetooth

Kryteria porównania	Bluetooth Low Energy (BLE)	Bluetooth standard
Zużycie energii	Niskie, zaprojektowany z myślą o urządzeniach działających długim okresem czasu (do kilku m-cy)	Wyższe, niż w przypadku BLE
Zasięg	Do 100 m	Ponad 100 m
Liczba możliwych połączń do nawiązania między urządzeniami	Możliwość stworzenia sieci MESH/MMDL od kilku do kilkunastu urządzeń (zależna od implementacji)	Mniejsza liczba możliwych jednoczesnych połączeń niż w przypadku sieci MESH/MMDL
Czas reklamowania/połączenia	Krótki (kilka ms)	Dłuższy, niż w przypadku BLE

Analizując dane przedstawione w tabeli 2.3. zdecydowano się na wybór wersji BLE. Protokół ten umożliwia znaczne ograniczenie zużycia energii, dzięki czemu można będzie osiągnąć planowany czas pracy układu. Ponadto oferuje gotowe rozwiązania (MESH/MMDL), których implementacja zapewni zdolność urządzenia do pracy w sieci podobnych urządzeń. Oferowany zasięg oraz moc transmisji jest wystarczająca na potrzeby projektu o charakterze bezprzewodowym.

2.4. Sposób przesyłu sygnału za pomocą standardu Bluetooth

W tabeli 2.4. porównano dwa możliwe rozwiązania przesyłu sygnału za pomocą BLE, które brano pod uwagę podczas wstępnej analizy rozwiązań konstrukcyjnych. Głównymi kryteriami wybory były:

- łatwość w implementacji,
- niewielki koszt,
- możliwość zapoznania się z funkcjonowaniem układu przed przystąpieniem do realizacji fizycznej układu.

Tab. 2.4. Porównanie rozwiązań w zakresie przesyłu sygnału

Kryteria porównania	Mikrokontroler ze zintegrowanym modułem Bluetooth	Moduł Bluetooth
Wydajność	Wysoka wydajność ze względu na zintegrowane rozwiązanie, zoptymalizowane do współpracy pomiędzy układami (MCU, moduł BLE)	Mogą wystąpić problemy z dopasowaniem oraz kompatybilnością układów, co skutkować może koniecznością zastosowania dodatkowych komponentów pomiędzy mikrokontrolerem a modułem
Możliwość pracy w standardzie BLE	Tak	Tak
Rozmiar	Zależny od obudowy układu	Zależny od obudowy układów
Koszt*	Nie przekraczający kilkudziesięciu złotych	Nie przekraczający kilkudziesięciu złotych
Możliwość zapoznania się z funkcjonowaniem układu przed fizyczną realizacją urządzenia	Tak, w postaci układu deweloperskiego zwierającego mikrokontroler wraz z różnymi modułami demonstrującymi jego funkcjonalność	Tak, poprzez montaż na płytce stykowej mikrokontrolera oraz modułu w wersji PDIP

*Brano pod uwagę jedynie układy, których cena mieści się w założonym budżecie.

Na podstawie tabeli 2.4. zdecydowano się, że w projekcie komunikacja za pomocą protokołu BLE będzie realizowana za pomocą mikrokontrolera z wbudowanym modułem BLE. Takie zintegrowane rozwiązanie pozwala na znaczne ułatwienie tworzenia protokołu komunikacji ze względu na brak konieczności konfigurowania obu urządzeń do pracy wspólnej. Dodatkowo za wyborem rozwiązania przemawia wcześniejsze doświadczenie w pracy z tego typu układami.

2.4.1. Rozdzielcość sygnału

Jednym z podstawowych wymagań, postawionym przed projektowanym układem, jest zapewnienie rozdzielcość sygnału wejściowego na poziomie min. 10 bit. W tabeli 2.5. przedstawiono dwa możliwe rozwiązania cyfrowej części układu.

Tab. 2.5. Porównanie dwóch możliwych realizacji cyfrowej części układu

Kryteria porównania	Mikrokontroler z wbudowanym przetwornikiem ADC	Połącznie zewnętrznego przetwornika ADC oraz mikrokontrolera
Rozdzielcość	Niższa, niż w przypadku zewnętrznego modułu ADC. Jednakże większość obecnie dostępnych na rynku MCU oferuje rozdzielcość na poziomie 10 bit-12 bit.	Wysoka, możliwe uzyskanie 24 bitów rozdzielcości sygnału
Rozmiar	Zależny od obudowy układu	Zależny od obudów układów
Koszt	Nie przekraczający kilkudziesięciu złotych*	Niski*
Trudność w realizacji	Niska, jest to zintegrowane rozwiązanie	Wyższa, ze względu na konieczność dopasowania komponentów
Doświadczenie w pracy z rozwiązaniami	Tak	Nie

*Brano pod uwagę jedynie układy, których cena mieści się w założonym budżecie.

Na podstawie danych przedstawionych w tabeli 2.5. zdecydowano się wykorzystać mikrokontroler z wbudowanym przetwornikiem ADC. Za wyborem tego rozwiązania przemawiało głównie wcześniejsze doświadczenie w pracy z tego typu układami. Mikrokontroler z wbudowanym przetwornikiem ADC jest w stanie zapewnić wymaganą rozdzielcość sygnału wejściowego.

2.5. Filtracja

Przed rozpoczęciem rozważań dotyczących konstruowania bloków filtrujących porównano główne cechy filtrów pasywnych i aktywnych (tabela 2.6.).

Tab. 2.6. Porównanie głównych cech filtrów pasywnych i aktywnych

Kryteria porównania	Filtr pasywny	Filtr aktywny
Koszt	Niski, związany jedynie z zakupem komponentów pasywnych	Wyższy koszt ze względu na obecność komponentów aktywnych
Komponenty	Składa się z komponentów pasywnych takich, jak cewki, kondensatory oraz rezystory	Składa się ze wzmacniaczy operacyjnych oraz komponentów pasywnych takich, jak cewki, kondensatory oraz rezystory
Zajmowana powierzchnia w układzie	Stosunkowo niewielka, gdyż zależna jest jedynie od wielkości obudów elementów pasywnych	Większa, gdyż zależna jest od wielkości obudów komponentów aktywnych oraz pasywnych.
Trudność w zaprojektowaniu	Niska	Wysoka, jednakże możliwość zastosowania oprogramowania do projektowania filtrów znacznie ułatwia ten proces
Wpływ na tłumienie sygnału w paśmie przepustowym	Ostrość filtracji, oraz spadek na dekadę z reguły mniejszy niż w przypadku filtrów aktywnych	Spadek na dekadę wysoki.

Ze względu na mniejszy koszt realizacji oraz kompaktowość zdecydowano się na wybór filtrów pasywnych.

2.6. Sekcja zasilania

Jednym z podstawowych założeń stawianych przed projektowanym układem jest konieczność zrealizowania zasilania w sposób baterijny. Ponadto powinien zostać zapewniony przynajmniej 6 h czas działania układu. W tabeli 2.7. przedstawiono rozważane sposoby otrzymania napięcia symetrycznego w projektowanym układzie.

Tab. 2.7. Porównanie sposobu otrzymania napięcia symetrycznego w układzie wzmacniacza

Kryteria porównania	Wirtualna masa	DC/DC
Koszt	Zależny od wyboru układów	Konwertery DC/DC są zazwyczaj droższe niż LDO, co może przełożyć się na wyższy koszt realizacji.
Wcześniejsza realizacja konfiguracji sekcji zasilania za pomocą komponentów	Tak	Nie
Zakłócenia	Zazwyczaj niższe niż przy użyciu konwertera DC/DC	Zastosowanie konwerterów DC/DC może wprowadzić wyższy poziom szumów w układzie oraz konieczność filtracji zakłóceń, szczególnie w układach o małych wymiarach.
Kompleksowość rozwiązania	Niższa	Wyższa

Na podstawie analizy danych z tabeli 2.7. zdecydowano się na wybór zasilania wykorzystującego akumulator/ baterię w połączeniu z linowym regulatorem napięcia oraz VGND, ze względu na doświadczanie w pracy z układami wchodzącyymi w skład konfiguracji oraz niższy koszt komponentów wymaganych do realizacji rozwiązania.

W dalszej części przeprowadzono analizę wykonania układu wirtualnej ziemi (VGND). Układ VGND zaprojektowano w oparciu o notę katalogową „A Designer ‘s Guide to

Instrumentanion Amplifires 3rd Edition”[19]. Rozwiążanie pozwala na uzyskanie niskiej impedancji na wyjściu układu, dzięki czemu będzie można nadać wirtualna masę bezpośrednio na piny kompletów aktywnych (wzmacniacz instrumentalny, wzmacniacz operacyjny), co zmniejsza złożoność projektowania układu oraz pozwala na oszczędność budżetu.

2.6.1. Poziom napięcia zasilania

Poziom napięcia zasilania ustalono na 3.3 V. Jest on kompatybilny z większością komponentów (z ang. Low Power), które zaplanowano wykorzystać w celu bardziej ekonomicznego uzyskania założonego czasu pracy układu na baterii. Ponadto mikrokontrolery obsługujące protokół komunikacji BLE mają w większości napięcie zasilania wynoszące 3.3 V-3.6V, np.: mikrokontrolery z serii ARM CORTEX, co niweluje potrzebę dodatkowej konwersji poziomu napięcia.

3. Projekt urządzenia

W rozdziale tym przedstawiono szczegółowe rozważania dotyczące procesu projektowania poszczególnych bloków.

3.1. Wstępny podział budżetu przeznaczonego na prototyp

Jednym z założeń projektowych jest ograniczony koszt wytworzenia układu. Z tego powodu na potrzeby dalszych etapów projektowania przenalizowano wpływ poszczególnych bloków układu na proces akwizycji sygnału. Pozwoli to na odpowiednie dopasowanie wielkości budżetu przypadającego na wyróżnione bloki.

W tabeli 3.1. przedstawiono proponowany procentowy budżet przypadający na każdy z bloków wraz z uzasadnieniem wartości.

Tab. 3.1. Proponowany podział budżetu

Nazwa	Ustalony procent budżetu przypadający na blok	Uzasadnienie dokonanego podziału
Blok filtracji (filtr dolnoprzepustowy, górnoprzepustowy)	≈5%	Załожona 20% tolerancja dotycząca wartości tolerancji częstotliwości odcienia umożliwia wykorzystanie niedrogich komponentów w celu zmieszczenia się w założonym przedziale. Ponadto wybór filtrów pasywnych znacznie obniży koszt realizacji tego bloku.
Blok mikrokontrolera	≈20%	Istotny blok projektowanego wzmacniacza odpowiadający za transmisję sygnału za pomocą protokołu komunikacji Bluetooth oraz sterowania cyfrowymi układami we wzmacniaku.
Blok przedwzmacniacza	≈30%	Pierwszy blok wchodzący w skład toru pomiarowego. Jego odpowiednie zaprojektowanie jest kluczowe w celu zarejestrowania wysokiej jakości sygnału EMG. Dlatego wybrane układy powinny cechować się bardzo dobrymi parametrami. Ponadto do bloku może należeć również podkład „Right Leg Drive”.
Blok wzmacnienia II stopnia	≈15%	Istotny element toru pomiarowego wzmacniacza. Ze względu na charakterystykę sygnału należy zastosować w nim regulowane wzmacnianie. Ustalony procent budżetu pozwoli na zakup cyfrowego układu do regulacji wzmacniania.
Blok zasilania	≈30%	Główny koszt tego bloku związany będzie z zakupem źródła zasilania. Nie są one drogie szczególnie dla układów o niskim poborze energii tak, jak projektowany wzmacniacz. Ponadto w skład tego bloku wchodzi jeszcze układ wirtualnej ziemi.

3.2. Blok przedwzmacniacza

Rozważono dwa podejścia do wykonania układ przedwzmacniacza (tabela 3.2.):

- zintegrowany wzmacniacz instrumentalny,
- wykonanie wzmacniacza instrumentalnego za pomocą układu wzmacniaczy operacyjnych oraz elementów pasywnych.

Tab. 3.2. Porównanie konstrukcji wzmacniacza instrumentalnego

Kryteria porównania	Zintegrowany wzmacniacz instrumentalny	Układ wzmacniacza instrumentalnego złożony z wzmacniaczy operacyjnych
Komponenty występujące w układzie	Zintegrowany układ wzmacniacza instrumentalnego	Dwa lub trzy wzmacniacze operacyjne oraz rezystory
Współczynnik CMRR	Wysoki	Wysoki
Koszt	Zależny od wybranego układu	Zależny od wybranych komponentów
Rozmiar	Niewielki	Zależy od liczby oraz obudów komponentów wykorzystanych w konstrukcji
Impedancia wejściowa	Bardzo wysoka	Bardzo wysoka
Poprawność rozwiązania	Rozwiązanie wielokrotnie sprawdzone przez producenta	Rozwiązanie opracowywane w procesie projektowania wzmacniacza wymaga testowania oraz korekcji potencjalnych błędów

Reasumując, na podstawie analizy danych w tabeli 3.2. można stwierdzić, iż oba rozwiązania posiadają podobne parametry oraz mogą być użyte w projektowanym wzmacniaczu. Jednakże ze względu, iż zintegrowane wzmacniacze instrumentalne są rozwiązaniem wielokrotnie testowanym a ich implementacja w układzie jest znacznie prostsza, dlatego też zdecydowano się na ich wybór w bloku przedwzmacniacza. Za ich wykorzystaniem przemawia również doświadczenie w pracy z tego typu układami.

W dalszej kolejności dokonano wyboru wzmacniacza instrumentalnego. W tym celu w tabeli 3.3. zestawiono wzmacniacze instrumentalne, które spełniają narzucone im następujące kryteria:

- minimalne napięcie wymagane do zasilenia układu ≤ 3.3 V,
- $CMRR \geq 100$ dB,
- niewielki rozmiar (ekwiwalent SOIC-8 lub mniejszy),
- cena nie przekraczająca 30 zł,
- typ montażu SMD.

Tab. 3.3. Charakterystyka porównawcza wzmacniaczy instrumentalnych do wykorzystania w bloku przedwzmacniacza

Kryteria porównania	AD623ARZ	INA326EA/2K5	595-INA337AIDGKT
Producent	Analog Devices	Texas Instruments	Texas Instruments
Dostępność w ofercie międzynarodowych dostawców elektroniki	Tak	Tak	Tak
Obudowa	SOIC-8	VSSOP-8	SOIC-8
Napięcie zasilania	2.7 V - 12 V	2.5 V - 5.5 V	2.7 V-5.5 V
Typowy prąd roboczy zasilania	375 μ A	2mA	2.4 mA
Koszt	15.80*zł	22.51 zł	27.01 zł
CMRR	105 dB	100 dB	106 dB

*Cena zakupu na stronie TME

Dane pochodzą z not katalogowych układów oraz od dystrybutorów elektroniki Mouser i DigiKey.

Na podstawie analizy danych (tabela 3.3.) dotyczących rozpatrywanych wzmacniaczy instrumentalnych przyjęto, iż układ AD623 będzie w najlepszym stopniu spełniał przyjęte założenia. Układ ten cechuje się wysoką wartością CMRR wynoszącą 105 dB oraz bardzo niskim poborem prądu wynoszącym 375 μ A.

Wejście referencyjne wzmacniacza pomiarowego wymaga utrzymania niskiego poziomu impedancji wejściowej ze względu na następujące czynniki: [19]

- 1) Wysoka impedancja wprowadzana na punkty referencyjne szczególnie w przypadku wzmacniacza instrumentalnego obniża zdolność układu do tłumienia sygnału wspólnego.
- 2) Utrzymanie niskiego poziomu impedancji na wyjściu układu referencji zapewnia poprawne przenoszenie poziomu napięcia referencyjnego w układzie.
- 3) Wysoka impedancja może spowodować zmniejszenie stabilności układu, co może objawiać się poprzez wystąpienie niepożądanych oscylacji.

Na wejście referencyjne wzmacniacza zdecydowano się podać VGND. Wirtualna masa posiada niską impedancję.

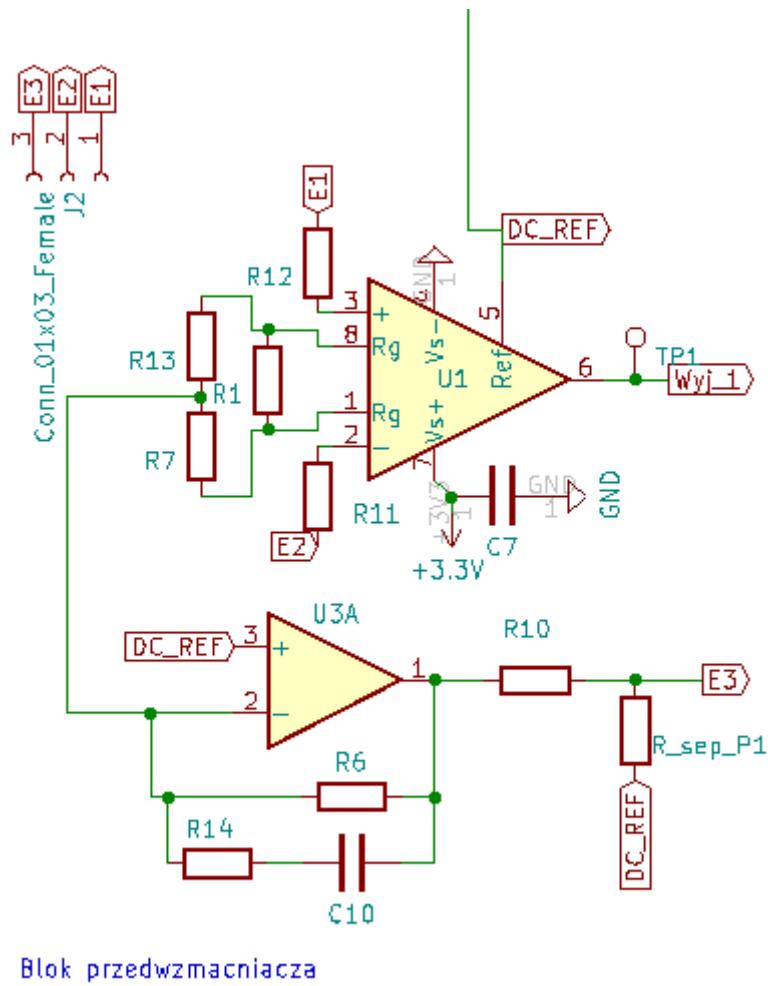
W dalszej części przeprowadzono analizę wykonania układu sprzężenia zwrotnego RLD (Right Leg Drive). Zastosowanie tego układu umożliwi uzyskanie lepszej jakości sygnału przez zwiększenie zdolności układu do tłumienia zakłóceń wynikających z napięcia wspólnego.

Układ RLD został zaprojektowany w oparciu o notę katalogową układu AD620 [20] oraz „A Designer's Guide to Instrumentation Amplifiers, 3rd Edition”[19].

Zdecydowano się na wykorzystanie wybranego do realizacji układu VGND do podwójnego wzmacniacza operacyjnego AD8538ARZ. Układ ten oprócz wysokiej wydajności prądowej wymaganej od wzmacniacza VGND, posiada również wartość CMRR >150dB. Takie rozwiązanie (zastosowanie podwójnego wzmacniacza) upraszcza proces projektowania ze względu na ograniczenie liczby wymaganych komponentów potrzebnych do realizacji prototypu, jak również umożliwia redukcję kosztów oraz poboru prądu, co koreluje z sformowanymi założeniami projektowymi dotyczącymi projektowanego wzmacniacza. W układzie istnieje również możliwość zmiany sygnału podawanego na elektrodę referencyjną: zamiast RLD może być to wirtualna masa urządzenia (VGND)- opcja ta została przewidziana na wypadek trudności w implementacji RLD (rysunek 3.1. R_sep_1).

3.2.1. Projekt bloku przedwzmacniacza

Na rysunku 3.1. przedstawiono schemat zaprojektowanego bloku przedwzmacniacza.



Rys. 3.1. Schemat bloku przedwzmacniacza

W tabeli 3.4. przedstawiono wartości komponentów bloku przedwzmacniacza.

Tab. 3.4. Oznaczenie wartości komponentów bloku przedwzmacniacza

Oznaczenie w układzie	Wartość
C7	0.1 μF [E6 tol. 20%]
C10	0.01 μF [E6 tol. 20%]
R1	4.7 k Ω [E96 tol. 1%]
R6	1 M Ω [E96 tol. 1%]
R7	24.9 k Ω [E96 tol. 1%]
R10	100 K Ω [E96 tol. 1%]
R11	220 k Ω [E96 tol. 1%]
R12	220 k Ω [E96 tol. 1%]
R13	24.9 k Ω [E96 tol. 1%]
R14	10 k Ω [E96 tol. 1%]
R_sep_P1	0 Ω [E96 tol. 1%]

Wartości dobrano na podstawie przykładowej aplikacji układu RLD znajdującej się w nocie katalogowej układu AD620 [20] oraz A Designer's Guide to Instrumentation Amplifiers, 3rd Edition [19].

Zgodnie z przyjętymi założeniami, układ przedwzmacniacza powinien realizować wzmacnianie wzmacniacza AD623ARZ ustalone z zastosowaniem rezystora RG według wzoru:

$$RG = \frac{100\text{k}\Omega}{G-1} \quad (3.1)$$

gdzie:

RG - jest wartością rezystora umieszczonego na pinach RG,

G - wzmacnianie otrzymywane dla wartości rezystora RG.

Rezystancja zastępcza połączenia równoległego rezystora R1 z parą szeregowo połączonych rezystorów R13 oraz R7 wynosi 4294,678 Ω. Stąd teoretyczne wzmacnianie wzmacniacza obliczone za pomocą wzoru 3.1. wynosi 24,2846. Otrzymana różnica procentowa wartości wzmacnienia wynikającego z komponentów użytych do realizacji układu względem wartości wzmacnienia założonej dla I stopnia wzmacnienia wynosi 2,8% i nie mieści się w założonym przedziale dokładności.

3.3. Blok filtracji

Zdecydowano się na buforowanie stopni filtracji poprzez wzmacniacz operacyjny. Ze względu na mniejszą kompleksowość rozwiązania oraz łatwiejsze ograniczenie niepożądanego oddziaływanie stopni filtracji na siebie.

Wzmacniacz, wykorzystany w tym bloku jako bufor, został wybrany w podrozdziale 3.4. Blok II stopnia wzmacnienia. Zastosowanie podwójnego wzmacniacza do realizacji dwóch bloków pozwoli na redukcję kosztów oraz ograniczenie powierzchni zajmowanej przez bloki, a także zmniejszyć kompleksowość układu.

3.3.1 Schemat bloku filtracji (filtr górnoprzepustowy, filtr dolnoprzepustowy)

Projektowanie filtrów rozpoczęto od ustalenia wartości pojemności kondensatorów na typowych wartościach z szeregu. Następnie na podstawie ustalonych wartości kondensatorów oraz częstotliwości granicznych założonego pasma przepustowego wyznaczono wartość rezystorów przy pomocy wzoru:

$$f = \frac{1}{2\pi RC} \quad (3.2)$$

gdzie:

R - wartość dobranego rezystora [Ω],

C - wartość dobranego kondensatora [F],

π - stała pi (przyjęto przybliżenie 3,14)

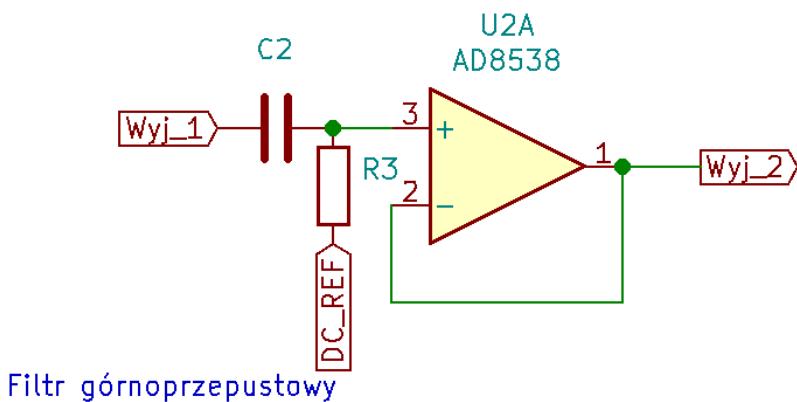
f - wartość częstotliwości [Hz].

Teoretyczne wartości rezystorów W tabeli 3.5. przedstawiono wartości komponentów w bloku filtracji.

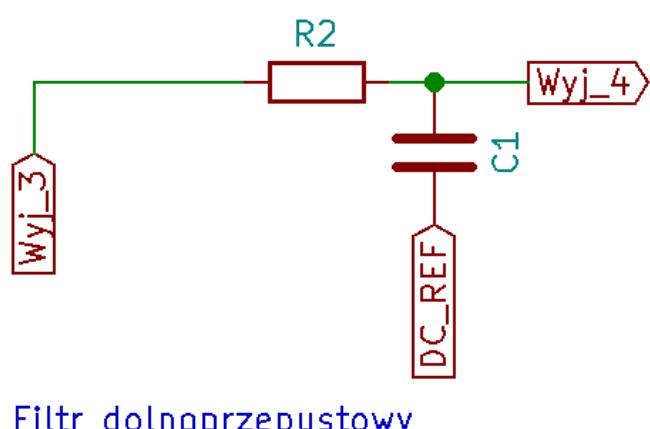
Tab. 3.5. Zestawienie wartości komponentów użytych do realizacji bloku filtracji

	Filtr górnoprzepustowy	Filtr dolnoprzepustowy
Ustalona wartość kondensatora	C2 - 0.47 μ F [E6 tol. 20%]	C1 - 100 nF [E6 tol. 20%]
Wartość rezystora obliczona ze wzoru 3.2	56000 Ω	6366,19 Ω
Wartość rezystora dobrane	R3 - 51000 Ω [E24 tol. 1%]	R2 - 6200 Ω [E96 tol. 1%]
Teoretyczna wartość odcięcia wynikająca z nominalnych wartości komponentów zastosowanych w projekcie	6.639 Hz	256.7 Hz
Czy teoretyczna wartość częstotliwości granicznej wnikająca z nominalnej wartości komponentów zastosowanych w projekcie mieści się w zakładanej tolerancji?	Tak	Tak

Na rysunku 3.2. przedstawiono schemat filtra górnoprzepustowego, natomiast na rysunku 3.3. przedstawiono schemat filtra dolnoprzepustowego.



Rys. 3.2. Schemat filtra górnoprzepustowego



Rys. 3.3. Schemat filtra dolnoprzepustowego

3.4. Blok II stopnia wzmacnienia

Projektowanie bloku wzmacnienia zaczęto od wybrania konfiguracji wzmacniacza. W tabeli 3.6. przedstawiono porównanie dwóch konfiguracji wzmacniaczy operacyjnych.

Tab. 3.6. Porównanie konfiguracji odwracającej i nieodwracającej wzmacniacza operacyjnego

Kryteria porównania	Konfiguracja odwracająca	Konfiguracja nieodwracająca
Wzmocnienie	$-\frac{R_f}{R_i}$	$1 + \frac{R_f}{R_i}$
SNR	Wysokie	Wysokie
Przesunięcie fazowe	180	0
Stabilność	Wysoka	Wysoka
Impedancja wejściowa	Niska	Wysoka

Zdecydowano się na wybór konfiguracji odwracającej. W tabeli 3.7. zestawiono, wzmacniacze operacyjne, które spełniają narzucone im następujące wymagania:

- minimalny napięcie wymagane do zasilania układu ≤ 3.3 V,
- niewielki rozmiar (rozmiary nie przekraczające footprintu SOIC8),
- działanie „rail to rail”,
- współczynnik CMRR > 90 dB,
- niski koszt do 15 zł,
- typ montażu SMD,
- niski pobór prądu < 200 μ A (typowy),
- GBP > 300 kHz.
- podwójny

Tab. 3.7. Zestawienie danych wzmacniaczy operacyjnych do użycia w bloku II stopnia wzmacnienia

Nazwa	AD8538ARZ	MAX9618AXA+T	AS2333QS-13
Dostępność	Tak	Tak	Tak
Koszt	14.99 zł (9.43 zł*)	21.90 zł	14.62 zł
Producent	Analog Devices	Analog Devices	Diodes Incorporated
Napięcie zasilania	2.5 V - 5.5 V	1.8 V - 5.5 V	1.8 V - 5.5 V
Typowy prąd roboczy zasilania	180 μ A	59 μ A	12 μ A
Rozmiar/obudowa	SOIC-8	SC-70-8	SO-8
CMRR	150dB	122 dB	90 dB
GBP	430kHz	1.5 MHz	350kHz
Rail to Rail	Tak	Tak	Tak

*Cena zakupu na stronie TME

Dane zamieszczone w tabeli pochodzą z not katalogowych układów oraz od dystrybutorów elektroniki Mouser i DigiKey.

Na podstawie analizy danych zebranych w tabeli 3.7. wybrano układ AD8538ARZ. Układ ten cechuje się odpowiednim napięciem zasilania, niskim poborem prądu oraz wysokim parametrem CMRR. Ponadto działa „Rail to Rail”.

Po wyborze wzmacniacza operacyjnego rozważano możliwe sposoby implementacji regulacji poziomu wzmocnienia. Ze względu na mobilny charakter projektowanego wzmacniacza zdecydowano się na analizę rozwiązań, które mogą być sterowane w sposób cyfrowy za pomocą interfejsu I²C.

W tym celu porównano trzy rozwiązania (tabela 3.8.):

- 1) Regulacja wartości wzmocnienia za pomocą potencjometru cyfrowego.
- 2) Regulacja wartości wzmocnienia za pomocą przetwornika DAC (PGA).
- 3) Regulacja wartości za pomocą multiplekserów/demultiplekserów analogowych.

Tab. 3.8. Porównanie cyfrowych sposobów regulacji wzmocnienia

Kryteria porównania	Potencjometr cyfrowy	Przetwornik DAC	Demultiplekser/multiplekser analogowy
Rozdzielcość	Różna, zależna od układu, maksymalnie do 1024 indeksów	Różna, zależna od układu, zazwyczaj znajduje się w przedziale od 2^8 - 2^{18}	Demultiplekser/multiplekser analogowy nie ma określonej rozdzielcości w sensie tradycyjnym, jak ma to miejsce w przypadku potencjometrów cyfrowych czy przetworników DAC. Podsiadają kanały, pomiędzy którymi przełączają się zmieniając wartość sygnałów analogowych
Wymogi dotyczące implementacji	Wymaga podciągnięcia ścieżek interfejsów komunikacyjnych do stanu wysokiego ¹	Wymaga zastosowania buforu w postaci wzmacniacza operacyjnego ² oraz podciągnięcia ścieżek interfejsów komunikacyjnych do zasilania ¹	Wymaga podciągnięcia ścieżek interfejsów komunikacyjnych do stanu wysokiego
Koszt	Średni, nie przekracza kilku złotych	Średni, nie przekracza kilku złotych	Duży,
Interfejsy sterujące	3-wire, 2-wire	3-wire, 2-wire	3-wire, 2-wire
Rozmiar	Zależny od obudów układu	Zależny od obudowy układu	Zależny od specyfikacji układu
Koszt	Niski	Wyższy, niż potencjometr cyfrowy	Zależny od specyfikacji układu, podobny do potencjometrów cyfrowych

¹Jeśli mikrokontroler nie posiada już podciągniętych ścieżek szeregowych interfejsów komunikacyjnych.

² W przypadku, gdy układ nie posiada wbudowanego wzmacniacz wewnętrznego.

Analizując dane przedstawione w tabeli 3.8. zdecydowano, iż implementacja regulacji poziomu wzmocnienie zostanie zrealizowana za pomocą potencjometru cyfrowego. Układy potencjometrów cyfrowych cechują się średnio niższą ceną od układów DAC oraz niskim poborem prądu. Ponadto, w przypadku wyboru potencjometru cyfrowego obsługa jest niezbyt skomplikowana. Za ich wykorzystaniem przemawia również doświadczenie w pracy z tego typu układami.

W tabeli 3.9. zestawiono, cyfrowe potencjometry, które spełniają narzucone im następujące wymagania:

- minimalny napięcie wymagane do zasilenia układu ≤ 3.3 V,
- niewielki rozmiar (rozmiary nie przekraczające obudowy SC-70-6),
- cena nie przekraczająca 10 zł,
- rezystancja maksymalna ≥ 50 k Ω ,

- liczba indeksów ≥ 30 ,
- montaż SMD.

Tab. 3.9. Tabela porównawcza potencjometrów cyfrowych

Kryteria porównania	MCP40D18T-104E/LT	AD5246BKSZ50-RL7	CAT5110TBI-00GT3
Dostępność w ofercie międzynarodowych dostawców elektroniki	Tak	Tak	Tak
Producent	Microchip	Analog Devices	Catalyst Semiconductor Inc.
Koszt	3.24 zł	8 zł	3 zł
Rozdzielcość	128 indeksów	128 indeksów	32 indeksów
Tolerancja	20 %	20 %	20 %
Prąd roboczy zasilania	1.8 V - 5.5V	2.7V - 5.5V	2.7V - 5.5V
Rezystancja minimalna	5 kΩ	0 kΩ	0 kΩ
Rezystancja maksymalna	100 kΩ	50kΩ	100 kΩ
Obudowa	SC-70-6	SC-70-6	SOT-23-6

Dane pochodzą z not katalogowych układów oraz od dystrybutorów elektroniki Mouser i DigiKey.

Podsumowując zgromadzone dane w tabeli 3.9. wybrano układ MCP40D18T-104E/LT. Komponent ten wyróżnia się niskim kosztem zakupu oraz posiada obudowę SC.-70-6 o niewielkich rozmiarach. Potencjometr posiada jedynie jeden rejestr i obsługuje dwie funkcje Read/Write, opisane w nocy katalogowej, co potencjalnie ułatwia proces tworzenia oprogramowania służącego do komunikacji za pomocą interfejsu szeregowego I²C. Ponadto układ z serii MCP40D18T wraz z przystawką umożliwiającą umieszczenie go na płytce stykowej dostępny był w Zakładzie Inżynierii Biomedycznej, dzięki czemu możliwe było zapoznanie się z jego działaniem przed realizacją fizyczną układu.

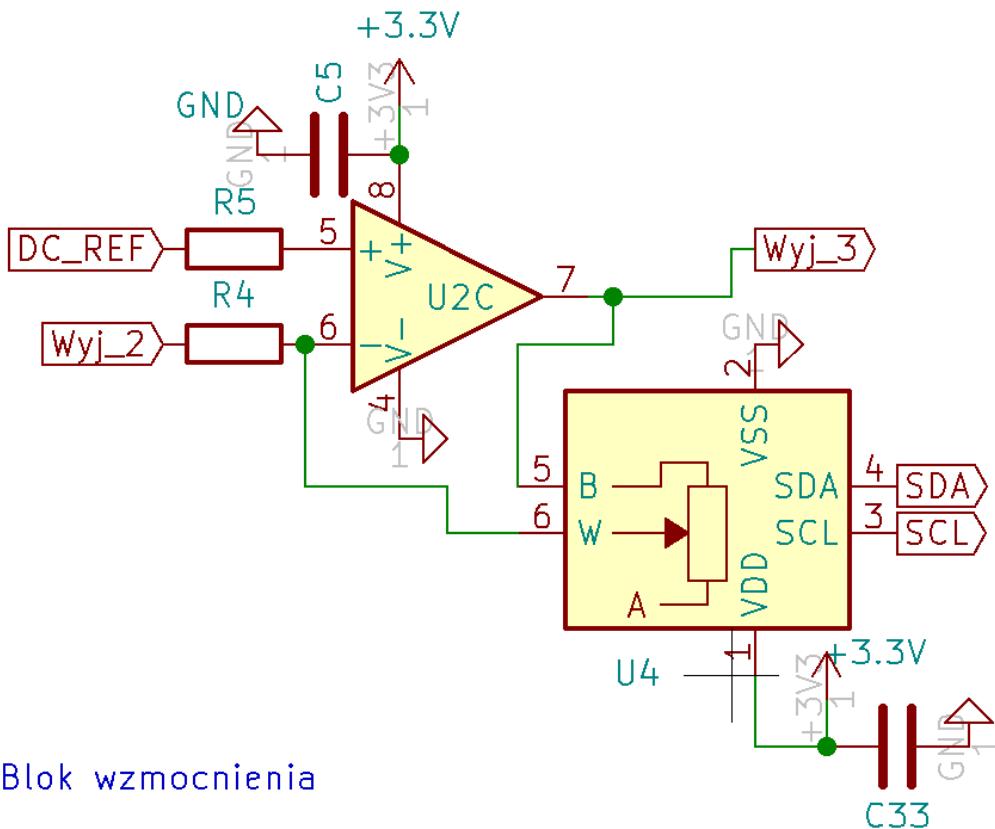
3.4.1. Schemat bloku II stopnia wzmacnienia

Wzmocnienie w tym bloku regulowane jest za pomocą potencjometru MCP40D18T-104E/LT. W tabeli 3.10. przedstawiono parametry urządzenia pochodzące z noty katalogowej [19]. Przyjęto wartość rezystora R4 jako 250 Ω [E96 tol. 1%].

Tab. 3.10. Parametry potencjometru MCP40D18T-104E/LT

Wartość rezystancji w pozycji 128	100 kΩ ($\pm 20\%$)		
Tolerancja	20 %		
Obudowa	SC-70-6		
Wartość rezystancji w pozycji 0	325 Ω (max)		
Wartość zmiany rezystancji w potencjometrze	Min. 629,921	Typowa 787.402	Max. 944.882

Na rysunku 3.4. przedstawiono schemat bloku II stopnia wzmacnienia, natomiast w tabeli 3.11. oznaczenia tego bloku.



Rys. 3.4. Schemat zaprojektowanego bloku II stopnia wzmocnienia

Tab. 3.11. Spis elementów wchodzących w skład bloku II stopnia wzmocnienia

Oznaczenie w układzie	Numer seryjny	Wartość
C5	-----	0.1 μ F [E6]
C33	-----	0.1 μ f [E6]
R4	-----	250 Ω [E96]
R5	-----	0 Ω [E96]
U2	A8538ARZ	-----
U4	MCP40D18-104E/LT	-----

Zgodnie z założeniami II stopień powinien oferować wzmocnienie w zakresie do 26-300 V/V z dokładnością 1%. Dokonano obliczeń w celu sprawdzenia czy przy 20% tolerancji potencjometru cyfrowego można uzyskać 1% dokładność wzmocnienia poprzez dobranie w procesie kalibracji offsetu w pozycji suwaka. Rozważono trzy przypadki, gdy rezystancja maksymalna potencjometru wynosi 80 k Ω (wartość minimalna), 100 k Ω (wartość nominalna) oraz 120 k Ω (wartość maksymalna). W przypadku rezystancji minimalnej wartość najbliższą 26 V/V otrzyma się ustawiając potencjometr na indeksie 10 daje to wzmocnienie rzędu 26.5 V/V wartość ta nie mieści się w założonym poziomie tolerancji wzmocnienia. Wartość najbliższą 300 V/V otrzyma się ustawiając suwak potencjometru na indeksie 119 daje to wzmocnienie rzędu 301.14 V/V, wartość ta mieści się w założonym poziomie tolerancji wzmocnienia. W przypadku rezystancji typowej wartość najbliższą 26V/V otrzyma się ustawiając potencjometr na indeksie 8 daje to wzmocnienie rzędu 26.5 V/V wartość ta nie mieści się w założonym poziomie tolerancji wzmocnienia. Wartość najbliższą 300 V/V otrzyma się ustawiając suwak potencjometru na indeksie 95 daje to wzmocnienie rzędu 300.51 V/V,

wartość ta mieści się w założonym poziomie tolerancji wzmacnienia. W przypadku rezystancji maksymalnej wartość najbliższą 26V/V otrzyma się ustawiając potencjometr na indeksie 7 daje to wzmacnienie rzędu 23.97 V/V wartość ta mieści się w założonym poziomie tolerancji wzmacnienia. Wartość najbliższą 300 V/V otrzyma się ustawiając suwak potencjometru na indeksie 79 daje to wzmacnienie rzędu 299.882 V/V, wartość ta mieści się w założonym poziomie tolerancji wzmacnienia. Wartości zostały obliczone za pomocą następującego wzoru:

$$G = \left(\frac{\frac{R_{AB} * N}{127} + R_W}{R_4} \right) \quad (3.3)$$

gdzie :

R_{AB} – Rezystancja potencjometru,

R_W – Rezystancja suwaka (przyjęto maksymalną wartość),

R_4 – Rezystancja rezystora R4,

N – Index suwaka.

W przypadkach, których nie uzyskano wartości tolerancji, zgodnie z przyjętą, należałyby zmienić pozycję zgodnie ze wzorem 3.3 po przeprowadzeniu pomiaru rezystancji potencjometru.

3.5. Blok mikrokontrolera

W tabeli 3.12. zestawiono mikrokontrolery, które spełniają narzucone im następujące kryteria:

- posiadający moduł Bluetooth,
- posiada interfejs I²C,
- cena nie przekraczająca 30 zł,
- rozmiary obudowy nie przekraczający rozmiarów footprintu QFN-48 (lub podobnych),
- typ montażu SMD,
- rozdzielcość przetwornika ADC ≥ 10 bit.

Tab. 3.12. Zestawienie wybranych parametrów mikrokontrolerów rozważanych do zastosowania w projektowanym urządzeniu

Kryteria porównania	BLUENRG-248N	BLUENRG-238N	NRF52832-QFAB-R	EFR32BG1B232F256 GM32-C0R
Dostępność w ofercie międzynarodowych dostawców elektroniki	Tak	Tak	Tak	Tak
Koszt	20,2 zł	19,06 zł	22,65 zł	28,60 zł
Producent	STMicroelectronics	STMicroelectronics	Nordic Semiconductor ASA	Silicon Labs
Napięcie robocze zasilania	1.7 V - 3.6 V	1.7 V - 3.6 V	1.7 V - 3.6 V	1.71 V - 3.8 V
Obudowa	QFN-48	QFN-32	QFN-48	QFN-32
Rozdzielcość przetwornika ADC	10 bit	10 bit	10 bit	10 bit
Dostępność zestawu ewaluacyjnego	Możliwy do wypożyczenia z Zakładu Inżynierii Biomedycznej	Możliwy do wypożyczenia z Zakładu Inżynierii Biomedycznej	Konieczność zakupu, koszt 170 zł	Konieczność zakupu, koszt 143 zł
Protokół komunikacji	Bluetooth 5.0	Bluetooth 5.0	Bluetooth 5.3	Bluetooth 4.0
GPIO	32	15	16	32
Posiadane interfejsy	I ² C, SPI, UART	I ² C, SPI, UART	I ² S, UAR, SPI	I ² C, I ² S, SPI, UART
Moc transmisji	8 dBm	8 dBm	4 dBm	10.5 dBm
Konieczność zakupu układu	Układ dostępny do wykorzystania w ramach pracy inżynierskiej	Tak	Tak	Tak

Dane zamieszczone w tabeli pochodzą z not katalogowych układów oraz od dystrybutorów elektroniki Mouser i DigiKey.

Na podstawie danych zgromadzonych danych w tabeli 3.12. zdecydowano się na wybór mikrokontrolera BlueNRG-248N. Układ ten dostępny jest w zakładzie, w którym realizowana jest praca dyplomowa stąd ogólny koszt realizacji projektu obniża się. Dostępny do wypożyczenia zestaw ewaluacyjny pozwala na zaznajomienie się z funkcjonowaniem układu oraz przygotowanie oprogramowania przed fizyczną realizacją układu, co znaczco przyśpiesza oraz ułatwia proces projektowania. Nie byłoby to możliwe w przypadku pozostałych układów, gdyż koszt układów deweloperskich znacznie przewyższa przeznaczony na wykonanie prototypu budżet. Ponadto znajomość tworzenia oprogramowania na mikrokontrolery producenta STMicroelectronics oraz duża baza dedykowanych materiałów i publikacji ułatwia tworzenie oprogramowania. Mikrokontroler BlueNRG-248N cechuje się wysoką mocą transmisji oraz czułością wynoszącą – 88 dBm, co umożliwia poprawne działanie protokołu komunikacji Bluetooth na większych odległościach.

Następnie na podstawie noty katalogowej wybranego układu, biorąc pod uwagę dwa podejścia do realizacji podukładu odpowiadającego za siłę transmisji dobrano (RF Matching Network) (tabela 3.13.).

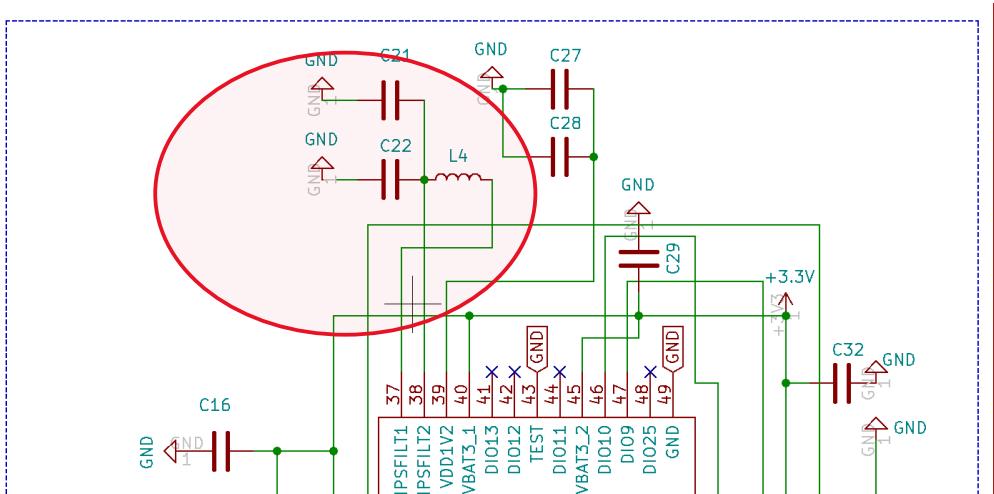
Tab. 3.13. Porównanie rozwiązań dotyczących podukładu RF-Matching Network

Kryteria porównania	Zastosowanie BALF-NRG	Wykonanie układu za pomocą komponentów pasywnych
Koszt	Nie przekracza kilku złotych	Nie przekracza kilku złotych
Powierzchnia zajmowana przez rozwiązanie	Powierzchnia jest sumą rozmiaru obudowy układu BALF-NRG wynoszącej $1.385 \text{ mm} \pm (0.05\text{um}) \times 0.855 \text{ mm} \pm (0.05\text{um})$ oraz obudów dobranych komponentów. Zgodnie z prezentacją układu BlueNRG przykładowa realizacja wymaga 3 komponentów pasywnych oraz BALF-NRG.	Powierzchnia zależna jest od obudów wybranych komponentów. Zgodnie z prezentacją układu BlueNRG przykładowa realizacja wymaga 9 komponentów pasywnych.
Charakterystyka rozwiązania	Gotowe, wielokrotnie testowane rozwiązanie przygotowane przez producenta	Rozwiązanie pozwalające dostosować parametry w zależności od potrzeb
Potencjalne trudności w realizacji	Układ posiada niewielkich rozmiarów obudowę BGA, która może stworzyć znaczną trudność w procesie montażu	Mogą wystąpić trudności w doborze wartości elementów odpowiadających za moc transmisji
Moc transmisji	Wysoka	Możliwe problemy z osiągnięciem mocy transmisji na poziomie układu BALF-NRG

Podsumowując dane przedstawione w tabeli 3.13. do realizacji prototypu wybrano rozwiązanie zakładające wykonanie podukładu RF-Matching Network za pomocą komponentów pasywnych. Przyjęte rozwiązanie daje możliwość wielokrotnej rekonfiguracji wartości elementów podukładu w celu dobrania odpowiedniej mocy transmisji. Jednocześnie w przypadku napotkania trudności w doborze wartości elementów firma STMicroelectronics oferuje gotowe schematy aplikacyjne oraz materiały ułatwiające ten proces. Względem przyjętych założeń projektowych wybór jest również odpowiedni ze względu na niski koszt realizacji oraz potencjalną małą powierzchnię zajmowaną przez rozwiązanie.

3.5.1. Projekt bloku mikrokontrolera

Wartości oraz rodzaj poszczególnych elementów dobrano zgodnie danymi zamieszczonymi przez producenta w nacie katalogowej układu BlueNRG-2 [20]. Rozmieszczenie komponentów zostało natomiast oparte o notę aplikacyjną układu BlueNRG-2 (rysunek 3.5.).



Rys. 3.5. Fragment schematu zaprojektowanego bloku mikrokontrolera BlueNRG-2. Czerwonym kolorem zaznaczono układ DC-DC

Dodatkowo zdecydowano się na aktywy układ DC-DC konwertera oznaczonego na rysunku 3.5. czerwonym kolorem. Konfiguracja ta pozwala na 13.7 mA oszczędności w poborze prądu przez mikrokontroler przy pracy z maksymalną mocą transmisji + 8 dB.

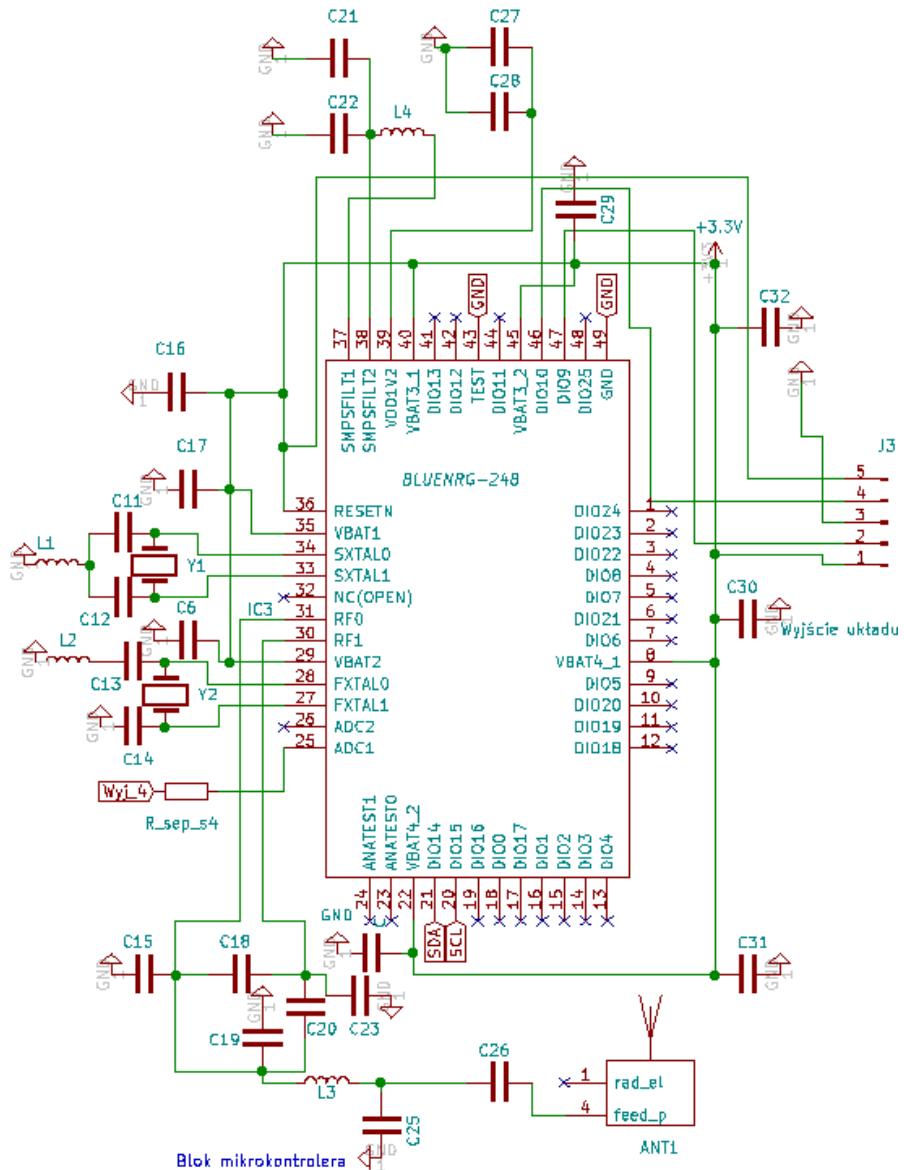
W przypadku podukładu RF-Matching Network wartości komponentów pasywnych zostały dobrane na podstawie prezentacji układu BlueNRG-2 firmy STMicroelectronics [21].

W tabeli 3.14. przedstawiono wartość poszczególnych elementów wchodzących w skład bloku mikrokontrolera.

Tab. 3.14. Spis komponentów wchodzących w skład bloku mikrokontrolera

Oznaczenie w układzie	Numer seryjny	Wartość
ANT1	ANT016008LCS2442MA1	-----
C11, C12	-----	22 pF
C13, C14	-----	15 pF
C15, C26	-----	51 pF
C20	-----	0.8 pF
C21	-----	1 µF
C23	-----	0.7 pF
C25	-----	0.8 pF
C27	-----	150 nF
C6, C16, C17, C22,C28 C30, C31, C32	-----	100 nF
L1	LQG18HN2N2S00D	2.1 nH
L2	LQW18AN3N9C10D	3.9 nH
L3	LQW18AN3N9C10D	3.7 nH
L4	LQW18AN3N9C10D	10 uH
IC3	BlueNRG-248	-----
Y1	ASAK-32.768KHZ-LRS-T	32.768 kHz
Y2	CX2016DB32000D0WZRC1	32 MHz

Na rysunku 3.6. przedstawiono schemat bloku mikrokontrolera wraz z oznaczeniami zgodnymi z tabelą 3.14.



Rys. 3.6. Schemat bloku mikrokontrolera

3.6. Blok zasilania

Znając wybrany poziom napięcia zasilania zdecydowano się na wybór liniowego regulatora napięcia zasilania. W tabeli 3.15. przedstawiono porównanie regulatorów liniowych spełniających następujące kryteria:

- napięcie wyjściowe wynoszące 3.3 V (Fixed),
- niewielki rozmiar,
- cena nie przekraczająca 15 zł,
- montaż SMD,
- współczynnik PSRR $\geq 60\text{dB}$
- maksymalny prąd wyjściowy $< 300\text{mA}$
- pobór prądu $< 200 \mu\text{A}$,
- napięcie zwolnienia $< 220 \text{ mA}$.

Tab. 3.15. Porównanie regulatorów linowych do wykorzystania w bloku zasilania

Nazwa	ADP7118ACPZN3.3-R7	LP2986IM-3.3/NOPB
Dostępność	Tak	Tak
Koszt	10.35 zł	11.80 zł
Producent	Analog Devices	Texas Instruments
Maksymalne napięcie wejścia	20 V	16 V
Napięcie na wyjściu	3.3 V	3.3 V
PSRR	88 dB (10kHz)	65 dB
Footprint	6-UDFN	SOIC-8
Pobór prądu	180 µA	100 µA
Prąd wyjścia	200 mA	200 mA
Napięcie zwolnienia	200 mV	180 mV

*wartości typowe

Dane zamieszczone w tabeli pochodzą z not katalogowych układów oraz od dystrybutorów elektroniki Mouser i DigiKey.

Na podstawie analizy danych (tabela 3.15.) wybrano układ ADP7118ACPZN3.3-R7. Układ ten cechuje się małą obudową 6-UDFN, wysokim współczynnikiem PSRR, niskim napięciem zwolnienia oraz maksymalnym napięciem wejściowym <20V, co czyni go kompatybilnym z większością baterijnych źródeł zasilania.

Następnie dokonano wyboru wzmacniacza operacyjnego do wykorzystania w układzie VGND jako bufor. W tym celu w tabeli 3.16. zestawiono wzmacniacze operacyjne spełniające następujące kryteria:

- napięcie zasilania 3.3 V,
- niewielki rozmiar,
- cena nie przekraczająca 15 zł,
- montaż SMD,
- wysoki współczynnik PSRR ≥ 80 dB,
- zdolność do pobierania/dostarczania prądu (z ang. sink/source current) > 20 mA. Wartość ta powinna być wystarczająca dla wszystkich układów wymagających napięcia VGND, tj. AD620ARZ, AD8538ARZ oraz bloków filtracji.
- niskoszumność

Tab. 3.16. Zestawienie wzmacniaczy operacyjnych do użycia w układzie VGND

Nazwa	AD8538ARZ	OPA835IDBVT	AD8671ARZ-REEL
Dostępność	Tak	Tak	Tak
Koszt	14.99 zł	13.35 zł	16.93 zł
Producent	Analog Devices	Texas Instruments	Texas Instruments
Napięcie zasilania	2.5 V - 5.5V	2.5 V - 5.5 V	2.7 V - 12 V
Typowy prąd roboczy zasilania	180 µA	250 µA	600 µA
Rozmiar/Obudowa	SOIC-8	SOIC-8	SOIC-8
Rail to Rail	Tak	Tak	Tak
Prąd wyjściowy	25 mA	40 mA	20 mA
PSRR	125 dB	105 dB	133,98 dB

Dane zamieszczone w tabeli pochodzą z not katalogowych układów oraz od dystrybutorów elektroniki Mouser i DigiKey.

Analizując dane przedstawione w tabeli 3.16. zdecydowano się na wybór układu AD8538ARZ. Układ ten posiada wysoką wartość współczynnika PSRR wynoszącą 125 dB, dzięki czemu znacznie redukuje zakłócenia pochodzące od źródła zasilania. Posiada również odpowiedni prąd wyjściowy wynoszący 25 mA. Dodatkowo niski prąd roboczy zasilania oraz stosunkowo mały rozmiar koreluje z założeniami dotyczącymi pracą układu baterii.

Na wstępie dokonano wyboru źródła baterijnego zasilania układu. W celu oszacowania maksymalnego poboru prądu przez projektowane urządzenie w tabeli 3.17. zestawiono maksymalne pobory prądu komponentów wzmacniacza.

Tab. 3.17. Sumaryczny pobór prądu przez poszczególne elementy

Nazwa	Pobór prądu
Wzmacniacz instrumentalny AD623	480 µA
Wzmacniacz operacyjny AD8538	215 µA
Wzmacniacz operacyjny AD8542	75 µA
Linowy regulator napięcia ADP7118	180 µA
Mikrokontroler BlueNRG2	15.1 mA
Potencjometr MCP40D18	80 µA
Suma dla 6 h	97,8 mAh

Z analizy danych zawartych w tabeli 3.17. wynika, iż sumaryczny pobór prądu wybranych komponentów potrzebny do zapewnienia 6 h czasu pracy układu wynosi 97,8 mAh.

Natomiast realny pobór dla całego układu może znacznie różnić się od sumarycznego poboru poszczególnych elementów, stąd zdecydowano się na przyjęcie 300% poboru sumarycznego jako oszacowany pobór prądu dla projektowanego układu.

Zdecydowano się na wybór akumulatora Li-Po. Posiada on wysoką gęstość energii, dzięki czemu mogą przechowywać więcej energii w stosunku do swojej masy i objętości co koreluje z przenośnym charakter urządzenia. W tym celu w tabeli 3.18. zestawiono źródła zasilania spełniające następujące kryteria:

- minimalne napięcie znamionowe >3.3 V,
- niewielki rozmiar (nie przekraczające wymiarów 8X25x40mm),
- cena nie przekraczająca 20 zł,
- pojemność ≥100 mHA,

Tab. 3.18. Porównanie źródeł zasilania do wykorzystania w bloku zasilania

Nazwa	ACCU-LP752030/CL	ACCU-LP402025/CL	L401230
Dostępność	Tak	Tak	Tak
Koszt	16 zł	20,91 zł	21,16 zł
Producent	CELLEVIA BATTERIES	CELLEVIA BATTERIES	CELLEVIA BATTERIES
Napięcie znamionowe	3.7 V	3.7 V	3.7 V
Pojemność	300 mAh	150 mAh	110 mAh
Rozmiar	7,5x20x30mm	4,5x21x27,5mm	4x12x30 mm

Dane pochodzą z not katalogowych układów oraz strony dystrybutora elektroniki TME.

Z analizy danych zawartych w tabeli 3.18. zdecydowano się na wybór akumulatora Li-Po ACCU-LP752030/C. Akumulator ten cechuje się odpowiednią wartością napięcia znamionowego, niską ceną wynoszącą 16 zł oraz posiada rozmiary zgodne z wymienionymi kryteriami. Ponadto posiada trzykrotnie większą pojemność od tej estymowanej, dzięki czemu

oprócz umożliwienia 6 h czasu pracy układu, pozwoli na wykonanie testów układu przy baterijnym napięciu zasilania. Dodatkowo jest on dostępny do wykorzystania w ramach pracy dyplomowej.

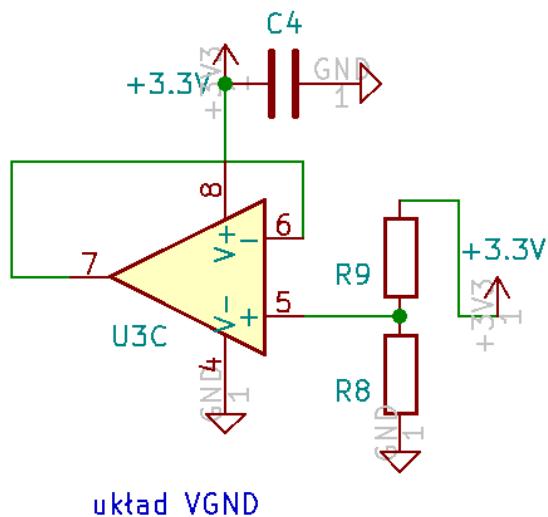
3.6.1. Projekt bloku zasilania

Wartości elementów bloku liniowego regulatora napięcia dobrano zgodnie aplikacją sugerowaną przez producenta zamieszczoną w nocy katalogowej APP7718ACPZN3.3 [22] (LDO). Dodatkowo podciągnięto pin EN bezpośrednio do zasilania. Wykonano również dzielnik napięcia (rysunek 3.7. R9, R8) dzięki czemu na wyjściu ukłdu U3C (rys. 3.19.) otrzymano zaplanowany poziom referencyjny VGND (1,65 V). W tabeli 3.17. przedstawiono elementy wchodzące w skład bloku zasilania.

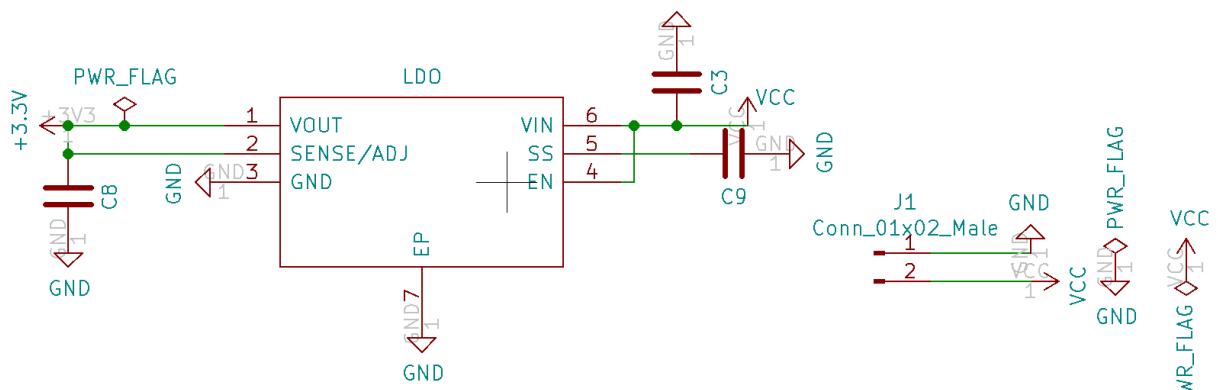
Tab. 3.19. Spis elementów wchodzących w skład bloku zasilania

Oznaczenie w układzie	Wartość
C3	2.2 μF [E6 tol. 20%]
C8	2.2 μF [E6 tol. 20%]]
C9	1 μF [E6 tol. 20%]
C4	0.1 μF [E6]
R8	1 $\text{k}\Omega$ [E96 tol. 1%]
R9	1 $\text{k}\Omega$ [E96 tol. 1%]]
LDO-liniowy regulator napięcia ADP7118ACPZN	-----
U3C Wzmacniacz operacyjny	

Na rysunku 3.7. przedstawiono schemat układ VGND, natomiast na rysunku 3.8. przedstawiono schemat bloku liniowego regulatora napięcia.



Rys. 3.7. Schemat zaprojektowanego bloku VGND



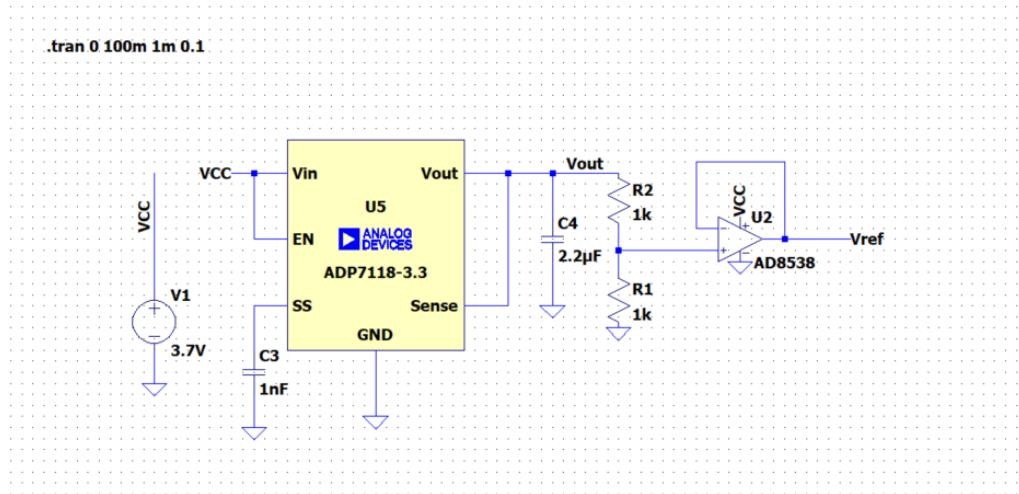
Rys. 3.8. Schemat zaprojektowanego bloku regulatora napięcia zasilania

4. Badanie symulacyjne układu

W tym rozdziale przedstawione zostały wyniki symulacji dla bloku zasilania, przedwzmacniacza oraz całego układu. Wykonana została analiza przejściowa (Transient) i zmiennoprądowa (AC).

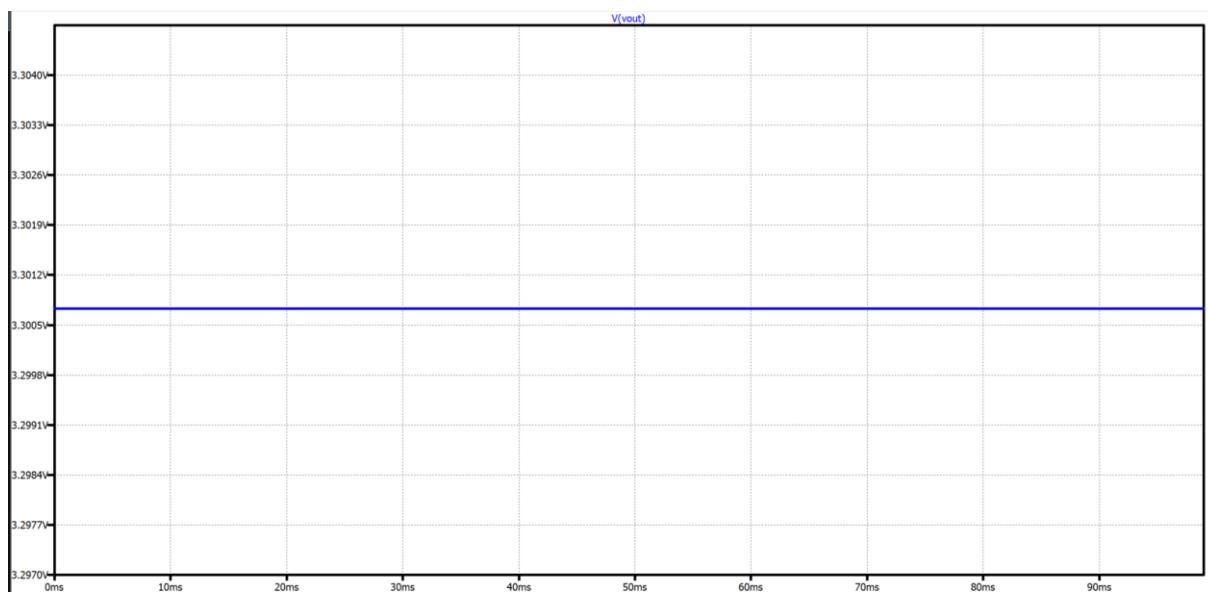
4.1. Symulacja bloku zasilania

Na rysunku 4.1. przedstawiono schemat symulacyjny bloku zasilania. VCC jest uproszczoną reprezentacją akumulatora wybranego jako źródło zasilania.

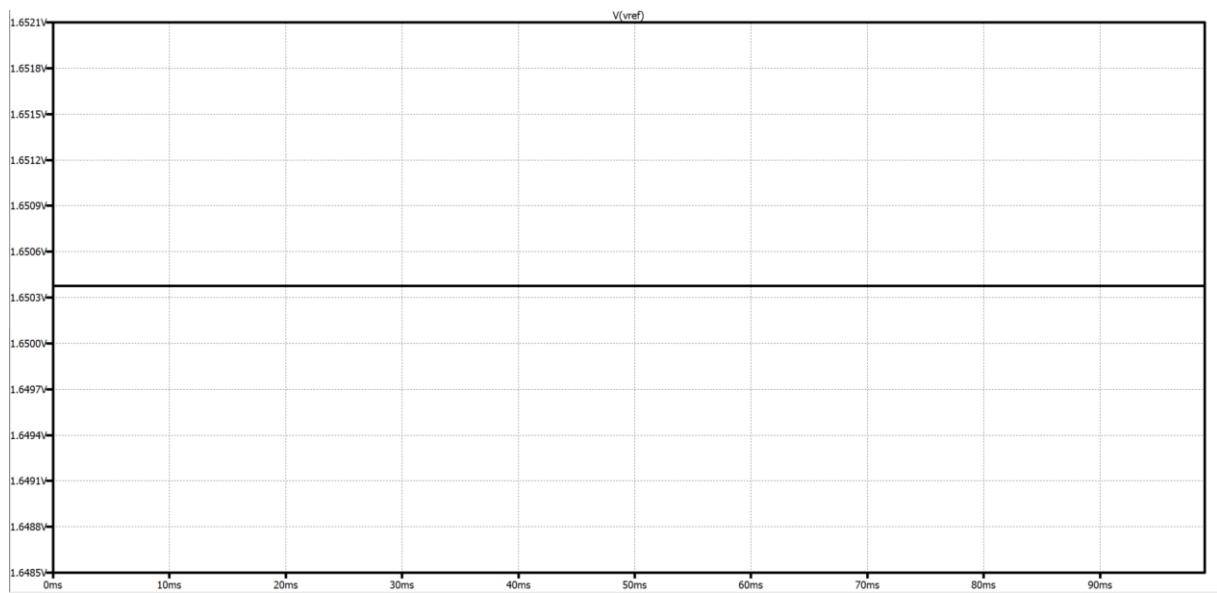


Rys. 4.1. Schemat symulacyjny bloku zasilania wykonany w programie LTspice

Przeprowadzono symulację przejściową w następujących punktach Vout (wyjście LDO) oraz Vref (wyjście z układu VGND). Na rysunku 4.2. przedstawiono wyniki symulacji przejściowej dla punktu Vout, natomiast na rysunku 4.3. przedstawiono wyniki symulacji przejściowej w punkcie Vref.



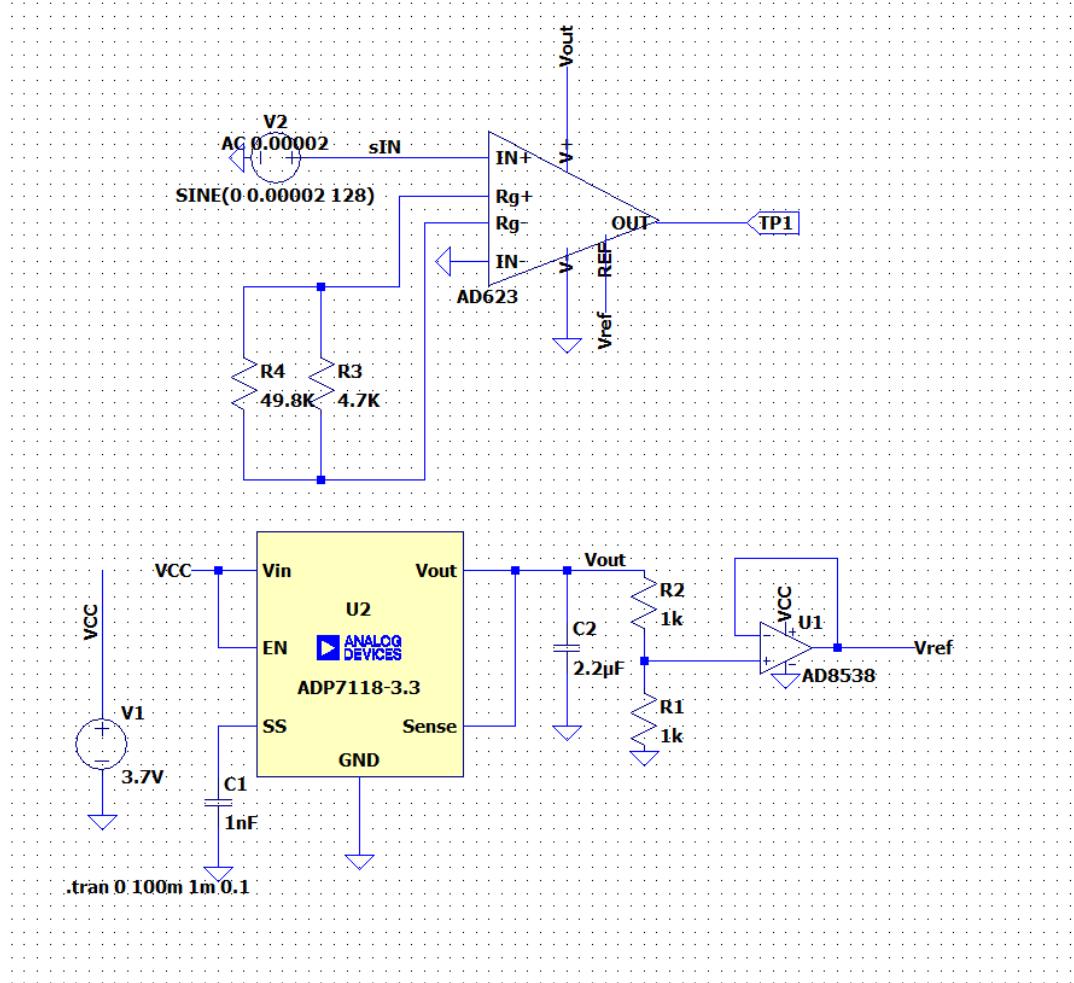
Rys. 4.2. Wyniki symulacji przejściowej w punkcie Vout (wyjście z układu LDO)



Rys. 4.3. Wyniki symulacji przejściowej w punkcie Vref (wyjście z układu VGND)

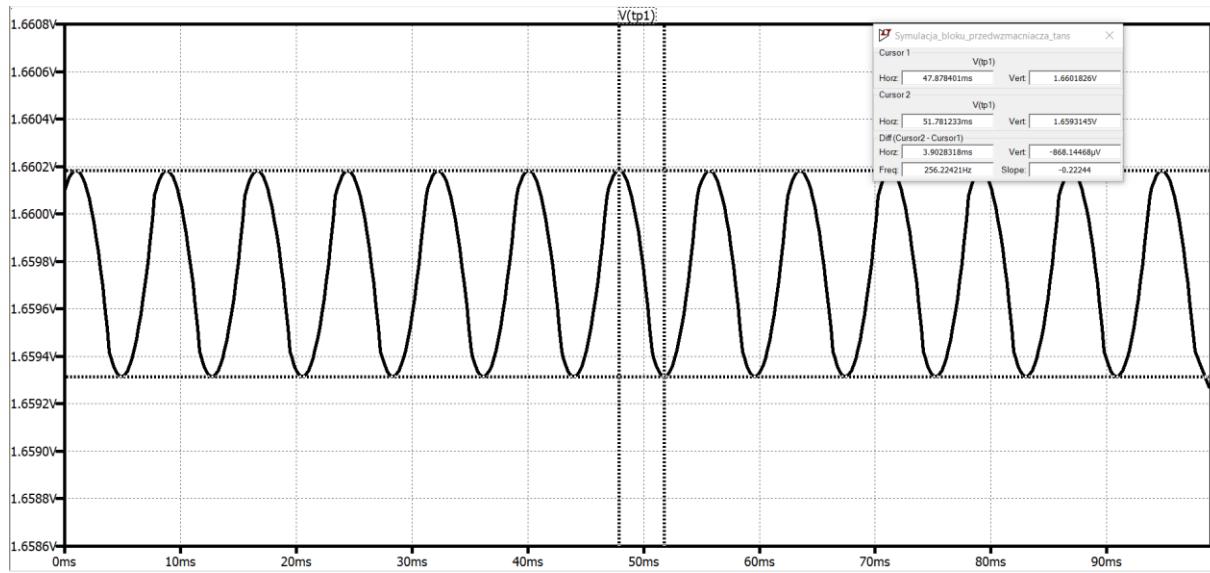
4.2. Symulacja bloku przedwzmacniacza

Na rysunku 4.4. przedstawiono schemat symulacyjny bloku przedwzmacniacza.

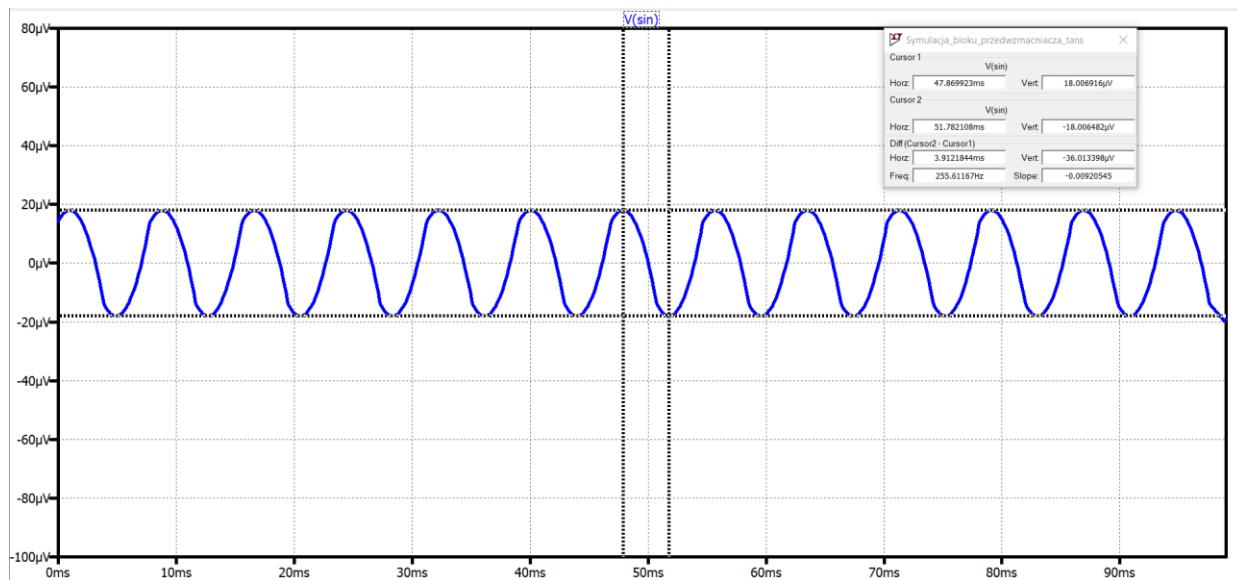


Rys. 4.4. Schemat symulacyjny bloku przedwzmacniacza

Przeprowadzono symulację przejściową na wyjściu wzmacniacza instrumentalnego AD623 - symulacja (TP1). Na rysunku 4.5. przedstawiono wynik symulacji przejściowej na wyjściu wzmacniacza instrumentalnego (TP1), natomiast na rysunku 4.6. przedstawiono sygnał wejściowy (sin).



Rys. 4.5. Wyniki symulacji przejściowej w punkcie TP1 (wyjście z wzmacniacza instrumentalnego). Użyto dwóch kurSORów (linie przerywane) w celu zmierzenia wartości „peak to peak” sygnału. Okno zatytuowane „Symulacje_bloku_przedwzmacniacza_trans” przedstawia różnicę w pozycjach kurSORów, która wyniosła 868.144686 μV.



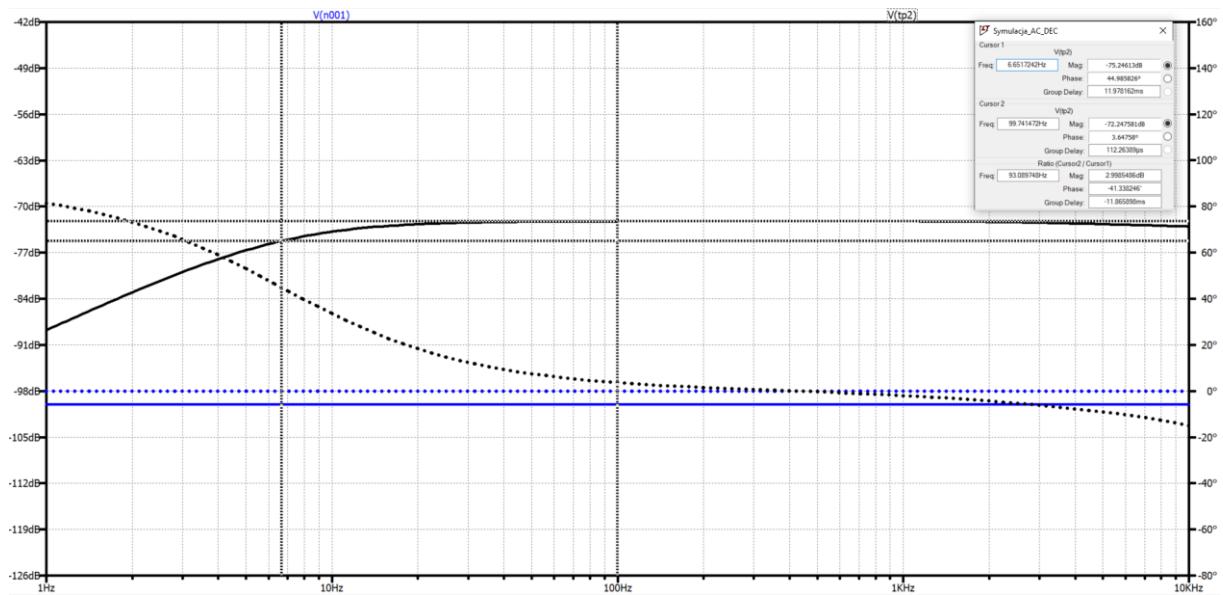
Rys. 4.6. Sygnał wejściowy, używany w symulacji w punkcie sin. Użyto dwóch kurSORów (linie przerywane) w celu zmierzenia wartości „peak to peak” sygnału. Okno zatytuowane „Sumulacje_bloku_przedwzmacniacza_trans” przedstawia różnicę w pozycjach kurSORów, wyniosła ona 36.0133 μV.

Wyznaczone na podstawie symulacji wzmacnienie wyniosło 24.106 V/V. Różnica % względem wartości założonej wynosi 3.576% i nie mieści się w założonej tolerancji.

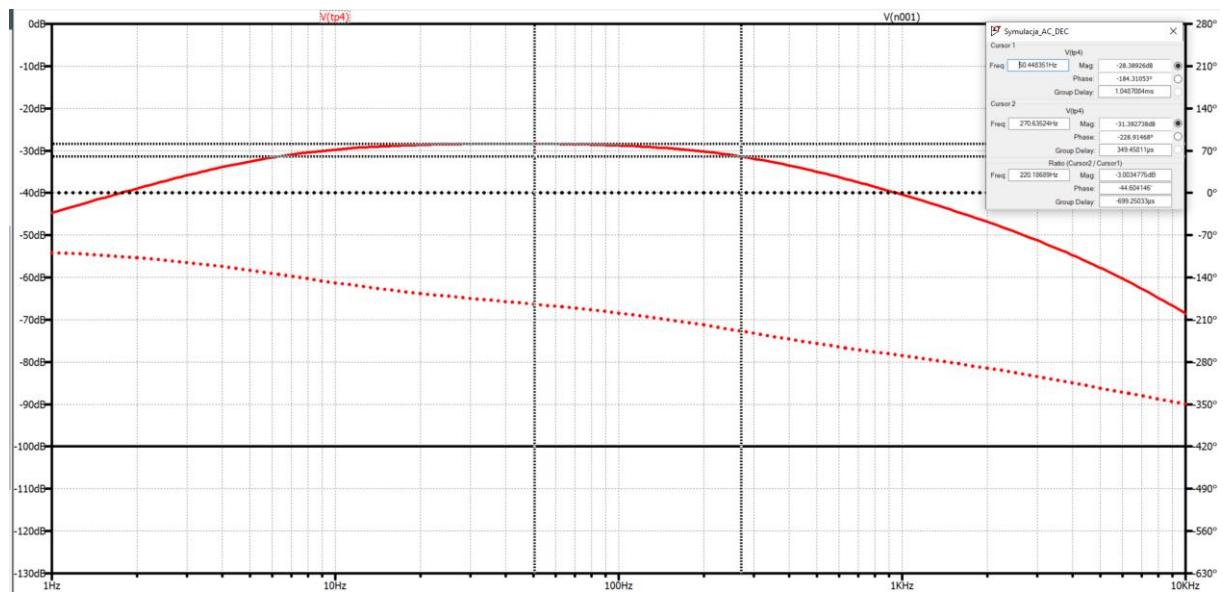
4.3. Charakterystyka amplitudowo-częstotliwościowa

W celu uzyskania charakterystyki amplitudowo-częstotliwościowej układu przeprowadzono symulację AD-DC.

Na rysunku 4.7. przedstawiono wykres charakterystyki amplitudowo-częstotliwościowej na wyjściu filtra górnoprzepustowego (TP2), natomiast na rysunku 4.8. przedstawiono wykres charakterystyki amplitudowo-częstotliwościowej na wyjściu filtra dolnoprzepustowego (TP4).



Rys. 4.7. Charakterystyka amplitudowo-częstotliwościowa na wyjściu filtra górnoprzepustowego (TP2). Użyto dwóch kurSORów (linie przerywane) w celu zmierzenia 3 dB spadku względem środka pasma przepustowego. Okno zatytułowane „Sumulacja_AC_DEC” przedstawia różnicę w pozycjach kurSORów, spadek 3dB przypada na częstotliwość 6.65 Hz.



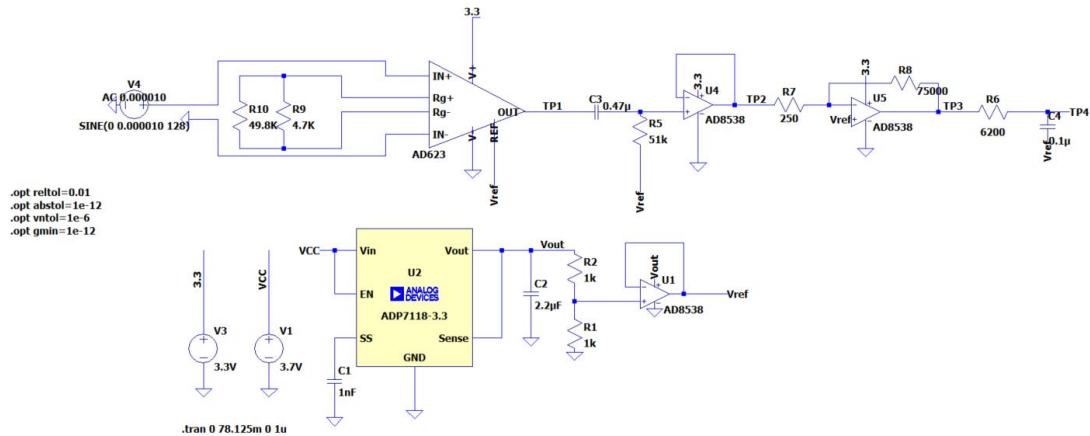
Rys. 4.8. Charakterystyka amplitudowo-częstotliwościowa na wyjściu filtra dolnoprzepustowego (TP4). Użyto dwóch kurSORów (linie przerywane) w celu zmierzenia 3dB spadku względem środka pasma przepustowego. Okno zatytułowane „Sumulacja_AC_DEC” przedstawia różnicę w pozycjach kurSORów, spadek 3dB przypada na częstotliwość 270.63 Hz.

W przypadku filtru górnoprzepustowego 3dB względem maksymalnej amplituda przypada na częstotliwość 6.65 Hz. Różnica % tej wartości względem wartości założonej wynosi 24.9% i nie mieści się w założonym przedziale tolerancji.

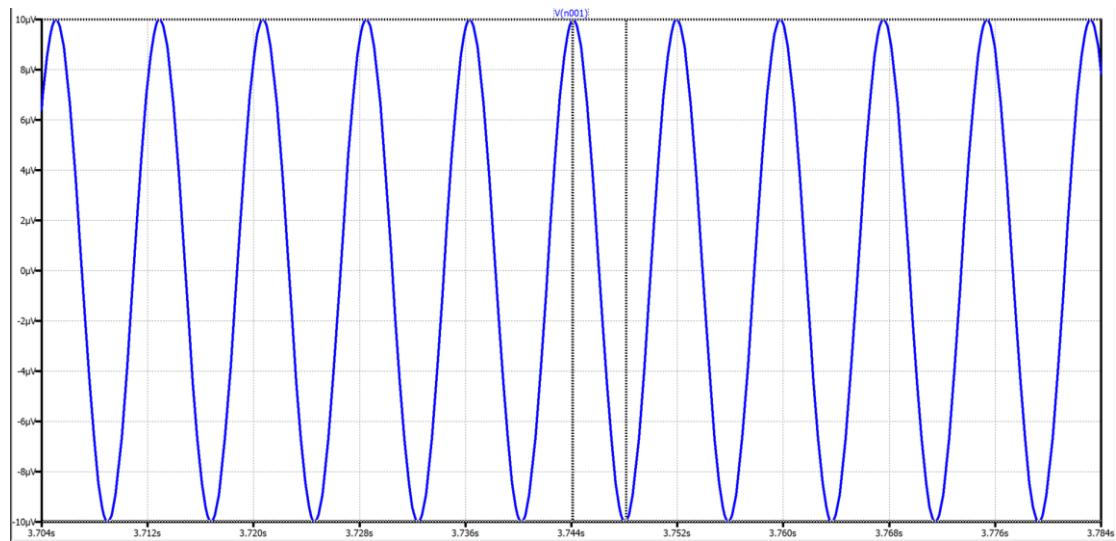
W przypadku filtru dolnoprzepustowego 3dB spadek względem maksymalnej amplituda przypada na częstotliwość 270,63 Hz. Różnica % wartości względem wartości założonej wynosi 5,78% i mieści się w założonym przedziale tolerancji.

4.4. Symulacja wzmacnienia

Przeprowadzono symulację przejściową wzmacnienia na wyjściu filtra dolnoprzepustowego (TP4). Na rysunku 4.9. przedstawiono schemat symulacyjny układu, natomiast na rysunku 4.10. przedstawiono sygnał wejściowy V4.



Rys. 4.9. Schemat symulacyjny toru pomiarowego projektowanego wzmacniacza

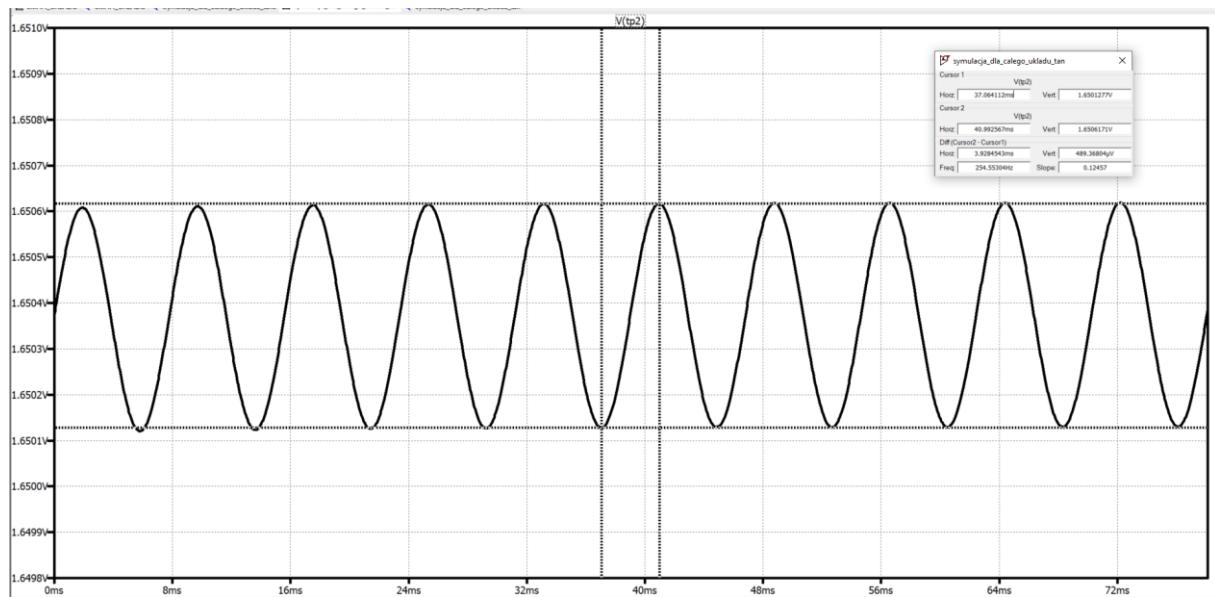


Rys. 4.10. Sygnał wejściowy V2. Jego amplituda wynosi 20 μ V

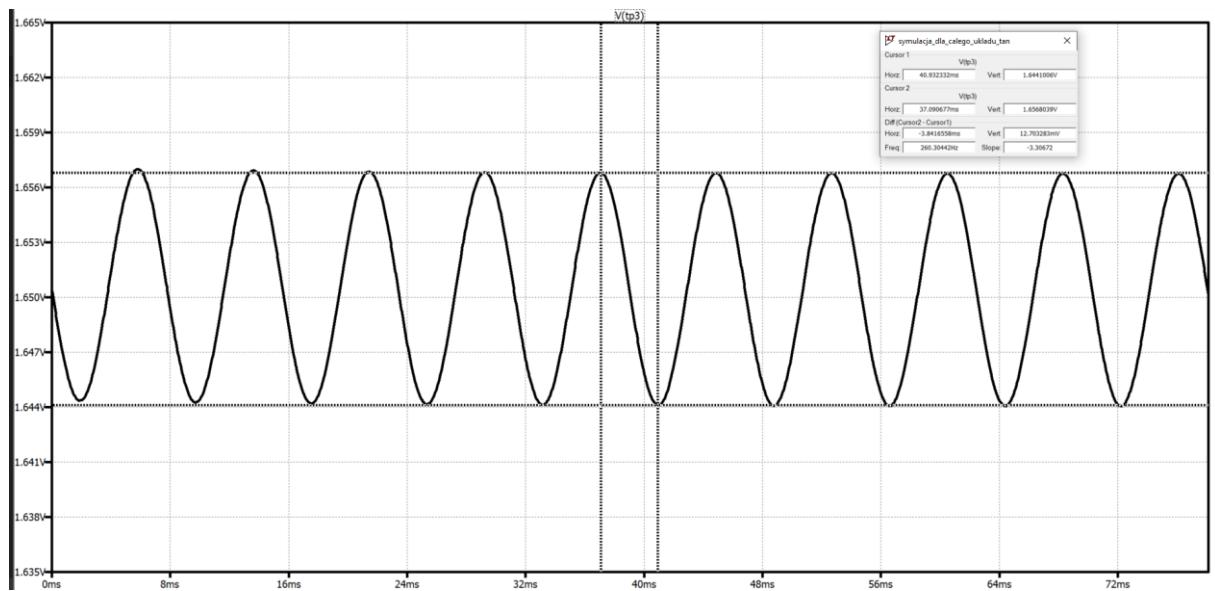
Wartość rezystancji potencjometru znajdującego się w gałęzi sprzężenia zwrotnego (rysunek 4.9.), regulującego wzmacnienie, reprezentowana jest jako rezistor R8. Przyjęto

wartość 6500Ω w celu sprawdzenia symulacji dolnej wartości zakresu wzmacnienia dla II stopnia wzmacnienia

Na rysunku 4.11. przedstawiono sygnał w punkcie TP2 (wyjście z filtra górnoprzepustowego), natomiast na rysunku 4.12. przedstawiono sygnał w punkcie TP3 (wyjście z II stopnia wzmacnienia).



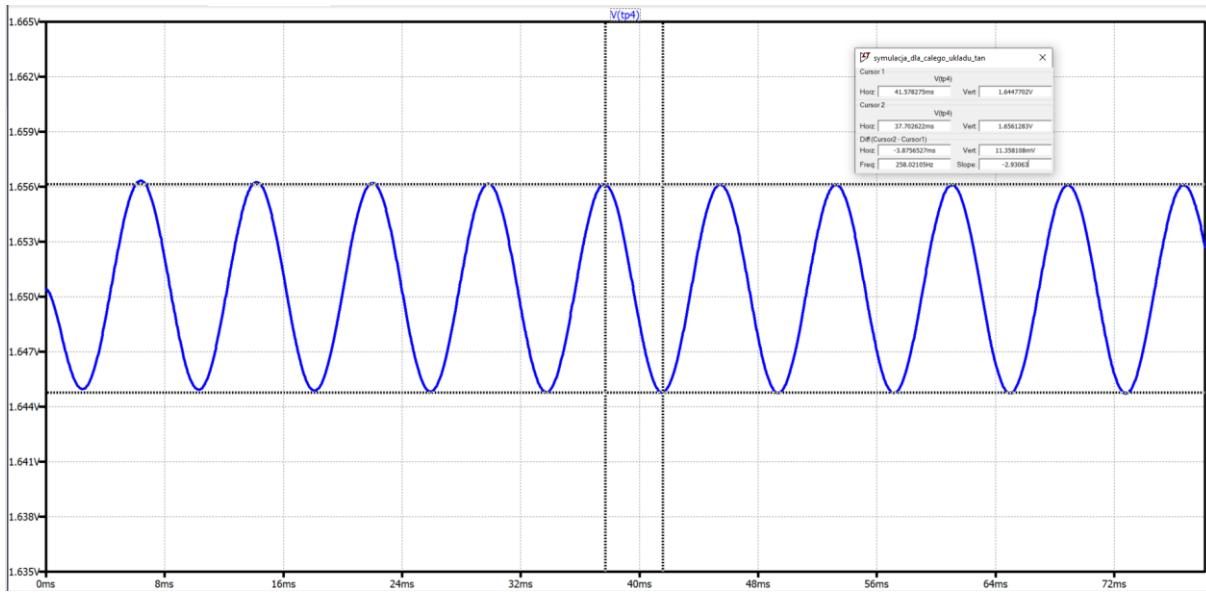
Rys. 4.11. Sygnał w punkcie TP2. Użyto dwóch cursorów (linie przerywane) w celu zmierzenia wartości międzyszczytowej . Okno zatytuowane „Sumulacja_dla_calego_ukladu_tan” przedstawia różnicę w pozycjach cursorów, która wyniosła $489.36 \mu\text{V}$.



Rys. 4.12. Wykres amplitudy sygnału na wyjściu II stopnia wzmacnienia (TP3). Użyto dwóch cursorów (linie przerywane) w celu zmierzenia wartości międzyszczytowej sygnału. Okno zatytuowane „symulacja_dla_calego_bloku_trans” przedstawia różnicę w pozycjach cursorów, która wyniosła 12.70 mV .

Wyznaczone na podstawie symulacji wzmacnienie wyniosło 25.95 V/V . Różnica % względem wartości założonej wynosi mieści się w założonej tolerancji.

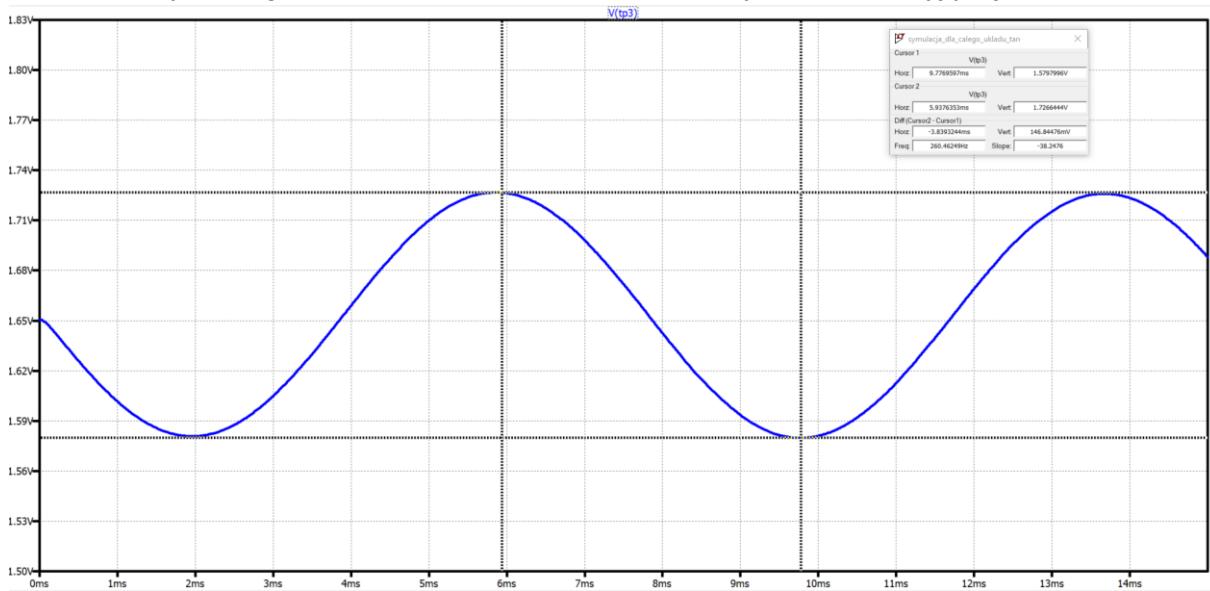
Na rysunku 4.13 przedstawiono wyniki wzmacnienie w punkcie TP4 (wyjście z układu).



Rys. 4.13. Wykres amplitudy sygnału na wyjściu filtra dolnoprzepustowego (TP4). Użyto dwóch kursorów (linie przerywane) w celu zmierzenia wartości międzyszczytowej sygnału. Okno zatytułowane „symulacja_dla_calego_bloku_trans” przedstawia różnicę w pozycjach kursorów, która wyniosła 11.36 mV.

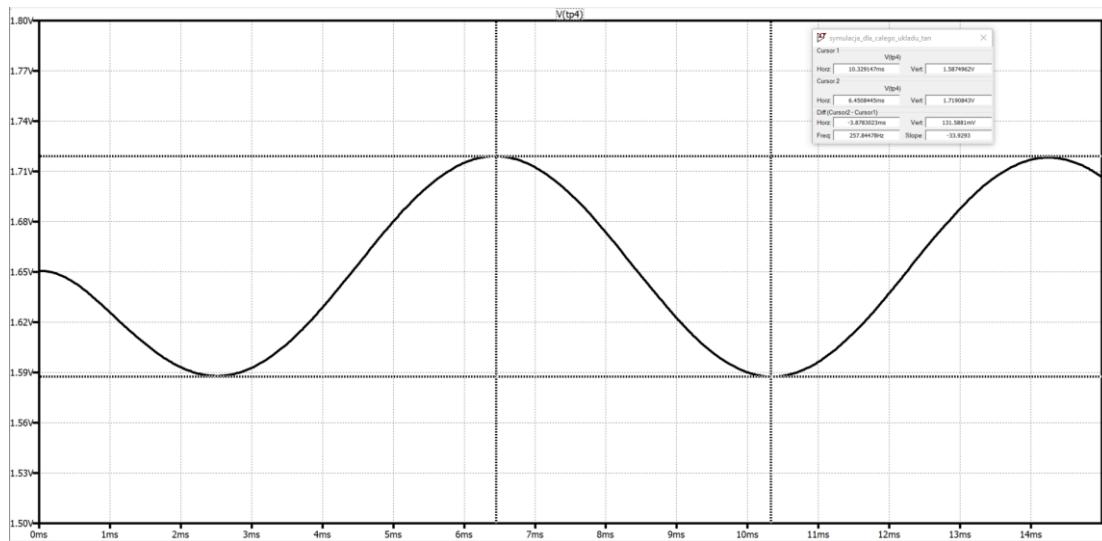
Wyznaczone na podstawie symulacji wzmacnienie wyniosło 568 V/V. Różnica % względem wartości założonej wynosi 9.12 %. Wpływ na różnicę wartości ma tłumienie sygnału przez filtry pasywne. W celu otrzymania wartości wzmacnienia, zgodnej z założoną tolerancją, należy zmienić pozycję suwaka (wzór 3.3).

Dla maksymalnego zakresu wzmacnienia wartość rezystora R8, przyjęto jako 75000 Ohm



Rys. 4.14. Wykres amplitudy sygnału na wyjściu II stopnia wzmacnienia (TP3). Użyto dwóch kursorów (linie przerywane) w celu zmierzenia wartości międzyszczytowej sygnału. Okno zatytułowane „symulacja_dla_calego_bloku_tran” przedstawia różnicę w pozycjach kursorów, która wyniosła 146.84 mV.

Wyznaczone na podstawie symulacji wzmacnienie wyniosło 300.06 V/V. Różnica % względem wartości założonej wynosi mieści się w założonej tolerancji. Na rysunku 4.15 przedstawiono sygnał w punkcie TP4 (wyjście z filtra dolnoprzepustowego).

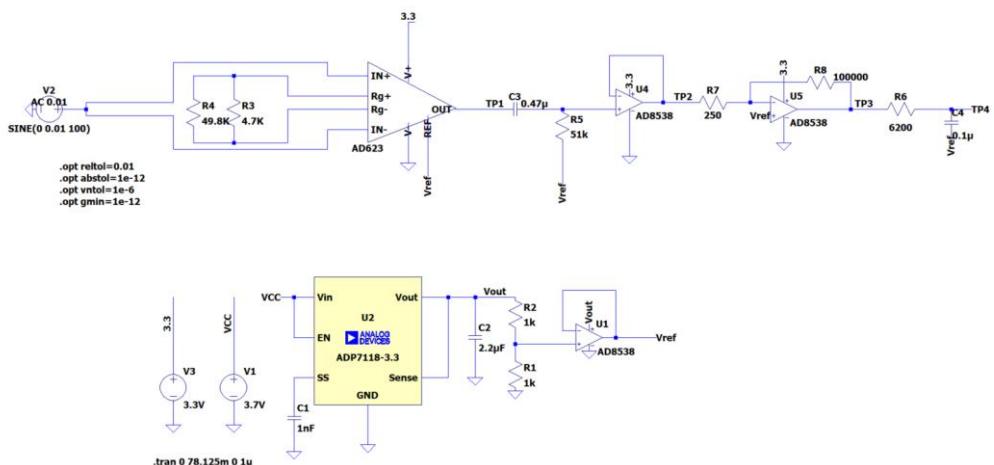


Rys.4.15. Wykres amplitudy sygnału na filtru dolnoprzepustowym (TP4). Użyto dwóch kurSORów (linie przerywane) w celu zmierzenia wartości międzyszczytowej sygnału. Okno zatytułowane „symulacja_dla_calaego_bloku_trans” przedstawia różnicę w pozycjach kurSORów, która wyniosła 131.58 mV.

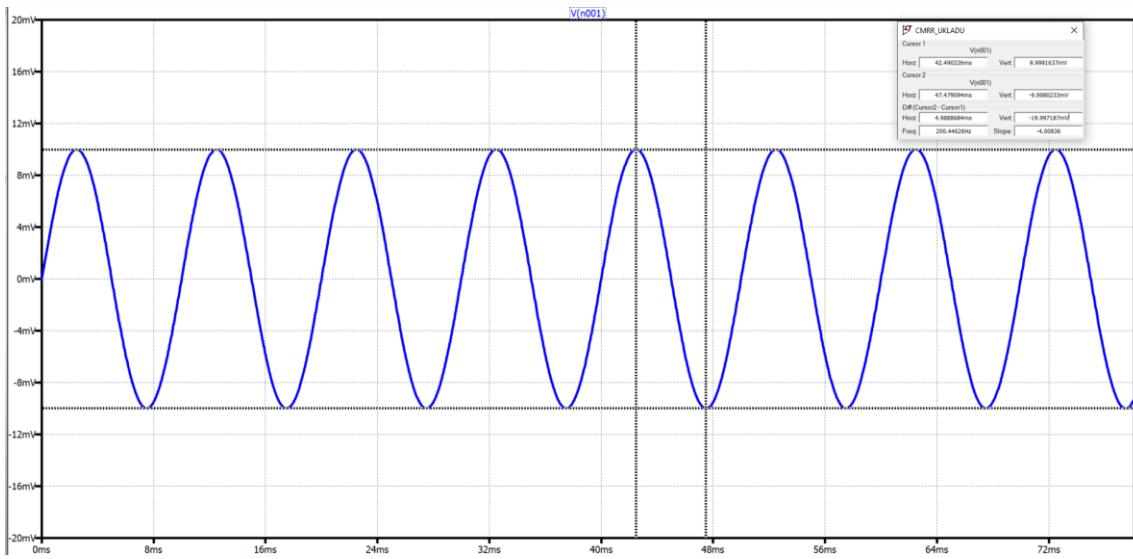
Wyznaczone na podstawie symulacji wzmacnienie wyniosło 6579,15 V/V. Różnica % względem wartości założonej wynosi 13.18% i nie mieści się w założonej tolerancji. Wpływ na różnicę wartości ma tłumienie sygnału przez filtry pasywne. W celu otrzymania wartości wzmacnienia, zgodnej z założoną tolerancją, należy zmienić pozycję suwaka (wzór 3.3).

4.5. Symulacja CMRR

Przeprowadzono symulacje w celu sprawdzenia wartości CMRR dla zaprojektowanego urządzenia. Na rysunku 4.16. przedstawiono schemat symulacyjny, natomiast na rysunku 4.17. przedstawiono sygnał wejściowy V2.

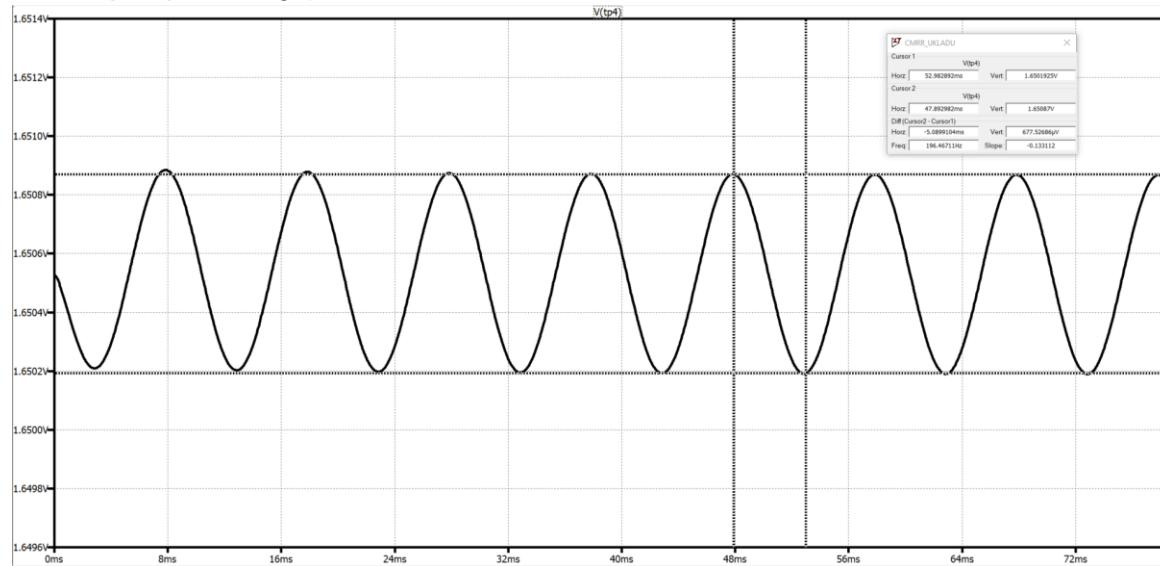


Rys. 4.16. Schemat symulacyjny pomiaru współczynnika CMRR dla toru pomiarowego projektowanego wzmacniacza



Rys. 4.17. Sygnał wejściowy V2 używany w symulacji. Użyto dwóch kursorów (linie przerywane) w celu zmierzenia wartości międzyszczytowej. Okno zatytuowane „CMRR_UKŁADU” przedstawia różnicę w pozycjach kursorów, która wyniosła 20 mV.

Na rysunku 4.18. przedstawiono wynik symulacji przejściowej w punkcie TP4 (wyjście z filtra dolnoprzepustowego)



Rys. 4.18. Wyniki symulacji przejściowej w punkcie TP4. Użyto dwóch kursorów (linie przerywane) w celu zmierzenia wartości międzyszczytowej. Okno zatytuowane „CMRR_UKŁADU” przedstawia różnicę w pozycjach kursorów, która wyniosła 677.52 μV.

Wyznaczone wzmacnienie różnicowe w podrozdziale 4.4. wyniosło 6579,15 V/V.

Wyznaczone na podstawie symulacji wzmacnienie wspólne wyniosło 0,0338 V/V.

Obliczenia zostały wykonane na podstawie poniższego wzoru:

$$CMRR = 20 \log_{10} \left(\frac{Ad}{|Acm|} \right) dB \quad (4.1)$$

gdzie:

Ad – wzmacnienie różnicowe sygnału [V/V],

Acm – wzmacnienie wspólne sygnału [V/V].

CMRR symulacyjne wyniosło 105,76 dB. Wartość ta mieści się z założonym przedziałem.

5. Oprogramowanie

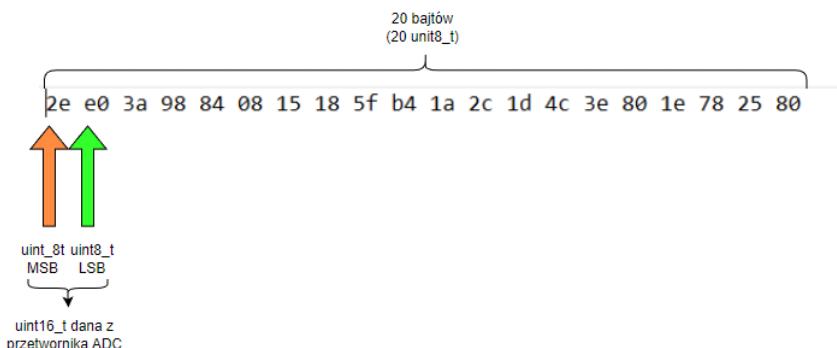
W rozdziale tym przedstawiono oprogramowanie mikrokontrolera BlueNRG-2 oraz aplikację służącą do komunikacji z urządzeniami BLE (aplikację desktopową). Zdefiniowany został również protokół komunikacji oraz zademonstrowano możliwość działania zaprojektowanego układu wzmacniacza w sieci Mesh.

5.1. Opis protokołu komunikacji

Protokół komunikacji pomiędzy urządzeniem nadawczym a odbierającym jest oparty jest o Bluetooth Low Energy (BLE), który opiera się o GATT (Generic Attribute Profile). W projekcie w celu komunikacji została utworzona własna charakterystyka wewnętrz GATT o następujących parametrach:

- adres UUID 128 bitowy 42821a40-e477-11e2-82d0-0002a5d5c51b cd20c480-e48b-11e2-840b-0002a5d5c51b,
- typ notyfikacyjny,
- brak autoryzacji oraz enkrypcji danych.

W ramach notyfikacji do urządzenia odbiorczego wysyłana jest tablica danych o długości 20 elementów typu `uint8_t`. Do tablicy tej zapisywane są dane z przetwornika ADC. Dane te są typu `uint16_t` i wymagają rozdzielania na dwa rodzaje bajtów, tj. starszy (MSB) i młodszy (LSB). Umieszczane są one jako rozdzielne elementy w tablicy w kolejności MSB przed LSB. Na rysunku 5.1. przedstawiono ramkę danych wysyłanych za pomocą charakterystyki.

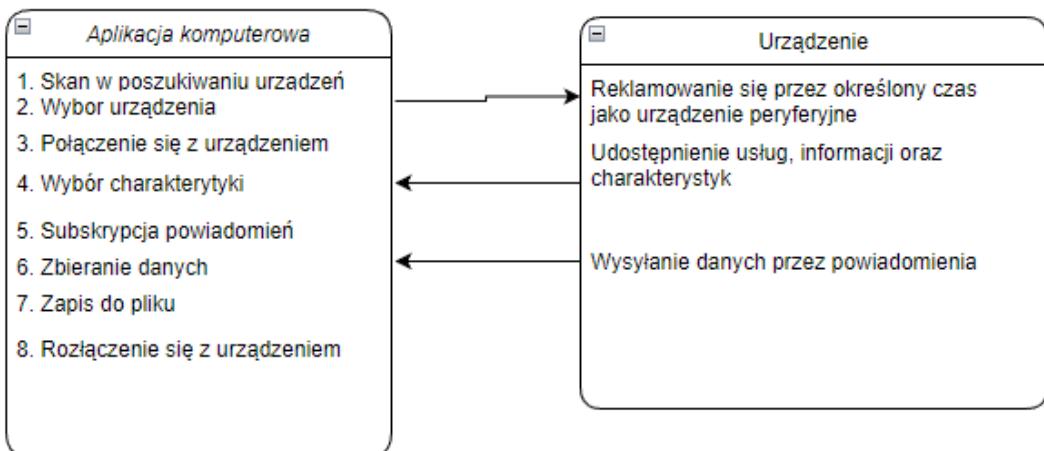


Rys. 5.1. Struktura ramki danych charakterystyki notyfikacyjnej. Ramka na rysunku wypełniona jest przykładowymi danymi

5.1.1. Protokół komunikacji pomiędzy aplikacją a urządzeniem

Protokół komunikacji pomiędzy aplikacją a urządzeniem oparty jest o Bluetooth Low Energy (BLE). Centralne urządzenie łączy się z urządzeniem peryferyjnymi dane wymieniane są za pomocą zdefiniowanych charakterystyk należących do usług.

Na rysunku 5.2. przedstawiono blokowy schemat ilustrujący komunikację między aplikacją a urządzeniem peryferyjnym. Wymiana danych odbywa się poprzez cykliczne notyfikacje.



Rys. 5.2. Diagram protokołu komunikacji między aplikacją a urządzeniem

5.1.2. Aplikacja komputerowa

Opracowane oprogramowanie bazuje na bibliotece SimpleBLE (Kevin Dewald, MIT License). Do jego utworzenia został wykorzystany, dostarczony z biblioteką, szablon aplikacji obsługującej komunikację BLE opartą na notyfikacjach. W opracowanej aplikacji pozbyto się większości funkcjonalności nie związanych bezpośrednio z przetwarzaniem notyfikacji wysłanych przez urządzenie peryferyjne. Miało to na celu zwiększyć wydajność przy odbiorze dużej ilości danych. Dane przechowywane są w buforze, po zakończeniu subskrypcji charakterystyki zapisywane są do pliku, o nazwie wybranej przez użytkownika.

Na rysunku 5.3. przedstawiono okno aplikacji, natomiast na rysunku 5.4. przedstawiono schemat przepływu danych w aplikacji.

C:\Users\48532\Desktop\SimpleBLE-main_v0_2\examples\simpleble\out\build\x64-Debug\bin\example_notify.exe

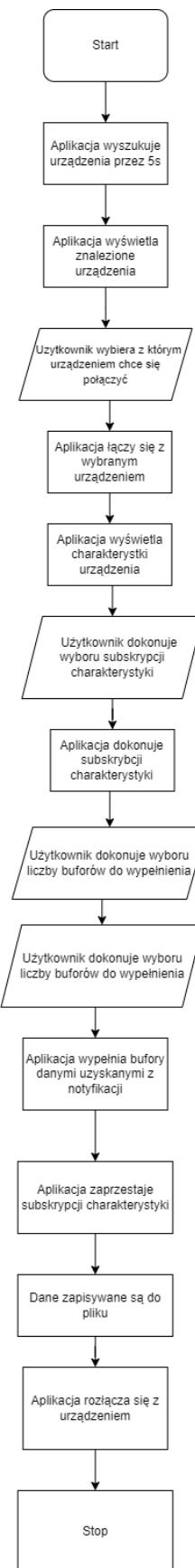
```

[INFO] SimpleBLE: C:\Users\48532\Desktop\SimpleBLE-main_v0_2\simpleble\src\backends\windows\Utils.cpp:33 in initialize_winrt: CoGetApartmentType: cotype=-1, qualifier=0, result=800401F0[INFO] SimpleBLE: C:\Users\48532\Desktop\SimpleBLE-main_v0_2\simpleble\src\backends\windows\Utils.cpp:41 in initialize_winrt: RoInitialize: result=0Using adapter: DESKTOP-KT
Rozpoczęto skanowanie.
Znaleziono urządzenia: [6f:41:a2:dc:fe:75]
Znaleziono urządzenia: [64:cc:6e:e6:aa:11]
Znaleziono urządzenia: BlueNRG [d6:e7:04:fd:6c:29]
Zakonczono skanowanie.Znaleziono następujące urządzenia:

[0] [6f:41:a2:dc:fe:75]
[1] [64:cc:6e:e6:aa:11]
[2] BlueNRG [d6:e7:04:fd:6c:29]
Wybierz urządzenie z którym chcesz nawiązując połączenie (0-2):

```

Rys. 5.3. Widok okna aplikacji



Rys. 5.4. Schemat przepływu danych w aplikacji komputerowej

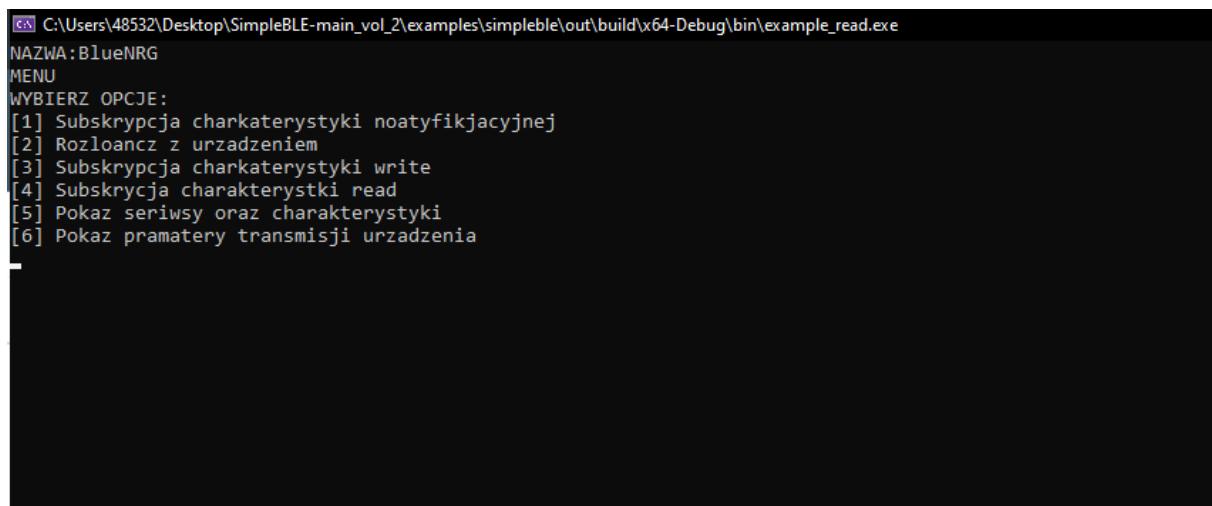
5.1.3. Ostateczna wersja aplikacji

Aplikacja do komunikacji z urządzeniami BLE została napisana w języku C++ z wykorzystaniem środowiska Visual Studios (Microsoft). Bazuje na bibliotece SimpleBLE (Kevin Dewald, MIT License). Dodatkowym celem tej aplikacji było umożliwienie użytkownikowi komunikowanie się z każdym urządzeniem BLE oraz sprawdzenie jego parametrów.

Ta aplikacja różni się od poprzedniej poprzez dodanie następujących funkcjonalności:

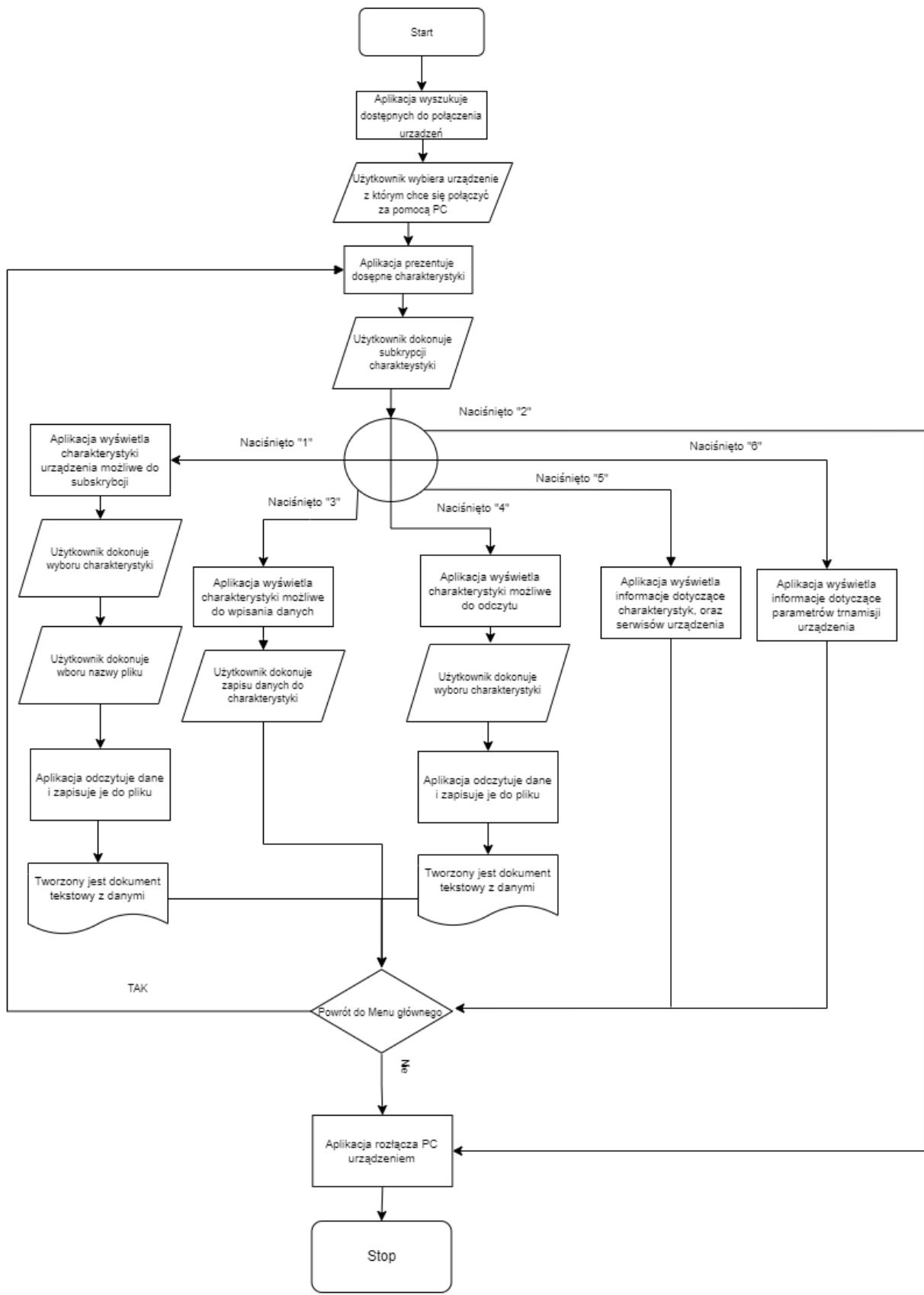
- wyświetla informacje dotyczące charakterystyk takie, jak: długość, tryb, serwis, dostęp,
- wyświetla parametry urządzenia takie, jak: nazwa, wartość RSSI, moc transmisji,
- obsługuje charakterystyki typu read, write.

Na rysunku 5.5. przedstawiono widok z menu aplikacji, natomiast na rysunku 5.6. przedstawiono diagram przepływu danych w aplikacji do komunikacji z urządzeniami BLE o rozszerzonej funkcjonalności.



```
C:\Users\48532\Desktop\SimpleBLE-main_v0_2\examples\simpleble\out\build\x64-Debug\bin\example_read.exe
NAZWA:BlueNRG
MENU
WYBIERZ OPCJE:
[1] Subskrypcja charakterystyki noatyfikacyjnej
[2] Rozloancz z urzadzeniem
[3] Subskrypcja charakterystyki write
[4] Subskrypcja charakterystki read
[5] Pokaz serwisy oraz charakterystyki
[6] Pokaz parametry transmisijs urzadzenia
```

Rys. 5.5. Widok z menu głównego aplikacji o rozszerzonej funkcjonalności



Rys. 5.6. Schemat przepływu aplikacji o rozszerzonej funkcjonalności

5.2. Oprogramowanie mikrokontrolera

Oprogramowanie mikrokontrolera BlueNRG-2 zostało napisane w języku C, z wykorzystaniem środowiska Wise-Studio (STM oraz EWRAM). Natomiast aplikację umożliwiającą komunikację z urządzeniem napisano w języku C++ w oparciu o bibliotekę SimpleBLE, służącą do komunikacji z urządzeniami BLE w środowisku Visual Studio. Do analizy zebranych danych, wykorzystano stworzone w programie MATLAB skrypty.

Tworzenie oprogramowania rozpoczęto od zapoznania się z mikrokontrolerem Blue-NRG2. W tym celu wykorzystano dostępny w Zakładzie Inżynierii Biomedycznej zestaw deweloperski STEVAL-BCN002V1B, w którego skład wchodzi mikrokontroler BlueNRG-2 oraz różnego rodzaju sensory takie, jak:

- czujnik nacisku,
- czujnika ciśnienia,
- akcelerometr,
- żyroskop.

Oprogramowanie mikrokontrolera zaimplementowane w opracowanym urządzeniu zostało oparte o szkielet programu BLE_SensorDemo dostarczanego przez producenta STMicroelectronics, jako przykładowa aplikacja dla zestawów deweloperskich zawierających mikrokontrolery z serii BlueNRG. Program ten umożliwia połączenie się z urządzeniem w ramach protokołu komunikacji BLE, odczyt charakterystyk oraz korzystanie z sensorów dołączonych do układu ewaluacyjnego.

Korzystając z bibliotek dedykowanych do mikrokontrolera BlueNRG-2 skonfigurowano działanie oraz parametry przetwornika ADC (Załącznik A, funkcja *ADC_Configuration()*). Częstotliwość próbkowania ustawiono na 5 kS/s (dostępne były 4 wartości częstotliwości próbkowania 5 kS/s, 10 kS/s, 15.625 kS/s, 31.25 kS/s). Następnie dostosowano działanie modułu DMA (Załącznik A, *Funkcja DMA_Configuration()*), do obsługi danych gromadzonych w buforze przetwornika ADC. Dodanie dwóch flag dla kanału DMA, indykujących połowiczne oraz całkowite zapełnienie bufora danych ADC umożliwia zarządzanie wieloma danymi pochodzącyymi z przetwornika.

Dodatkowo zdefiniowano własną charakterystykę o 128 bitowym adresie UUID 42821a40-e477-11e2-82d0-0002a5d5c51bcd20c480-e48b-11e2-840b-0002a5d5c51b, którą przyporządkowano do istniejącej już usługi. Zainicjalizowano również protokół komunikacji I²C wykorzystywany do komunikacji z potencjometrem cyfrowym MCP4018.[21]

Aplikację zmodyfikowano poprzez dodanie odpowiednich symboli preprocesora, zgodnych z konfiguracją projektowanego urządzenia:

- 1) HS_SPEED_XTAL=HS_SPEED_XTAL_32MHZ.
- 2) LS_SOURCE=LS_SOURCE_INTERNAL_RO.
- 3) SMPS_INDUCTOR=SMPS_INDUCTOR_10uH.
- 4) BLUENRG2_DEVICE.

5.2.1. Odbiór danych wejściowych

Mikrokontroler BluNRG oparty jest o 32 bitowy procesor ARM Cortex-M. Dane wejściowe pochodzące z przetwornika analogowo-cyfrowego, organizowane są jako pół słowa (z ang. Half Word) i przechowywane są w formacie uint16_t w buforze cyklicznym o długości 20, która zapełniana jest w sposób ciągły przez DMA. Dwie flagi wystawiane przez DMA informują o stanie napełnieniu bufora.

5.2.2. Algorytm przesyłu danych

Przesył danych wykonuje się w głównej pętli programu. Rozpoczyna się od każdorazowej inicjalizacji stosu BLE, który umożliwia komunikację, obsługę zdarzeń między urządzeniami BLE oraz ustawienia urządzenia w trybie umożliwiającym nawiązanie połączenia. Kolejno dane odebrane z przetwornika ADC (typu `uint16_t`), w przypadku wystąpienia flagi połowicznego zapełnienia tablicy ADC konwertowane są na dwa bajty typu `uint8_t`, które umieszczone są w tablicy danych do notyfikacji (Załącznik A, funkcja `APP_Tick1`) w kolejności MSB, LSB. Następnie flaga jest usuwana, wartość charakterystyki związanej z danymi ADC jest aktualizowana (Załącznik A, Funkcja `ADC_Update1()`). Proces powtarza się przy zapełnieniu całkowitym bufora DMA, zgodnie z techniką podwójnego buforowania (z ang. *ping-pong buffering*).

5.3. Działanie urządzenia w sieci MESH

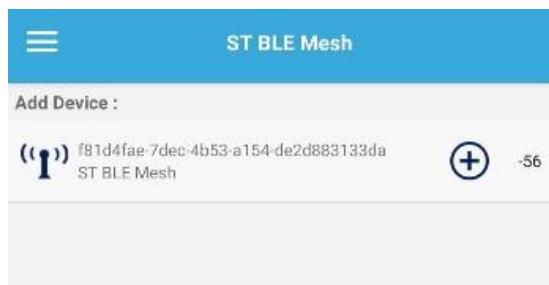
Jednym z podstawowych wymagań postawionych przed projektowanym urządzeniem jest przystosowanie do pracy w sieci minimum dwóch tego rodzaju urządzeń. Mikrokontrolery z rodziny BlueNRG, do której należy mikrokontroler BlueNRG-248 mogą pracować połączone siecią BlueNRG MESH [13,14]..

Rozwiązanie to jest dostępne dla rodziny produktów BlueNRG, w której skład wchodzi użyty do projektu mikrokontrolera BlueNRG-248N.

W projekcie wykorzystano przykładową aplikację Smart_Plug, dostarczaną w zestawie STW-BNRG-MESH. Aplikację zmodyfikowano poprzez dodanie odpowiednich symboli preprocesora, zgodnych z konfiguracją projektowanego urządzenia:

- 1) HS_SPEED_XTAL=HS_SPEED_XTAL_32MHZ.
- 2) LS_SOURCE=LS_SOURCE_INTERNAL_RO.
- 3) SMPS_INDUCTOR=SMPS_INDUCTOR_10uH.
- 4) BLUENRG2_DEVICE.

Program został wgrany za pomocą programatora ST-LINKV2. Następnie opracowano aplikację, przeznaczoną na urządzenia z systemem ST-BLE-MESH, która umożliwia stworzenie sieci mesh pomiędzy urządzeniami z rodziny BlueNRG2. Na rysunku 5.7. przedstawiono ekran główny z aplikacji ST-BLE-MESH, na którym widoczny jest mikrokontroler zastosowany w układzie.



Rys. 5.7. Ekran główny aplikacji ST-BLE-MESH przedstawiający gotowy do włączenia w sieć MESH mikrokontroler BlueNRG-2 znajdujący się w projektowanym układzie

Po wykonaniu węzła wewnętrz aplikacji, urządzenie jest gotowe do pracy w sieci BlueNRG-MESH.

Mając do dyspozycji dwa układy posiadające mikrokontroler z rodziny BlueNRG oraz korzystając z biblioteki SimpleBLE przykładowej aplikacji `notyfie_multi.cpp` stworzono aplikację (załącznik B, aplikacja `notyfie_multi`) umożliwiającą:

- połączenie się z dwoma urządzeniami BLE,
- 5 s subskrypcję jednej z dostępnych charakterystyk typu notyfikacyjnego,
- zapis danych do pliku.

Na rysunku 5.8. przedstawiono widok z aplikacji konsolowej z połączonymi jednocześnie dwoma urządzeniami:

- 1) BlueNRG [c4:de:cc:7B:6F:aa] - mikrokontroler znajdujący się w projektowanym wzmacniaczu.
- 2) BlueNRG [d6:e7:04:fd:6c:29] - mikrokontroler znajdujący się w zestawie deweloperskim STEVAL-BCN002V1B.

```

Konsola debugowania programu Microsoft Visual Studio
[0]
[1] BlueNRG [c4:de:cc:7b:6f:aa]
[2] BlueNRG [d6:e7:04:fd:6c:29]

Please select a device to connect to (0-2): 2
Connecting to BlueNRG [d6:e7:04:fd:6c:29]
Successfully connected, printing services and characteristics..
The following services and characteristics were found:
[0] 00001800-0000-1000-8000-00805f9b34fb 00002a00-0000-1000-8000-00805f9b34fb
[1] 00001800-0000-1000-8000-00805f9b34fb 00002a01-0000-1000-8000-00805f9b34fb
[2] 00001800-0000-1000-8000-00805f9b34fb 00002a04-0000-1000-8000-00805f9b34fb
[3] 00001801-0000-1000-8000-00805f9b34fb 00002a05-0000-1000-8000-00805f9b34fb
[4] 42821a40-e477-11e2-82d0-0002a5d5c51b cd20c480-e48b-11e2-840b-0002a5d5c51b

Please select a characteristic to read (0-4): 4
Scan started.
Found device: BlueNRG [c4:de:cc:7b:6f:aa]
Scan stopped.The following devices were found:
[
0] BlueNRG [c4:de:cc:7b:6f:aa]
Please select a device to connect to (0-0): 0
Connecting to BlueNRG [c4:de:cc:7b:6f:aa]
Successfully connected, printing services and characteristics..
The following services and characteristics were found:
[0] 00001800-0000-1000-8000-00805f9b34fb 00002a00-0000-1000-8000-00805f9b34fb
[1] 00001800-0000-1000-8000-00805f9b34fb 00002a01-0000-1000-8000-00805f9b34fb
[2] 00001800-0000-1000-8000-00805f9b34fb 00002a04-0000-1000-8000-00805f9b34fb
[3] 00001801-0000-1000-8000-00805f9b34fb 00002a05-0000-1000-8000-00805f9b34fb
[4] 42821a40-e477-11e2-82d0-0002a5d5c51b cd20c480-e48b-11e2-840b-0002a5d5c51b

Please select a characteristic to read (0-4): 4

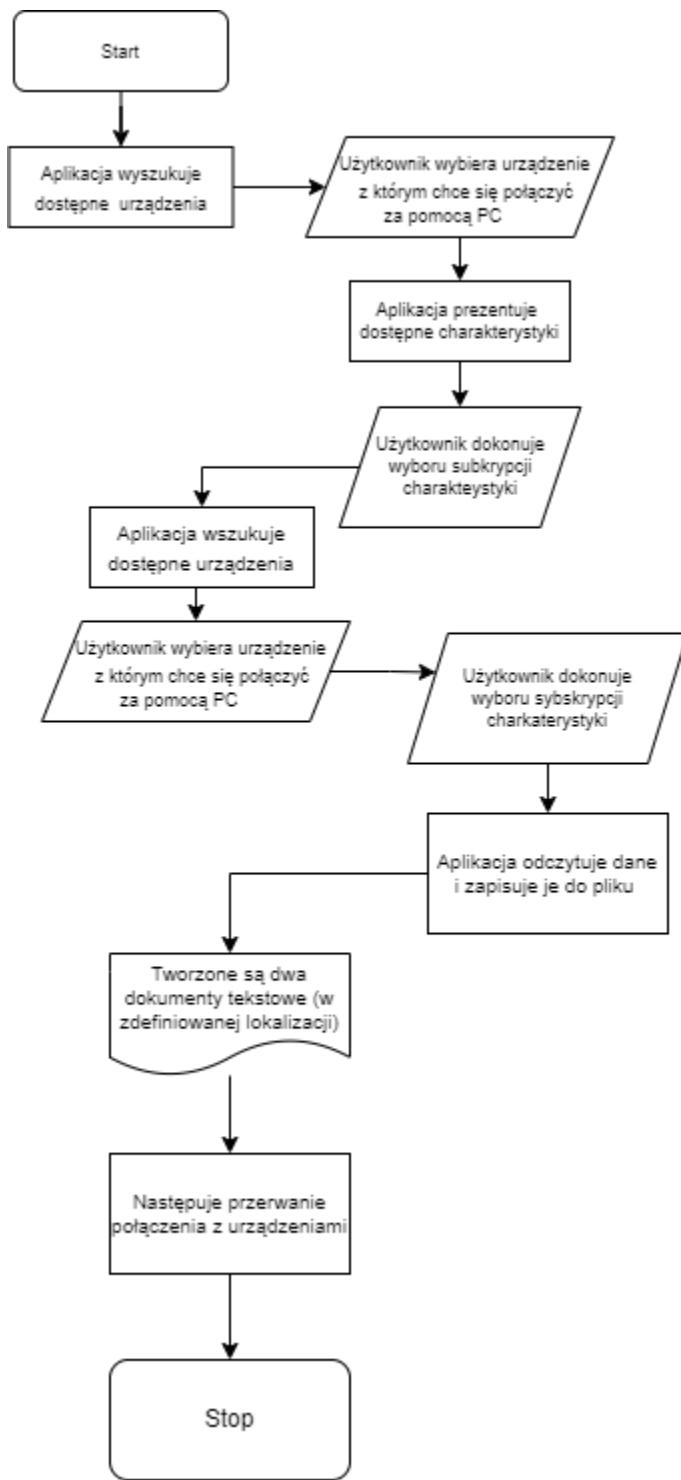
```

Annotations in the screenshot:

- Udane połączenie się z mikrokontrolerem BlueNRG-2 (Connected to BlueNRG-2 microcontroller)
- Dostępne charakterystyki (Available characteristics)
- Udane połączenie się z drugim mikrokontrolerem BlueNRG-2 (Connected to second BlueNRG-2 microcontroller)
- Subskrypcja charakterystyki (Characteristic subscription)

Rys. 5.8. Widok z aplikacji konsolowej, przedstawiający udane połączenie się z dwoma mikrokontrolerami BlueNRG oraz subskrypcję jednej z charakterystyk

Na rysunku 5.9. przedstawiono schemat przepływu danych w aplikacji do komunikacji z dwoma urządzeniami BLE.



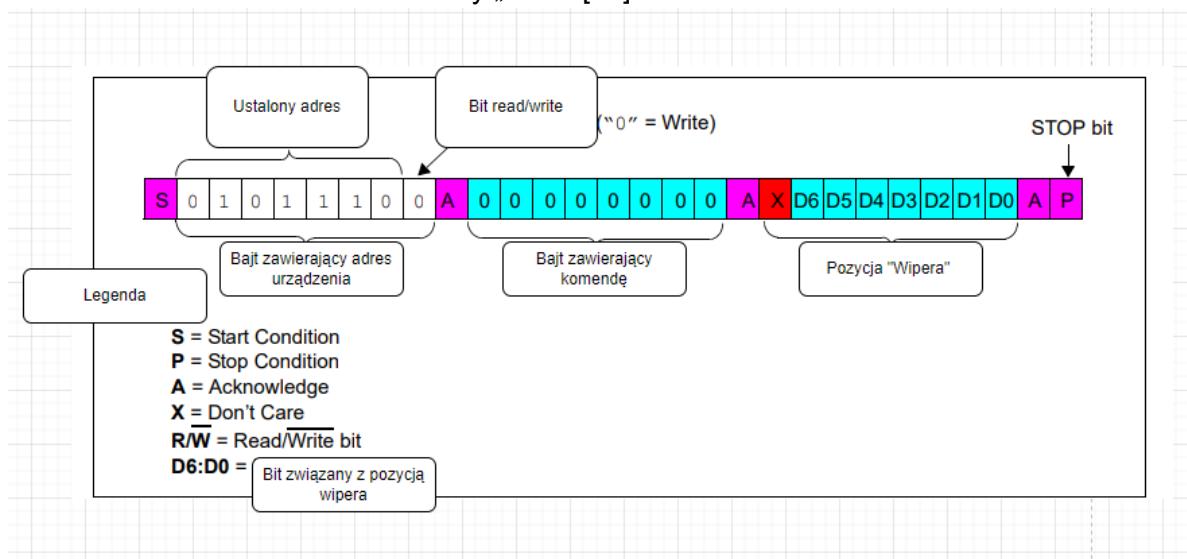
Rys. 5.9. Schemat przepływu danych w aplikacji służącej do połączenia się z dwoma urządzeniami BLE

Aplikację zdecydowano się zrealizować ze względu na bardzo słabą wartość RSSI mikrokontrolera w układzie STEVAL-BCN002V1B, co uniemożliwiło zrobienie węzła w aplikacji ST-BLE-MESH.

5.4. Komunikacja z potencjometrem cyfrowym MCP40D18T-104E/LT

Zgodnie z notą katalogową układu, MCP40D18T-104E/LT d [21] działa jedynie jako „Slave” oraz obsługuje tylko dwie komendy „Read/Write”, dzięki którym kontrolować można jego działanie za pomocą interfejsu I²C poprzez urządzenie MASTER (BlueNRG-2).

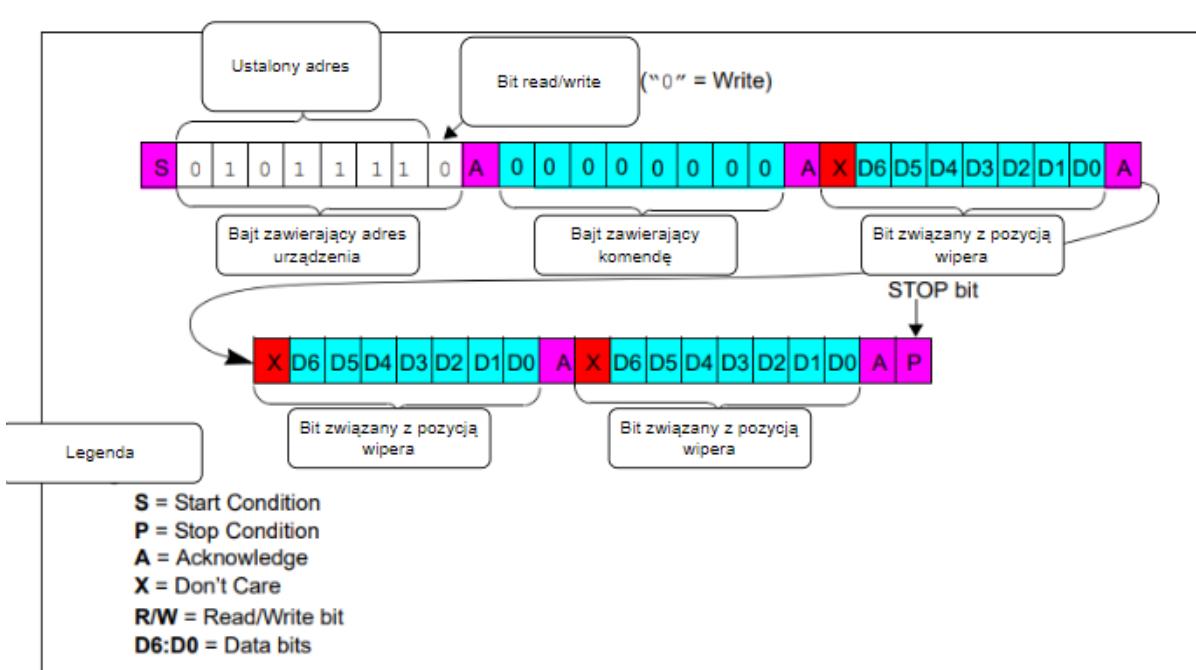
Komenda „Write” służy do zmiany wartości rejestru potencjometru. Na rysunku 5.10. przedstawiono schemat ramki komendy „Write”[21].



Rys. 5.10. Schemat ramki komendy „Write”

Na podstawie wyżej przedstawionego schematu napisano funkcję `I2C_WRITE_MC40D18T` należącą do oprogramowania mikrokontrolera (*Załącznik A, funkcja I2C_WRITE_MC40D18T*), która odwzorowuje komendę „Write”.

Komenda „Read” służy do odczytu pozycji suwaka. Na rysunku 5.11. przedstawiono schemat ramki komendy „Read”.



Rys. 5.11. Schemat ramki komendy „Read”

Na podstawie wyżej przedstawionego schematu napisano funkcję `int I2C_READ_MC40D18T`, należącą do oprogramowania mikrokontrolera (*Załącznik A, funkcja I2C_READ_MC40D18T*), która odwzorowuje komendę „Read”.

6. Prototyp urządzenia

W rozdziale tym przedstawiono:

- proces projektowania obwodu PCB,
- opis montażu urządzenia,
- całkowity kosztorys urządzenia
- projekt obudowy urządzenia

6.1. Stworzenie projektu obwodu drukowanego

Na wyjściu każdego z bloków dołączono rezystor separacyjny o wartości 0Ω oraz punkt testowy. Miało to na celu zapewnienie możliwości izolacji poszczególnych bloków w aspekcie łatwiejszego testowania układu. Projekt układu PCB został wykonany z wykorzystaniem oprogramowania KiCad (open source, GNU GPL V2).

Footprinty elementów w projekcie urządzenia, które nie były dostępne w standardowych bibliotekach, zostały pobrane ze stron <https://rs.componentsearchengine.com/> oraz <https://www.snapeda.com/.> Poprawność wykonania każdego footprintu, użytego w schemacie, została sprawdzona w oparciu o dane dotyczące obudów komponentów zamieszczonych przez producentów w notach katalogowych. Schemat ideowy zaprojektowanego urządzenia został zamieszczony w załączniku.(załącznik D. Schemat ideowy).

Utworzony schemat został zweryfikowany pod względem elektrycznym (ERC). Testy nie wykazały błędów.

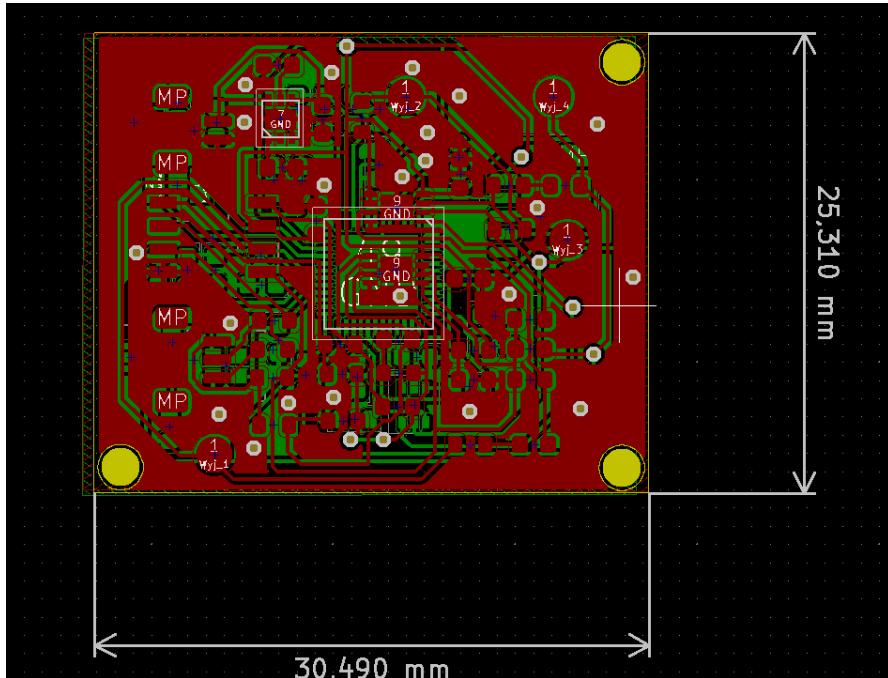
Prototyp urządzenia wykonano w rozmiarach większych niż ustalone w założeniach projektowych (rysunek 6.1.).



Rys. 6.1. Pierwotna wersja prototypu wzmacniacza

Takie podejście miało na celu ułatwienie testowania układu pod kątem poprawności jego działania oraz pozwalało na stosunkowo proste wprowadzanie modyfikacji w przypadku stwierdzenia nieprawidłowości w jego funkcjonowaniu. Po przeprowadzeniu testów opisanych

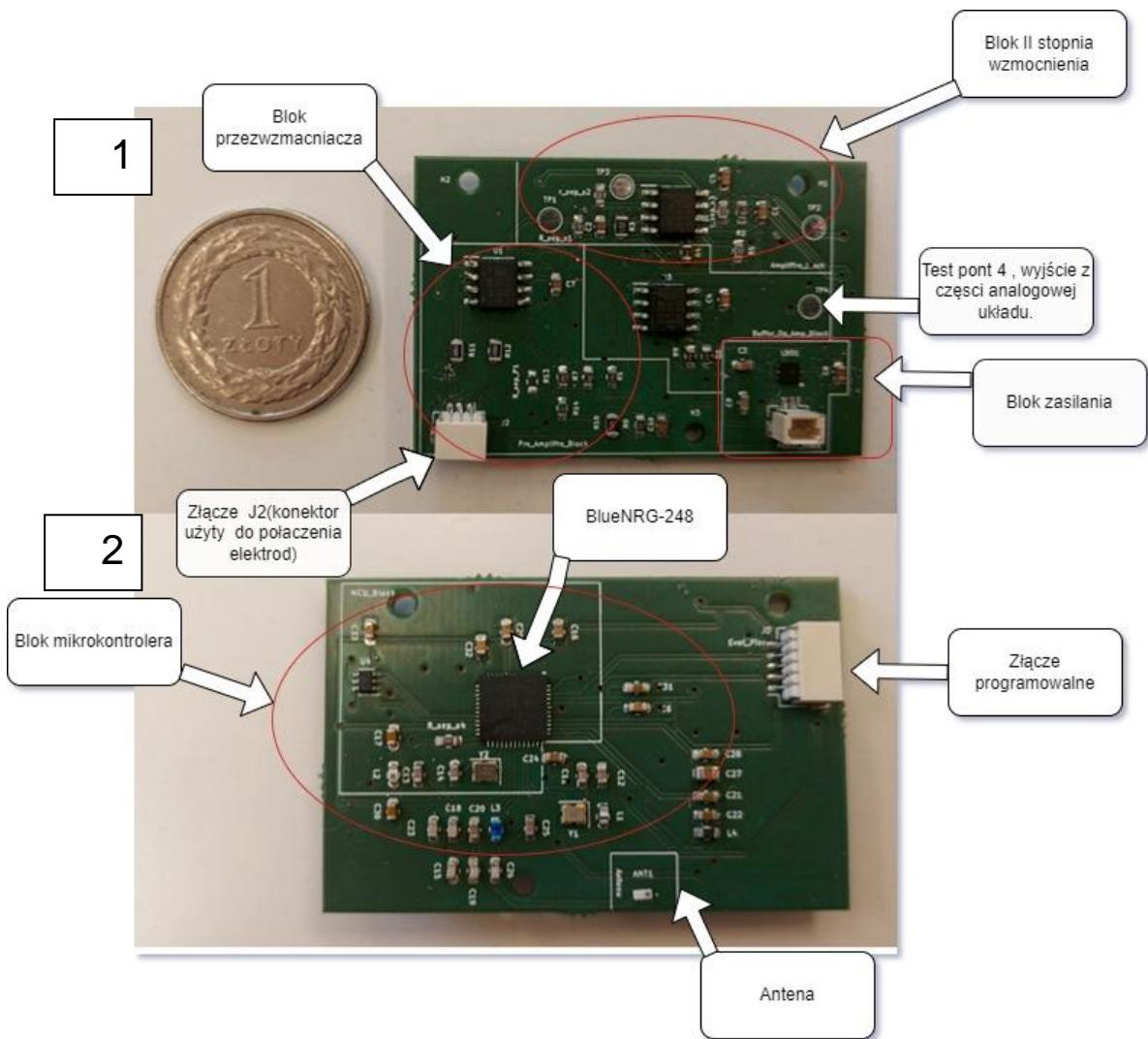
w rozdziale 7, elementy zostały rozmieszczone tak, aby rozmiar układu był możliwie jak najmniejszy, poprawiając błędy opisane w rozdziale 6.2.1 (rysunek 6.2.). W procesie projektowania PCB zostały uwzględnione uwagi producentów dotyczące zalecanego umiejscowienia poszczególnych układów. Elementy analogowe oraz cyfrowe zostały oddzielone poprzez umieszczenie ich po przeciwnych stronach płytki, aby zminimalizować ryzyko powstania zakłóceń sygnału. Projekt układu PCB zrealizowany został w firmie AISLER B.V (Lamieres,Holandia).



Rys. 6.2. Finalna wersja prototypu wzmacniacza

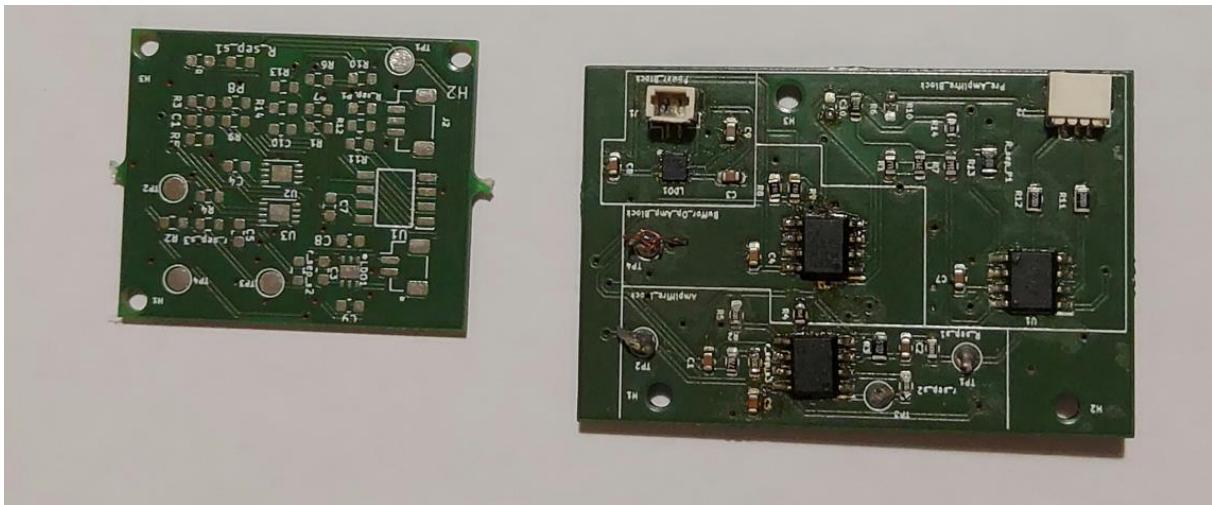
6.2. Montaż komponentów

Przed przystąpieniem do montażu, opakowania w których znajdowały się komponenty zostały podpisane numerami z BOM, zgodnie z oznaczeniami zastosowanymi w układzie. Ułatwiło to tym samym identyfikację elementów dzięki, czemu czas potrzebny na montaż uległ skróceniu. Następnie za pomocą lutownicy hot-air oraz kolbowej umieszczono na odpowiednich padach poszczególne elementy. Na tym etapie również zdecydowano się umieścić na punktach testowych miedziane złącza, które miały zapewnić stabilne połączenie między układem a przyrządem pomiarowym i w ten sposób ograniczyć wpływ błędu ludzkiego na później przeprowadzone testy układu. Gotowy układ przedstawiono na rysunku 6.3.



Rys. 6.3. Prototyp układu: 1) zdjęcie przedstawia warstwę wierzchnią obwodu PCB, 2) zdjęcie przedstawia warstwę dolną

Na rysunku 6.4 przedstawiono porównanie I wersji prototypu urządzenia oraz II wersji układu PCB.



Rys.6.4. Porównanie I wersji prototypu urządzenia oraz II wersji obwodu PCB

6.2.1. Napotkane problemy w realizacji fizycznej

Pierwszym zauważonym błędem był brak odpowiedniego poziomu napięcia podukładu wirtualnej ziemi na wyjściu wzmacniacza operacyjnego U3, który użyty został do budowy układu referencji. W projekcie zauważono, iż błędnie dobrano footprint dla układu U3 oraz U2, tj. wzmacniacz operacyjny AD8532ARZ (*Załącznik D. Schemat ideowy*). Spowodowane było to przypisaniem układowi AD8538ARZ footprintu wzmacniacza operacyjnego AD8539ARZ, który był planowany do wykorzystania na wczesnym etapie projektowania prototypu. Oba komponenty posiadają obudowę SOIC-8N, więc nie wykryto nieprawidłowości w rozmiarze w trakcie projektowania PCB. Natomiast mają one wejścia umiejscowione w różnej kolejności, co powodowało nieprawidłowość w działaniu układu wirtualnej masy. Zdemontowano układ U3 oraz U2 i w jego miejsce zamontowano dostępny w Zakładzie Inżynierii Biomedycznej układ AD8542ARZ, który posiada identyczny rozkład wejść (z ang. pinout) jak AD8539ARZ, dzięki czemu na wyjściu układu otrzymano zakładany poziom napięcia VGND.

Przeoczono również informację w nocy katalogowej układu BlueNRG-2 [16] o konieczności dodania zewnętrznych rezystorów podciągających do poprawnego funkcjonowania interfejsu I²C. Dlatego dolutowano dodatkowe rezystory podciągające do padów elementów C32 i C16, które połączono za pomocą zworki do ścieżek SCL, SDA układu U4 (*Załącznik D. Schemat ideowy*).

6.3. Kosztorys poszczególnych bloków

Przed przystąpieniem do zakupu komponentów na potrzeby realizacji prototypu wzmacniacza przygotowano BOM. Na podstawie danych w nim zawartych przygotowano tabelę 6.1., w której przedstawiono całkowite koszty realizacji prototypu podzielone na wyróżnione w projekcie bloki. W kosztorysie uwzględniono jedynie komponenty wykorzystane w realizacji fizycznej układu.

Tab. 6.1. Całkowite podsumowanie kosztu realizacji fizycznej układu

Nazwa	Numer seryjny	Liczba	Cena
Blok II stopnia wzmacnienia			
Kondensator	-----	2	0.10 zł
Potencjometr	MCP40D18T-104E/LT	1	3.24 zł
Rezystor	-----	5	0.10 zł
Wzmacniacz operacyjny	AD8542ARZ-REEL	1	3.17 zł*
Suma			7.11 zł
Blok przedwzmacniacza			
Kondensator	-----	3	0.10 zł
Rezystor	-----	11	0.10 zł
Wzmacniacz instrumentalny	AD623ARZ	1	15.80 zł
Wzmacniacz operacyjny	AD8542ARZ-REEL	1	3.17 zł*
Złącze	JST SH 1.0 3 PIN	1	2 zł
Suma			22.37 zł
Blok mikrokontrolera			
Antena	ANT016008LCS2442MA	1	3.20 zł
Cewka	LQG18HN2N2S00D	1	0.88 zł
Cewka	LQW18AN3N9C10D	1	0.88 zł
Cewka	LQW18AN3N6C00D	1	0.88 zł
Cewka	LQM18DH100M70L	1	0.88 zł
Mikrokontroler	BlueNRG-248N	1	20.20 zł
Kondensator	-----	23	0.10 zł
Rezystor	-----	2	0.10 zł
Kryształ 32Mhz	CX2016DB32000D0WZRC1	1	4.68 zł
Złącze	JST SH 1.0 mm PIN 5	1	2 zł
Suma			36.10 zł
Blok zasilania			
Akumulator	ACCU-LP752030/CL	1	18 zł
Kondensator	-----	3	0.10 zł
Linowy stabilizator	ADP7118	1	13.15 zł
Wzmacniacz operacyjny	AD8542ARZ-REEL	1	3.17* zł
Złącze	JST SH 1.0 2 PIN	1	1.72 zł
Suma			36.34 zł
Blok filtracji (filtr dolnoprzepustowy)			
Kondensator	-----	2	0.10 zł
Rezystor	-----	2	0.10 zł
Suma			0.4 zł
Blok filtracji (filtr górnoprzepustowy)			
Kondensator	-----	1	0.10 zł
Rezystor	-----	3	0.10 zł
Wzmacniacz operacyjny	AD8542ARZ-REEL	1	3.17 zł
Suma			3.57 zł

*Koszt wzmacniaczy operacyjnych rozłożono na poszczególne bloki zgodnie z ich rozmieszczeniem

Koszt realizacji fizycznej:

- bloku II stopnia wzmacnienie wyniósł 6% ustalonego budżetu,
- boku przedwzmacniacza wyniósł 20.3% ustalonego budżetu,

- bloku mikrokontrolera wyniósł 32.8% ustalonego budżetu,
- bloku zasilania wyniósł 33% ustalonego budżetu,
- bloku filtracji wyniósł 3.2% ustalonego budżetu.

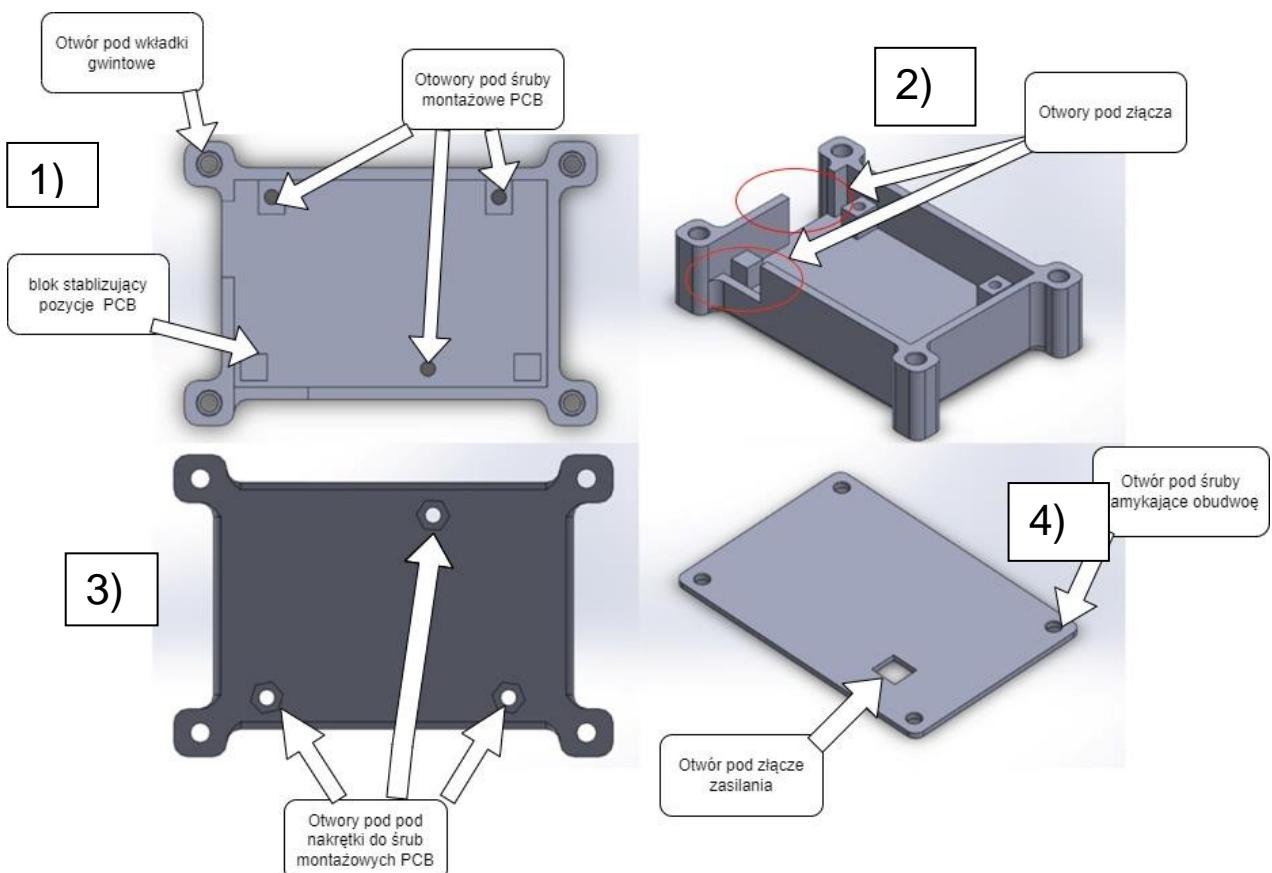
Podsumowując dane w tabeli 6.1. koszt realizacji fizycznej całego prototypu wyniósł 105.59 zł, co wynosi 96 % zakładanego maksymalnego budżetu przeznaczonego na realizację prototypu.

6.4. Projekt obudowy urządzenia

Na wstępnie sformułowano założenia projektowe dotyczące obudowy:

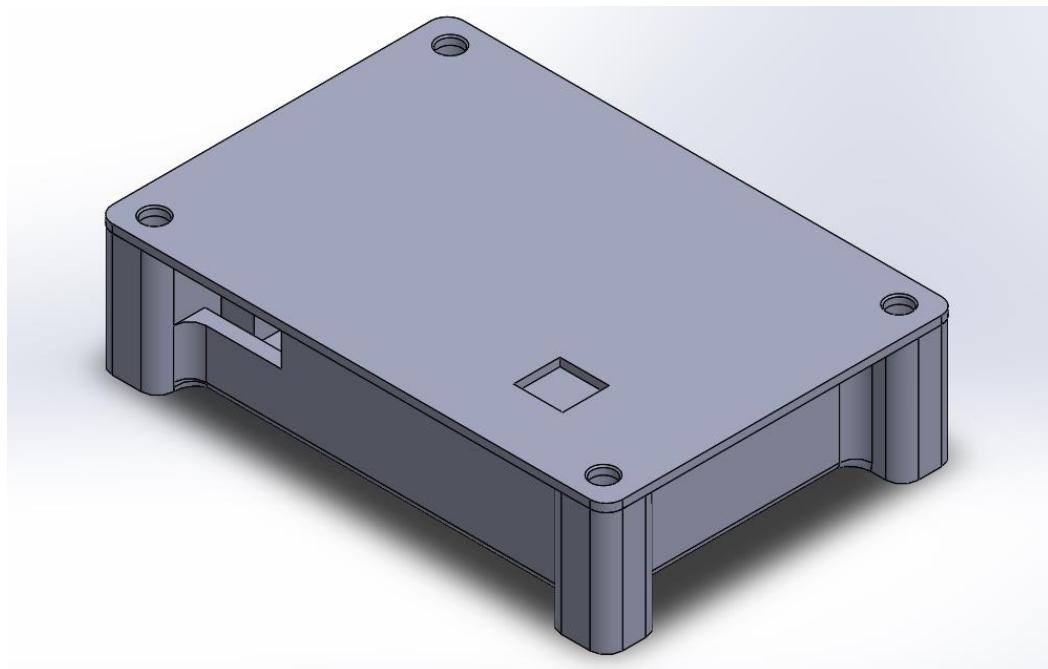
- dostosowanie projektu do charakteru wybranej metody wytwarzania (druk 3D),
- obudowa powinna chronić urządzenie,
- układ PCB powinien być stabilnie zamontowany,
- obudowa powinna mieć rozmiary nie przekraczające wymiarów 70x20x45 mm,
- powinien zostać zapewniony dogodny dostęp do wyprowadzonych złącz.

Projekt obudowy wykonany został w programie SolidWorks. Do korpusu obudowy wprowadzono, wkładki gwintowe (rysunek 6.7., czerwony okrąg), które umożliwiły zamknięcie obudowy. Na rysunku 6.5. zostały przedstawione części modelu obudowy. Do zamknięcia obudowy użyto śrubek M2.5, natomiast do montażu obwodu PCB użyto śrubek M2.



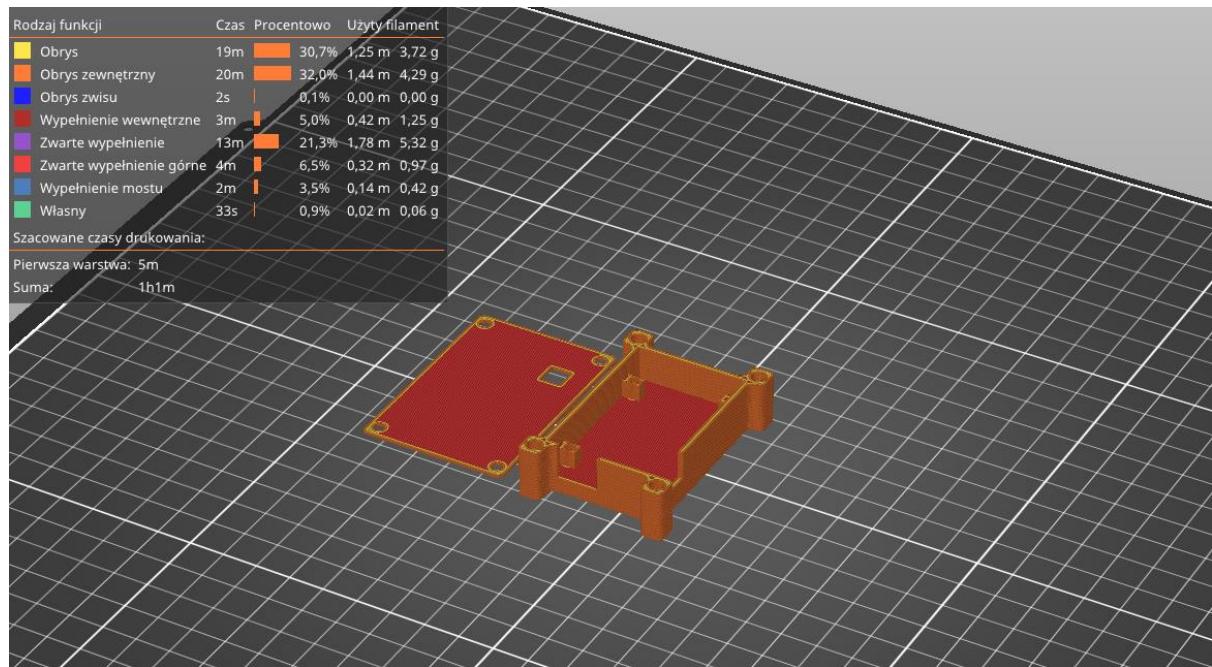
Rys. 6.5. Przedstawienie elementów obudowy: 1) korpus obudowy w rzucie z góry; 2) rzut z boku korpusu, gdzie widoczne są otwory wykonane pod śruby montażowe PCB jak również otwory zapewniające dostęp do wyprowadzonych złącz; 3) korpus obudowy w rzucie z dołu, na którym widoczne są hexagonalne otwory wykonane pod nakrętki oraz otwory montażowe; 4) pokrywa obudowy wraz z złączem zasilania oraz otworami wykonanymi pod śruby zamykające obudowę

Na rysunku 6.6. przedstawiono widok złożenia części obudowy.



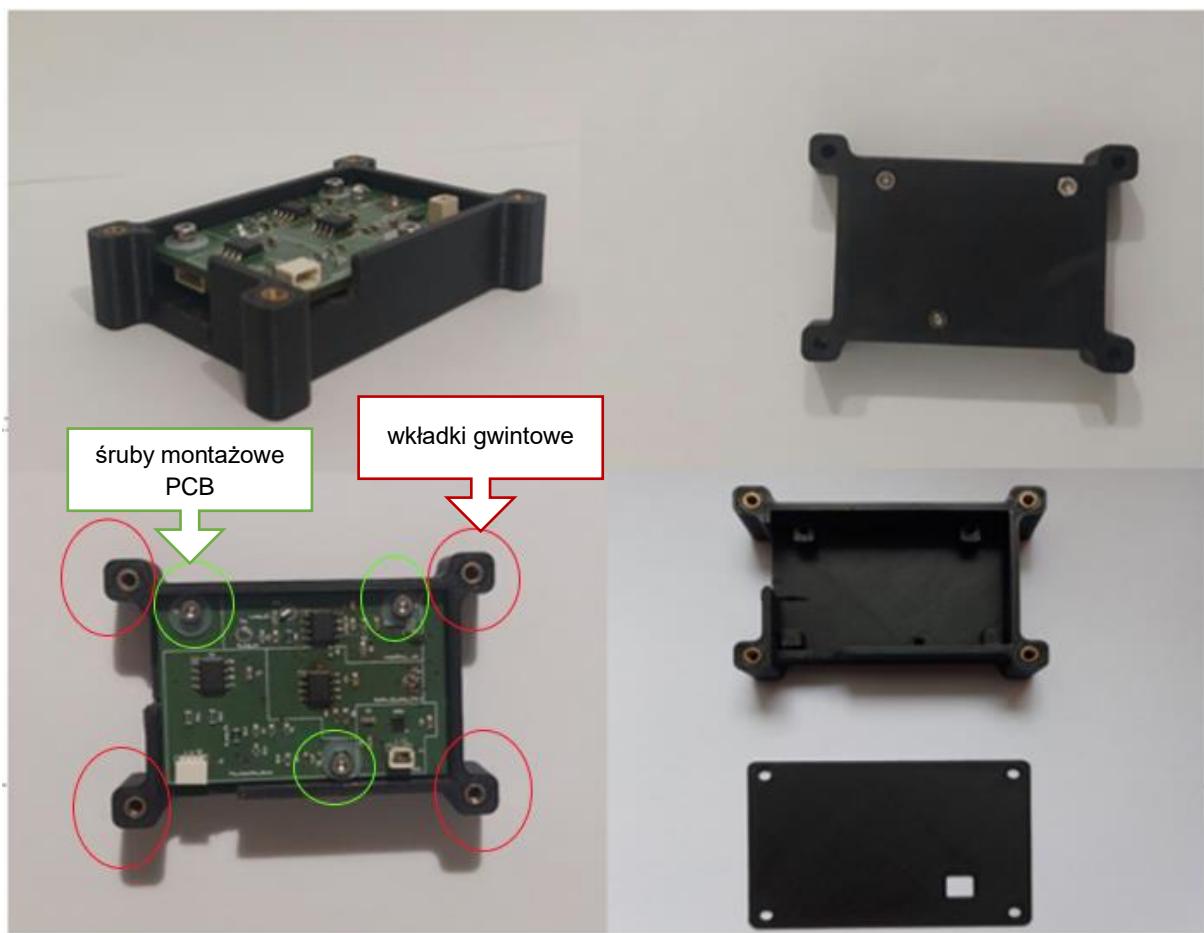
Rys. 6.6. Widok złożenia projektu obudowy urządzenia

W celu dokonania wydruku 3D projektu wyeksportowano pliki STL, które zaimportowano do programu PrusaSlicer. Na rysunku 6.7. przedstawiono widok oraz dane dotyczące druku części obudowy.



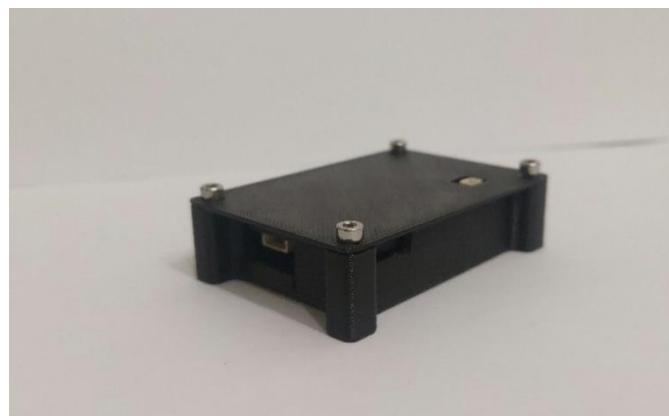
Rys. 6.7. Widok części składowych obudowy w programie PrusaSlicer

Wydrukowane części złożono. Na rysunku 6.8. przedstawiono wydrukowane części obudowy wraz z umieszczonymi wkładkami gwintowanym oraz zamontowaną płytka obwodu drukowanego.



Rys. 6.8. Rozłożone części obudowy z zamontowaną płytą obwodu drukowanego

Na rysunku 6.9. przedstawiono złożoną obudowę wraz z zamontowanym układem.



Rys. 6.9. Złożona obudowa wraz z zamontowanym układem

W załączniku E zostały przedstawione rysunki techniczne poszczególnych elementów obudowy oraz rysunek złożeniowy.

6.5. Pierwsze uruchomienie układu

Pierwsze uruchomienie układu miało na celu sprawdzenie poprawności działania bloku zasilania oraz mikrokontrolera znajdującego się w projektowanym wzmacniaku. W tym celu użyto zasilacza BK Precision Triple Output Dc Power Supply 165, który ustawiono na napięcie

3.7 V oraz organicznie prądowe 30 mA aby zminimalizować ryzyko uszkodzenia komponentów w przypadku wystąpienia błędów w układzie. W pierwszej kolejności programator STLink-V2 podłączono do złącza oznaczonego w układzie jako J3. Następnie uruchomiono aplikację BlueNRG-1 ST-Link-Utility i poprawnie połączono się z mikrokontrolerem. W dalszej kolejności wgrano przygotowane oprogramowanie mikrokontrolera, które umożliwia wykrycie oraz połączenie się z urządzeniem. Uruchomiono aplikację ST BLE Toolbox i wyszukano urządzenie. Udało się z nim połączyć oraz odczytać podstawowe dane takie, jak nazwa, siła transmisji oraz charakterystyka urządzenia.

7. Testy urządzenia

W rozdziale tym opisano sposób testowania układu, wykonanego oprogramowania oraz przedstawiono wyniki przeprowadzonych testów. Otrzymane wyniki w dalszej części poddano analizie i ocenie według przyjętych założeń dotyczących parametrów urządzenia.

Na wstępie należy zaznaczyć, iż regulacja poziomu wzmacnianie nie działała poprawnie. Układ działał w trakcie przeprowadzanych testów w domyślnym ustawieniu potencjometru cyfrowego, skutkującym wzmacnieniem teoretycznym układu wynoszącym 4000.

7.1. Test pomiaru napięcia zasilania

Przeprowadzono test sekcji zasilania. W czasie weryfikacji poprawności zasilania poszczególnych elementów aktywnych układ zasilany był za pomocą zasilacza BK Precision Triple Output Dc Power Supply 1651. Generator nastawiono na napięcie 3.7 V a organicznie prądowe na 40mA. Do pomiarów wykorzystany został multimeter cyfrowy Kyrotisu model 1009. W tabeli 7.1. przedstawiono otrzymane wyniki.

Tab. 7.1. Wykaz napięcia zasilania na poszczególnych komponentach aktywnych urządzenia

Nazwa elementu	Oznaczanie w układzie	Otrzymany rezultat
AD623	U1	3.298 V
AD8538	U3	3.291 V
AD8538	U2	3.293 V
ADP7118	LDO1	3.693 V (napięcie wejściowe na układ)
BlueNRG2	IC3	3.303V
MCP40D18	U4	3.301V

Napięcia zasilające wszystkie komponenty są poprawne.

7.2. Pomiar charakterystyki amplitudowo-częstotliwościowej

7.2.1. Filtr górnoprzepustowy

Przeprowadzono pomiary wartości, w celu wyznaczenia charakterystyki amplitudowo-częstotliwościowej na wyjściu filtra górnoprzepustowego.

W celu przeprowadzenia testu zbudowano stanowisko pomiarowe składające się z następujących przyrządów: dzielnika napięcia złożonego z rezystorów R1 i R2, generatora funkcji MXG-9802, zasilacza BK Precision Triple Output Dc Power Supply 1651 oraz oscyloskopu Tektronix MSO 2012 B Mixed Signal Oscilloscope (rysunek 7.1.).



Rys. 7.1. Schemat blokowy stanowiska pomiarowego

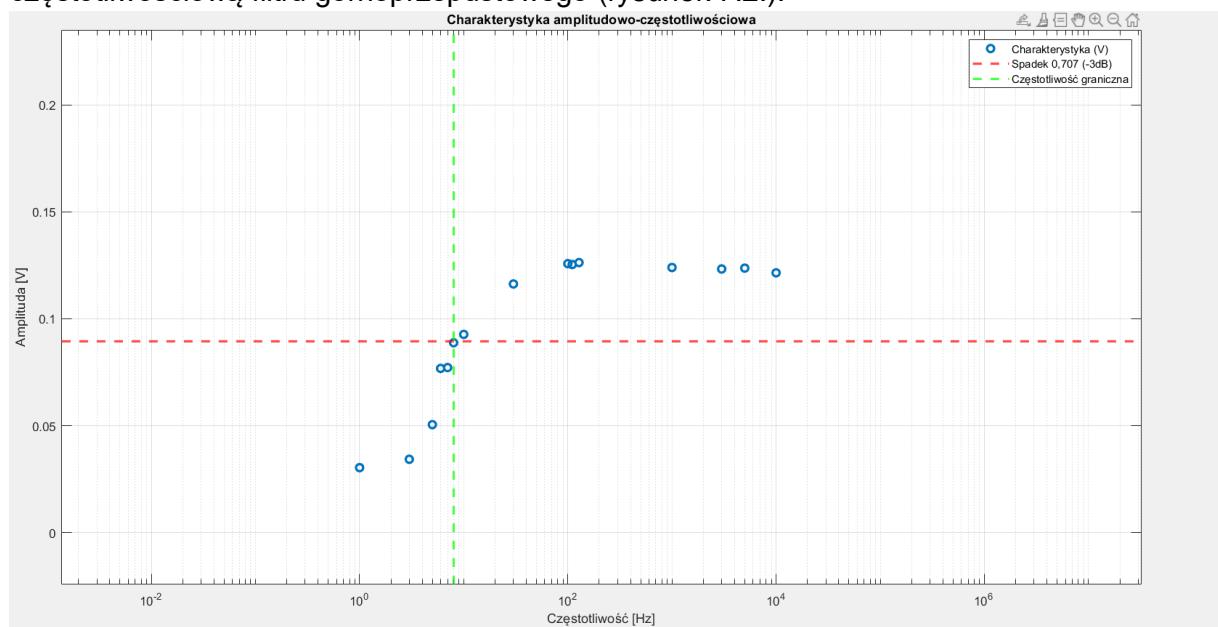
Na wejście badanego wzmacniacza zadawany był sygnał sinusoidalny o amplitudzie 5 V i zmiennej częstotliwości (zakres 1 Hz -10 kHz). W celu uzyskania wymaganej amplitudy sygnału wejściowego, sygnał z generatora funkcyjnego (MXG-9802) podano na dzielnik napięcia ok. 10000:1. Do wykonania dzielnika posłużono się rezystorami $R_1=99.6 \text{ k}\Omega$ i $R_2=99.2 \text{ k}\Omega$ (tol. 1%). Nominalna wartość dzielnika napięcia wynosiła 0,005 V.

Pomiary charakterystyki amplitudowo-częstotliwościowej za filtrem górnoprzepustowym wykonano dla 15 częstotliwości w zakresie od 1Hz do 10 kHz (tabela 7.2.). Wynik każdego z pomiarów wyznaczono jako średnią arytmetyczną z trzech pomiarów.

Tab. 7.2. Wartość amplitudy sygnału w zależności od jego częstotliwości w punkcie Test Point 3 (wyjście z filtra górnoprzepustowego)

Częstotliwość sygnału [Hz]	Wartość amplitudy sygnału [mV]
1	30.4
3	34.3
5	50.5
6	76.8
7	77.2
8	88.8
10	92.7
30	116.3
100	125.8
110	125.4
128	126.3
1k	124.0
3k	123.3
5k	123.7
10k	121.5

Na podstawie danych z tabeli 7.2. wykonano charakterystykę amplitudowo-częstotliwościową filtru górnoprzepustowego (rysunek 7.2.).



Rys. 7.2. Charakterystyka amplitudowo-częstotliwościowa filtru górnoprzepustowego

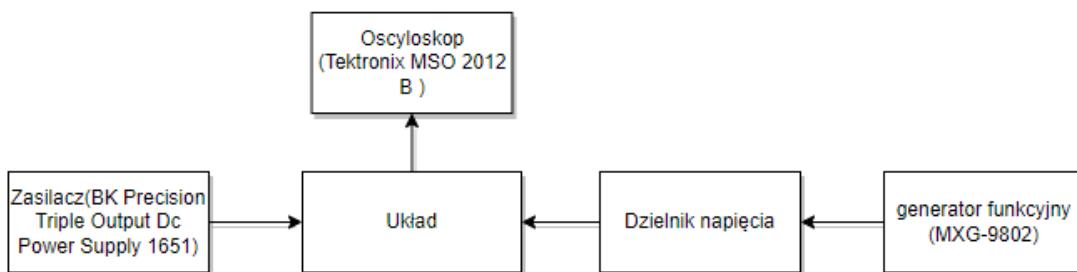
Maksymalna amplituda w okolicy środka pasma przenoszenia wynosi 126.3 mV. 3 dB spadek względem tej wartości przypada na częstotliwość 8 Hz.

Teoretyczna wartość odcięcia wynikająca z nominalnych wartości komponentów zastosowanych w projekcie wyniosła 6.639 Hz. Odchyłka wartości zmierzonej od wartości przyjętej w założeniach projektowych wyniosła 25%. Wartość ta nie mieści się w założonej tolerancji wynoszącej 20 %.

Wyznaczone wzmacnienie układu przedwzmacniacza w środku pasma wynosi 25.26. Teoretyczna wartość wzmacnienia wynikająca z nominalnych wartości komponentów zastosowanych w projekcie wyniosła 24.28. Odchyłka wartości zmierzonej od wartości przyjętej w założeniach projektowych wynosi 1.1% i nie mieści się w założonej tolerancji.

7.2.2. Filtr dolnoprzepustowy

W celu przeprowadzenia testu zbudowano stanowisko pomiarowe składające się z następujących przyrządów: dzielnika napięcia złożonego z rezystorów R1 i R2, generatora funkcji MXG-9802, zasilacza BK Precision Triple Output DC Power Supply 1651 oraz oscyloskopu Tektronix MSO 2012 B Mixed Signal Oscilloscope (rysunek 7.3.).



Rys. 7.3. Schemat blokowy stanowiska pomiarowego

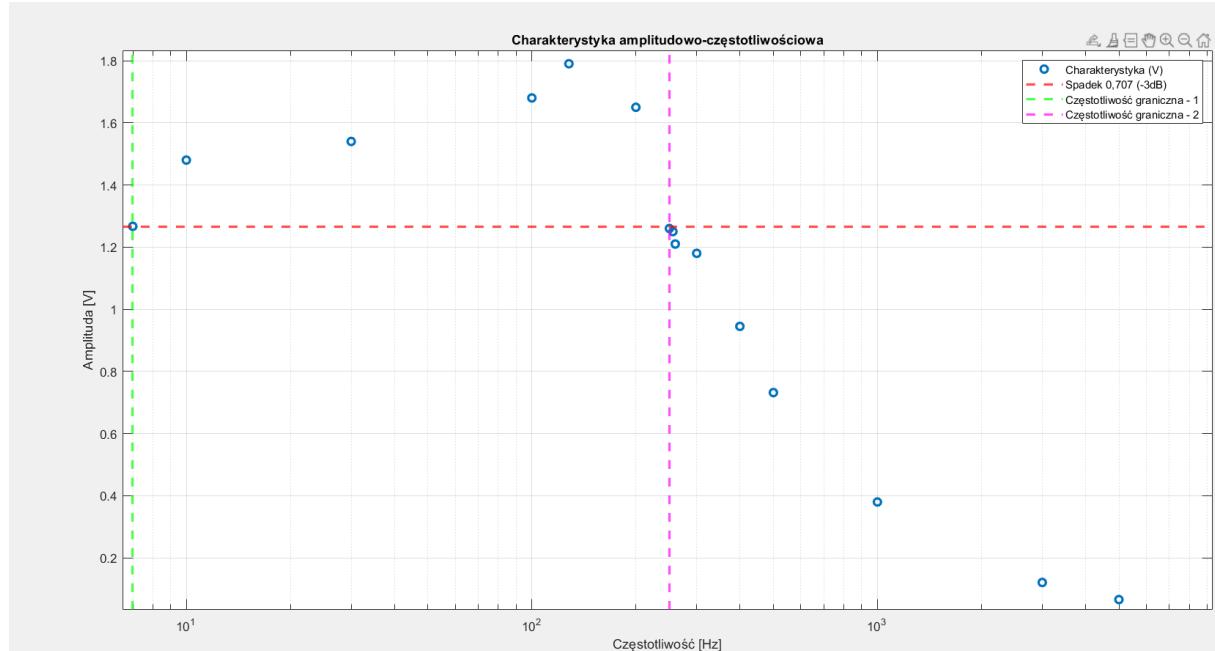
Na wejście badanego wzmacniacza zadawany był sygnał sinusoidalny o amplitudzie 5 V i zmiennej częstotliwości (zakres od 5 Hz - 10kHz). Dobrano wartości rezystorów $R_1=99,6 \Omega$ i $R_2=996,65\text{K}\Omega$ (tol.1%) tak, aby stosunek sygnału wejściowego do wyjściowego wynosił 100000:1. Obliczona teoretyczna amplituda sygnału na wyjściu dzielnika napięcia wyniosła 0,0005 V.

Pomiary charakterystyki amplitudowo-częstotliwościowej za filtrem dolnoprzepustowym (na wyjściu analogowej części układu) wykonano dla 16 częstotliwości w zakresie od 1 Hz do 10 kHz (tabela 7.3.). Wynik każdego z pomiarów wyznaczono jako średnią arytmetyczną z trzech pomiarów.

Tab. 7.3. Wartość amplitudy sygnału w zależności od jego częstotliwości w punkcie Test Point 2 (wyjście z filtra dolnoprzepustowego)

Częstotliwość sygnału	Wartość Amplitudy sygnału
7Hz	1,267V
10Hz	1,48 V
30Hz	1,54 V
100Hz	1,68V
128Hz	1,79 V
200Hz	1,65V
250Hz	1,26 V
256Hz	1,25 V
260Hz	1,21 V
300Hz	1,18V
400Hz	0,945 V
500Hz	0,732V
1KHz	0,380V
3KHz	0,121V
5KHz	0,066V
10KHz	0.042.5 V

Na podstawie danych z tabeli 7.3. wykonano charakterystykę amplitudowo-częstotliwościową filtru dolnoprzepustowego (rysunek 7.4.).

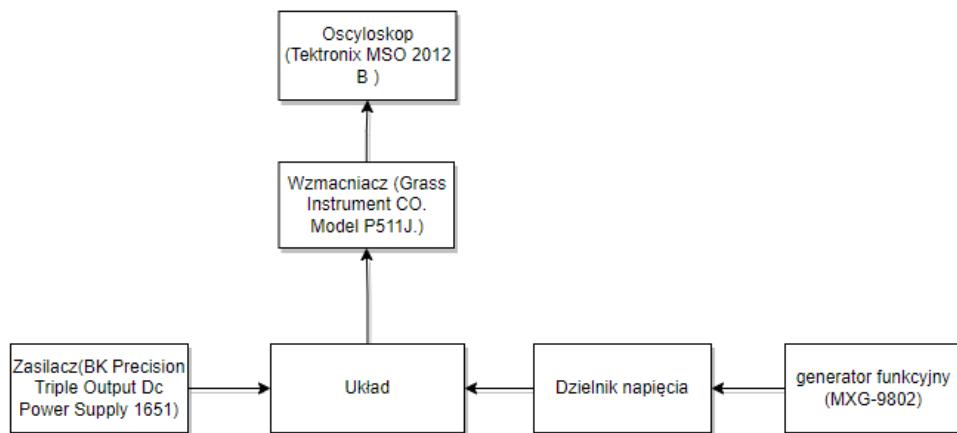


Rys. 7.4. Charakterystyka amplitudowo-częstościowa na wyjściu filtra dolnoprzepustowego (wyjście z toru pomiarowego wzmacniacza)

Analizując otrzymane wyniki można zauważyć, iż punkty 3dB spadku względem maksymalnej wartości amplitudy wyniosły 7Hz oraz 260Hz. Wyniki te są zgodne z oczekiwanymi oraz mieszczą się w przyjętym poziomie tolerancji dla pasma przepustowego wzmacniacza. Natomiast wzmacnianie w układzie dla potencjometru na 63 indeksie czyli połowie wartości potencjometru wyniosło 3582.

7.3. Badanie CMRR w układzie

Wartość współczynnika CMRR określono dla częstotliwości 128 Hz. Na oba wejścia układu zadano sygnał sinusoidalny o amplitudzie równej 95 mV (sygnał z generatora funkcyjnego MXG-9802 podany został na wejścia układu poprzez dzielnik napięcia w stosunku ok. 10:1 - R₂=99.6 kΩ, R₂=9.9 kΩ.) Amplitudę sygnału wejściowego zmierzono za pomocą oscyloskopu (Tektronix MSO 2012 B Mixed Signal Oscilloscope). Na rysunku 7.5. przedstawiono stanowisko pomiarowe do badania współczynnika CMRR w układzie.



Rys. 7.5. Schemat blokowy stanowiska pomiarowego do badania współczynnika CMRR w układzie

Za pomocą generatora funkcji MXG-9802 na dwa wejścia różnicowe wzmacniacza instrumentalnego nadano sygnał sinus o amplitudzie 1 V, i częstotliwości 128 Hz (środek pasma). Dokonano 10 pomiarów z czego wybrano pomiar o najwyższej wartości amplitudy sygnału wyjściowego wynoszący 0,106 V. Po uwzględnieniu nominalnego wzmacnienia wzmacniacza Grass P511, wyznaczone wzmacnienie sygnałów synfazowych w torze wzmacniacza projektowanego układu wyniosło 1.116. Współczynnik CMRR obliczono zgodnie z równaniem (7.1). Jego wartość wyniosła 70,137 dB. Wartość ta nie mieści się w założonym przedziale.

$$CMRR = 20 \log_{10} \left(\frac{Ad}{|Acm|} \right) dB \quad (7.1)$$

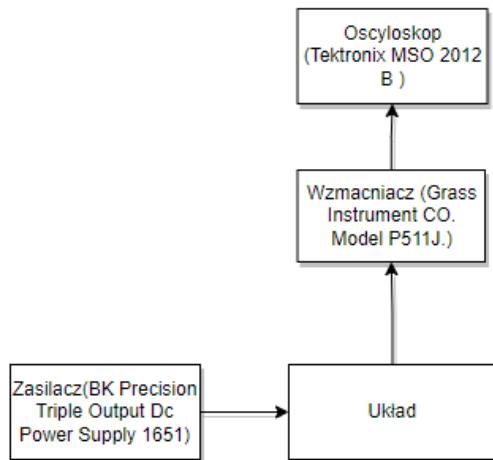
gdzie:

Ad – wzmacnienie sygnału różnicowego [V/V],

Acm – wzmacnienie wspólne sygnału [V/V].

7.4. Pomiar szumów własnych w układzie

Zbadano poziom szumów własnych układu. W tym celu zbudowano stanowisko pomiarowe składające się z generatora funkcji MXG-9802, oscyloskopu Tektronix MSO 2012 B Mixed Signal Oscilloscope, zasilacza BK Precision Triple Output Dc Power Supply 1651 oraz zewnętrznego wzmacniacza Grass Instrument CO. Model P511J. Na rysunku 7.6. przedstawiono schemat stanowiska pomiarowego do pomiaru szumów własnych układu.



Rys. 7.6. Schemat blokowy stanowiska pomiarowego do badania szumów własnych układów

Na wstępnie scharakteryzowano parametry szumowe niskoszumnego wzmacniacza Grass Instrument CO.Model P511J, którego użyto do dodatkowego wzmacnienia sygnału wyjściowego badanego wzmacniacza (tabela 7.4.).

Tab. 7.4. Wyniki pomiaru szumu własnego układu Grass Instrument CO.Model P511J

Pasmo	3Hz-300HZ
Wzmocnienie	1mln w konfiguracji 1000 (I stopień), 1000 (II stopień)
Wartość szumu RMS*	356 mV
Szum w stosunku do wejścia	356 nV

*Największa wartość otrzymana spośród trzech pomiarów.

W celu wyznaczenia szumów własnych zaprojektowanego wzmacniacza, sygnał wyjściowy (*TP4, Załącznik D. Schemat ideowy*) odbierany przy zwartych elektrodach wejściowych był dodatkowo wzmacniany (x2000 - 10 I stopień, 200 II stopień, pasmo 3 Hz – 300 Hz) z zastosowaniem wzmacniacza niskoszumnego Grass P511J i mierzony przy użyciu oscyloskopu cyfrowego (Tektronix MSO 2012 B). Szumy własne układu wyznaczano jako wartość średniokwadratową z 10 s tak zarejestrowanego sygnału podzieloną przez iloczyn wzmacnienia badanego układu (x3582.) i ustalonego wzmacnienia wzmacniacza Grass P511J (x2000). W tabeli 7.5. przedstawiono otrzymane wyniki.

Tab. 7.5. Otrzymane wyniki pomiaru szumu własnego układu w punkcie TestPoint4

Wzmocnienie całkowite*	7164000
RMS** sygnału na wyjściu	1.17 V
Szum w stosunku do wejścia	160 nV

Zapis trwał 10s

*Iloczyn wzmacnienia Grass Instrument CO. Model P511J oraz projektowanego wzmacniacza.

** Największa wartość otrzymana spośród trzech pomiarów.

Analizując otrzymane wyniki stwierdzono, iż szum własny układu w stosunku do wejścia znajduje się w założonym przedziale (<4 uV).

7.5. Test zasięgu transmisji układu

W celu sprawdzenia odległości, z jakiej można komunikować się z układem przeprowadzono test zasięgu transmisji układu. Stanowisko pomiarowe składało się z telefonu Samsung S10 Lite, aplikacji ST BLE ToolBox v 1.3.0, taśmy mierniczej TOPEX 5M Verticolour oraz układ projektowanego wzmacniacza.

Test zasięgu transmisji został przeprowadzony za pomocą aplikacji ST BLE ToolBox v 1.3.0 zainstalowanej na smartfonie Samsung S10 Lite. Używając taśmy mierniczej odmierzono odległość 5,0 m w linii prostej, a następnie oddalano się co 1 m od badanego układu, jednocześnie odświeżając aplikację w celu sprawdzenia wartość parametru RSSI. Po przekroczeniu odległości 5,0 m urządzenie BlueNRG-2 nie było już dostępne w aplikacji.

W tabeli 7.6. przedstawiono wartości RSSI w zależności od odległości od badanego urządzenia.

Tab. 7.6. Wartości RSSI w zależności od odległości

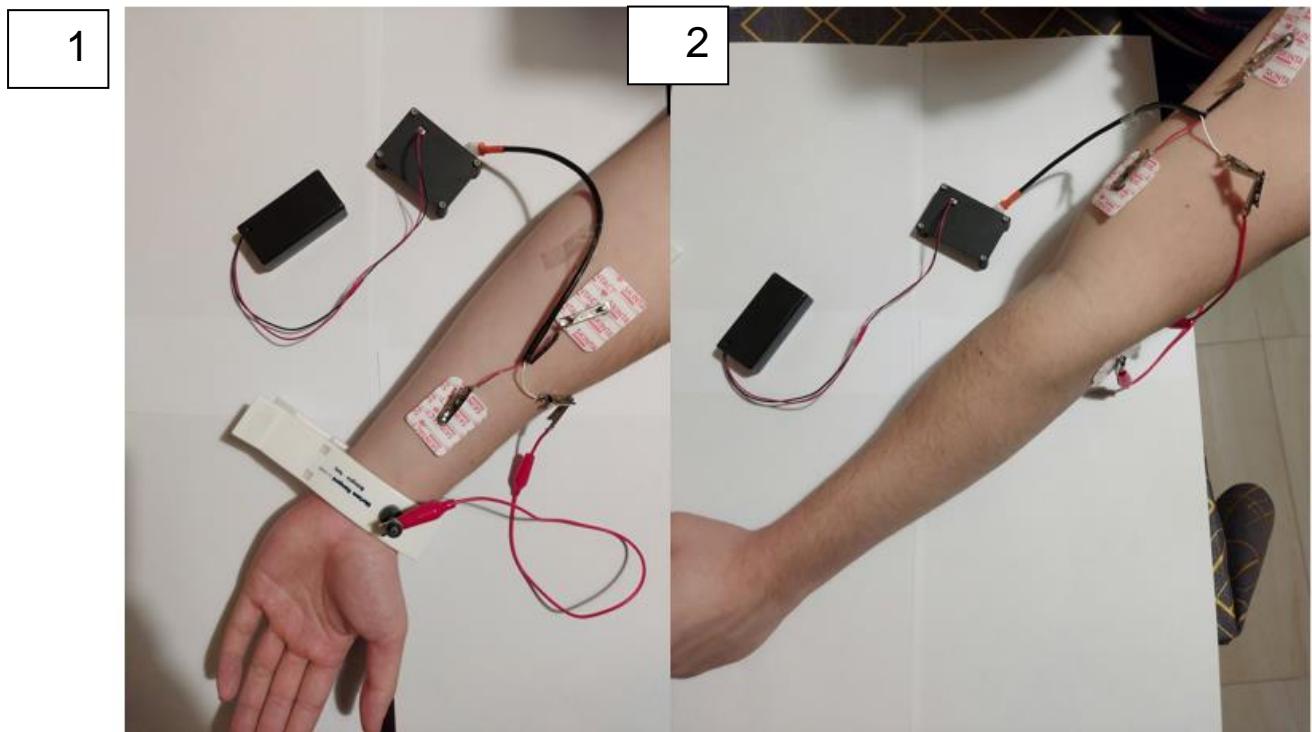
Odległość [m]	Wartość RSSI [dBm]
0.0 m	-56
1.0 m	-67
2.0 m	-78
3.0 m	-80
4.0 m	-87
5.0 m	-90

7.6. Pomiar sygnału elektromiograficznego

Rejestracja sygnału elektromiograficznego. Badania dokonano dla dwóch różnych lokalizacji elektrod odbiorczych (rysunek 7.7.)

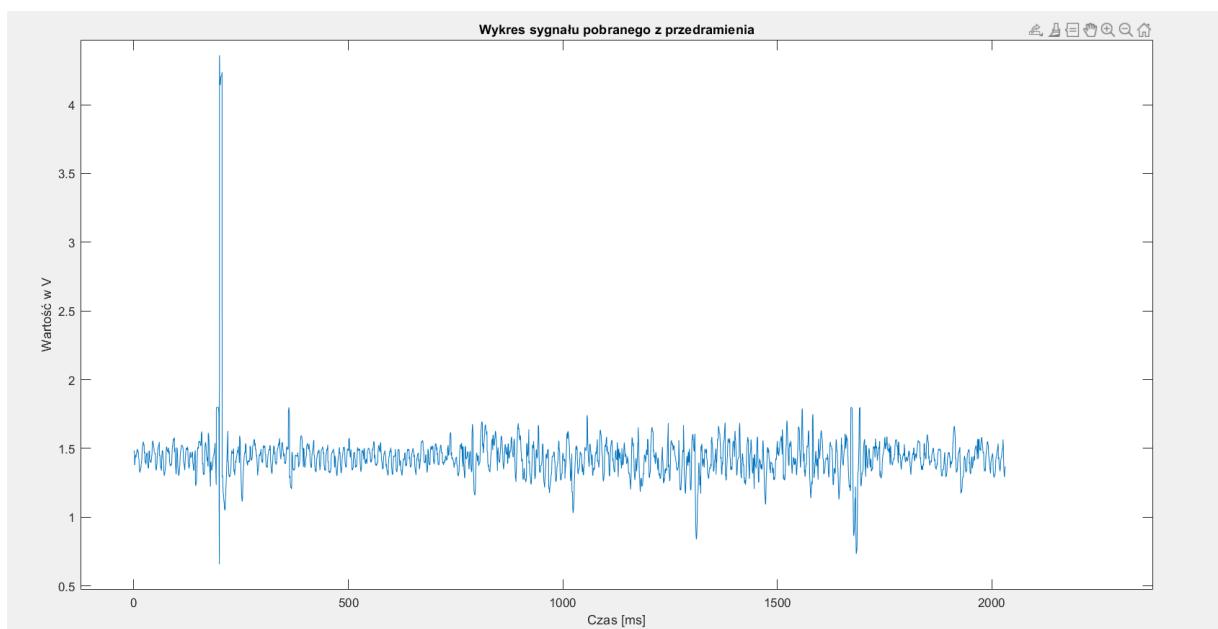
Stanowisko pomiarowe składało się z następujących elementów: aplikacja do komunikacji z urządzeniami BLE, układ projektowanego wzmacniacza, PC Lenovo Legion oraz skrypty w programie Matlab

Na rysunku 7.7. przedstawiono przykładowe umiejscowienie elektrod w zależności od badanych grup mięśniowych.

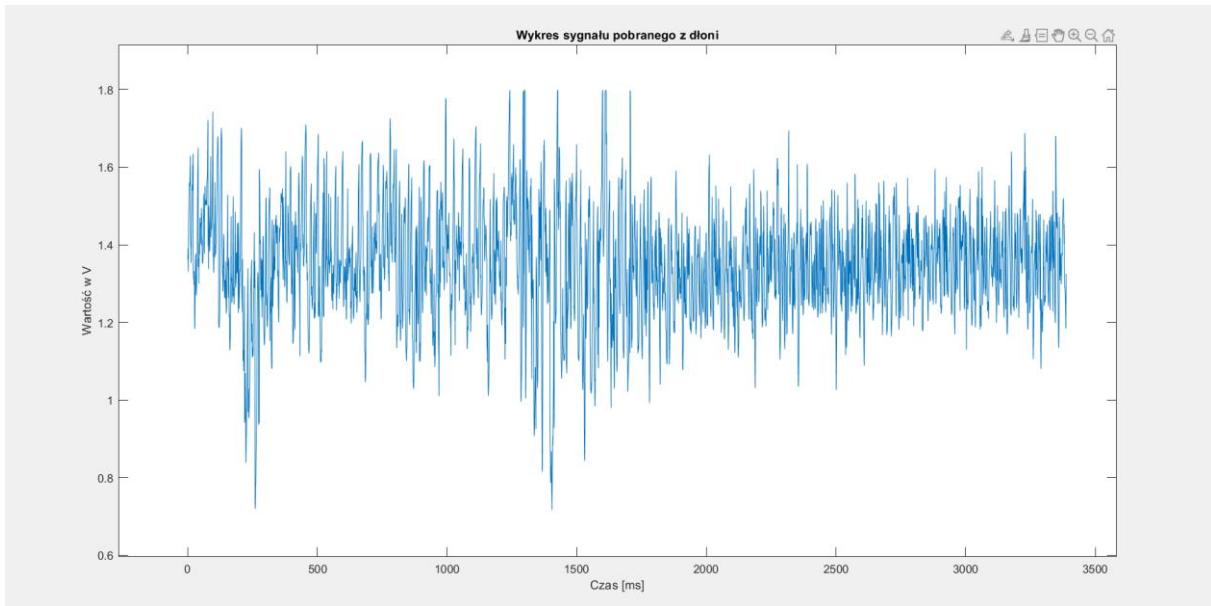


Rys. 7.7. Akwizycja sygnału EMG: 1) z grupy mięśni przedramienia; 2) z grupy mięśni bicepsa

Na rysunku 7.8. przedstawiono zarejestrowany sygnał EMG pobrany z mięśni przedramienia, natomiast na rysunku 7.9. przedstawiono zarejestrowany sygnał EMG pobrany z mięśni kłębu kciuka.



Rys. 7.8. Przebieg zarejestrowanego sygnału pobranego z mięśni przedramienia (wzmocnienie x3582)



Rys. 7.9. Przebieg zarejestrowanego sygnału poisanego z mięśni kłębku kciuka(wzmocnienie x3582)

7.7. Testy oprogramowania

7.7.1. Test poprawności zapisu danych oraz odczytu przez aplikację

Test ten przeprowadzono przy użyciu opracowanej aplikacji (*Załącznik B, notyfie.cpp*).

W celu przeprowadzenia testu poprawności przesyłania danych o określonych wartościach przez charakterystykę Bluetooth napisana została funkcja **Testowe Wartości()** należąca do oprogramowania mikrokontrolera (*Załącznik A, funkcja Testowe Wartości()*), gdzie kolejno wysyłano bufor ze znanyymi wartościami {1200, 1500, 33800, 5400, 24500, 6700, 7500, 1600, 7800, 9600}.

Następnie za pomocą stworzonej aplikacji do komunikacji z urządzeniami BLE, odczytano 10000 próbek. Fragment pliku TXT przedstawiono na rysunku 7.10.

Rys. 7.10. Fragment pliku tekstowego zapisywanego przez stworzoną aplikację

Dane zapisywane są w formacie HEX. Widoczne są w nim powtarzające się wartości, zgodne z tymi, które zostały wysłane (1200, 1500, 33800, 5400, 24500, 6700, 7500, 1600, 7800, 9600).

Otrzymany plik tekstowy wczytano do skryptu Matlab, który sprawdza czy znane wartości wpisywane w bufor są równe z danymi otrzymanym (*Załącznik C, matlab porównujący znane dane wraz z zapisanymi do pliku*).

Dane zostały poprawnie przesyłane i zapisane do pliku.

7.7.2. Test kolejności odbioru ramek notyfikacyjnych przez urządzenie

Test ten przeprowadzono przy użyciu opracowanej aplikacji (Załącznik B, zmodyfikowana aplikacja notyfie.cpp).

W celu przeprowadzenia testu poprawności odbioru ramek notyfikacyjnych przez aplikację do komunikacji z urządzeniami BLE. Stworzono funkcję APP_Tick2 () należącą do oprogramowania mikrokontrolera (Załącznik A, *Funkcja APP_Tick2()*), gdzie przesyłano bufor wypełniony zdefiniowanymi danymi. Na końcu każdego wysłanego bufora dodawano jego numer (2 ostatnie bajty).

Następnie za pomocą stworzonej aplikacji do komunikacji z urządzeniami BLE, odczytano 10000 próbek. Fragment pliku wynikowego TXT zawierającego 10000 próbek przedstawiono na rysunku 7.11.

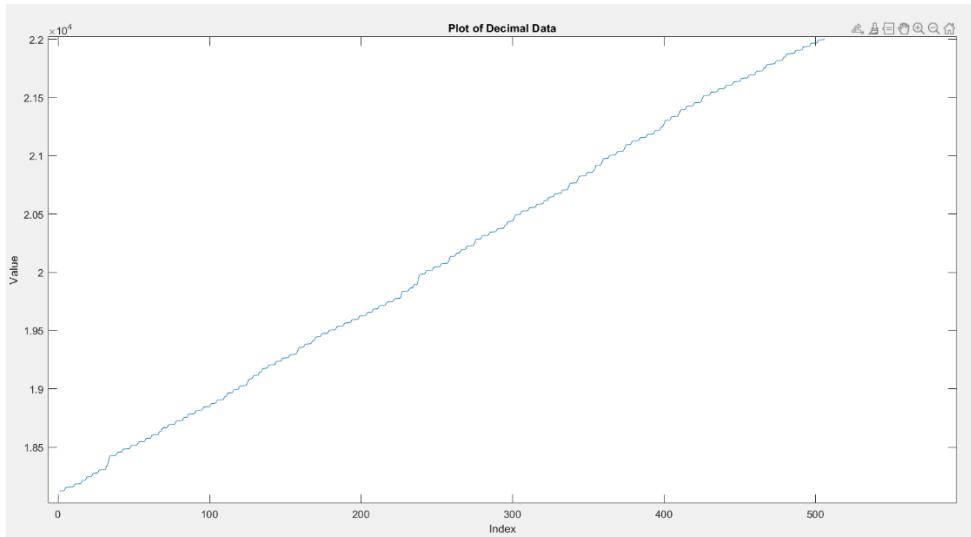
```

3a 42 3a 3c 3a 3b 3a 40 3a 44 3a 45 3a 43 3a 3b 3a 3b cc 46
3a 35 3a 7d 3a af 3a 72 3a 5e 3a 6d 3a 5e 3a 50 3a 4f cd 46
3a 35 3a 7d 3a af 3a 72 3a 5e 3a 6d 3a 5e 3a 50 3a 4f ce 46
3a 37 3a 3d 3a 40 3a 40 3a 41 3a 3d 3a 36 3a 34 3a 36 cf 46
3a 2e 3a 34 3a 36 3a 2f 3a 2e 3a 33 3a 3a 35 3a 2f ea 46
3a 40 3a 7f 3a 87 3a 65 3a 8b 3a 82 3a 6b 3a 9b 3a 9c eb 46
3a 40 3a 7f 3a 87 3a 65 3a 8b 3a 82 3a 6b 3a 9b 3a 9c ec 46
3a 83 3a 9f 3a 77 3a 5f 3a 6c 3a 63 3a 50 3a 47 3a 43 ed 46
3a 83 3a 9f 3a 77 3a 5f 3a 6c 3a 63 3a 50 3a 47 3a 43 ee 46
3a 32 3a 2e 3a 31 3a 38 3a 37 3a 33 3a 34 3a 33 3a 30 ef 46
3a 39 3a 2f 3a 2e 3a 38 3a 38 3a 2e 3a 2f 3a 36 3a 35 08 47
3a 47 3a 86 3a 94 3a 52 3a 60 3a 94 3a 7a 3a 6b 3a 96 09 47
3a 47 3a 86 3a 94 3a 52 3a 60 3a 94 3a 7a 3a 6b 3a 96 0a 47
3a 58 3a 5a 3a 56 3a 4c 3a 47 3a 42 3a 39 3a 35 3a 36 0b 47
3a 58 3a 5a 3a 56 3a 4c 3a 47 3a 42 3a 39 3a 35 3a 36 0c 47
3a 39 3a 2f 3a 2d 3a 34 3a 35 3a 37 3a 3a 3a 37 3a 39 26 47
3a 3c 3a 73 3a 78 3a 47 3a 75 3a ab 3a 77 3a 6a 3a a4 27 47
3a 3c 3a 73 3a 78 3a 47 3a 75 3a ab 3a 77 3a 6a 3a a4 28 47
3a 36 3a 80 3a 89 3a 43 3a 60 3a 99 3a 67 3a 52 3a 91 45 47
3a 36 3a 80 3a 89 3a 43 3a 60 3a 99 3a 67 3a 52 3a 91 46 47
3a 53 3a 53 3a 4d 3a 41 3a 38 3a 38 3a 3a 3a 3a 3a 47 47
3a 53 3a 53 3a 4d 3a 41 3a 38 3a 38 3a 3a 3a 3a 3a 48 47
3a 32 3a 6f 3a 80 3a 49 3a 5a 85 3a 64 3a 5a 3a 8c 63 47
3a 32 3a 6f 3a 80 3a 49 3a 5a 85 3a 64 3a 5a 3a 8c 64 47
3a 57 3a 55 3a 4b 3a 40 3a 3b 3a 38 3a 38 3a 37 3a 33 65 47
3a 57 3a 55 3a 4b 3a 40 3a 3b 3a 38 3a 38 3a 37 3a 33 66 47
3a 32 3a 35 3a 34 3a 34 3a 34 3a 35 3a 3d 3a 41 3a 3f 80 47
3a 31 3a 75 3a 7f 3a 4e 3a 7c 3a b4 3a 90 3a 88 3a bc 81 47
3a 31 3a 75 3a 7f 3a 4e 3a 7c 3a b4 3a 90 3a 88 3a bc 82 47
3a 5a 3a 4f 3a 40 3a 3f 3a 3d 3a 38 3a 3a 38 3a 31 83 47
3a 5a 3a 4f 3a 40 3a 3f 3a 3d 3a 38 3a 3a 38 3a 31 84 47
3a 34 3a 35 3a 34 3a 32 3a 2e 3a 30 3a 30 3a 30 3a 30 9e 47
3a 2f 3a 74 3a 82 3a 4d 3a 74 3a a6 3a 7b 3a 71 3a ba bd 47
3a 3a 3c 3a 3a 3a 36 3a 36 3a 36 3a 3c 3a 42 f8 47
3a 3f 3a 78 3a 86 3a 57 3a 81 3a ba 3a 8a 3a 7d 3a c0 f9 47
3a 3f 3a 78 3a 86 3a 57 3a 81 3a ba 3a 8a 3a 7d 3a c0 fa 47
3a 60 3a 56 3a 4d 3a 46 3a 3e 3a 38 3a 36 3a 35 3a 35 fb 47
3a 60 3a 56 3a 4d 3a 46 3a 3e 3a 38 3a 36 3a 35 3a 35 fc 47
3a 38 3a 7c 3a 85 3a 50 3a 77 3a a6 3a 84 3a 87 3a bb 17 48
3a 38 3a 7c 3a 85 3a 50 3a 77 3a a6 3a 84 3a 87 3a bb 18 48
3a 69 3a 5e 3a 4d 3a 45 3a 41 3a 36 3a 31 3a 34 3a 36 19 48
3a 69 3a 5e 3a 4d 3a 45 3a 41 3a 36 3a 31 3a 34 3a 36 1a 48
3a 32 3a 36 3a 3a 3a 43 3a 45 3a 3d 3a 39 3a 3c 3a 3c 34 48
3a 36 3a 6f 3a 7b 3a 43 3a 6a 3a 9a 3a 65 3a 63 3a a7 35 48
3a 36 3a 6f 3a 7b 3a 43 3a 6a 3a 9a 3a 65 3a 63 3a a7 36 48
3a 50 3a 45 3a 41 3a 3d 3a 39 3a 3c 3a 40 3a 3e 3a 3d 37 48

```

Rys. 7.11. Fragment pliku wynikowego TXT

Otrzymany plik tekstowy wyczytano do skryptu Matlab, który sprawdza czy znane wartości wpisywane w buforze są równe z danymi otrzymanym (*Załącznik C, skrypt matlab zliczający otrzymane ramki charakterystyki*). Na rysunku 7.12. przedstawiono wykres kolejności otrzymanych danych.



Rys. 7.12. Wykres przedstawiający kolejność ramek danych otrzymywanych w aplikacji. Oś Y przedstawia numer otrzymanej próbki, natomiast oś X przedstawia kolejność jej otrzymania

Zaznaczyć należy, iż wszystkie ramki, które zostały przesłane były poprawne. Natomiast stworzona aplikacja cyklicznie nie nadąża z wysłaniem żądania danych notyfikacji na czas, stąd część wartości jest tracona. Zmiana wartości mocy transmisji oraz ułożenie układu w bliższej odległości od urządzenia umożliwiało otrzymanie większej liczby ramek zgodnie z kolejnością.

7.7.3. Test poprawności przesyłu oraz zapisu danych do pliku

Test ten przeprowadzono przy użyciu opracowanej aplikacji (Załącznik B, aplikacja notyfie_multiple.cpp).

W celu sprawdzenia poprawności działania aplikacji w pętli main oprogramowania mikrokontrolera zastosowano stworzoną funkcję void APP_Tick2() należącą do oprogramowania mikrokontrolera (*Załącznik A, Funkcja APP_Tick2()*), która aktualizuje charakterystykę danymi stworzonego bufora, posiadającego dziewięć znanych wartości typu uint16_t oraz dwa znaki znaczace 00 typu uint8_t.

Następnie program wgrano do mikrokontrolerów BlueNRG obu posiadanych układów (układ projektowanego wzmacniacza, układ deweloperski).

Uruchomiono aplikację, połączono się prawidłowo z oboma urządzeniami oraz dokonano subskrypcji charakterystyki.

Otrzymano dwa pliki tekstowe, posiadające po 10000 próbek, które wczytano do skryptów Matlab (*Załącznik C, skrypty Matlab sprawdz_zera.m , sprawdz_dane.m*).

Analizując otrzymane wyniki można stwierdzić, iż wszystkie odebrane ramki danych zostały odebrane oraz zostały poprawnie zapisane do pliku.

Podsumowanie

Rezultatem wymiernym pracy jest prototyp urządzenia służący do pomiaru sygnału elektromiograficznego z różnych partii mięśni. Przeprowadzony dobór komponentów, poprzedzony analizą ich funkcjonowania, umożliwił opracowanie konstrukcji, która pozwala zarejestrować sygnał EMG. Jednakże w procesie konstruowania urządzenia pojawiły się trudności, które zostały poniżej scharakteryzowane.

Zrealizowany protokół komunikacji opiera się o Bluetooth, natomiast wyniki przedstawione w rozdziale 7 dotyczące oprogramowania pokazują, iż napisana aplikacja ma problem z rejestracją wszystkich ramek notyfikacyjnych wysyłanych przez mikrokontroler. Próbowano zoptymalizować działanie aplikacji poprzez usunięcie fragmentów kodu nie związanych bezpośrednio z odbiorem notyfikacji, zmianę zaimplementowanych funkcji w bibliotece SimpleBLE oraz przesunięciu zadania zapisu do pliku po zakończeniu subskrypcji charakterystyki. Dokonano również kilkukrotnej modyfikacji długości charakterystyki oraz zmian jej typu (z notyfikacyjnej na odczytywalną). Mimo tych modyfikacji nadalauważono utratę ramek wysłanych przez BlueNRG-2.

Zarejestrowany sygnał EMG również nie odzwierciedla spodziewanych przykładowych wyników.

Nie osiągnięto również wzmacnienia z zakładaną tolerancją. W pierwszym stopniu wzmacnienia należałoby wybrać rezistor $4.53\text{k}\Omega$ [E96 tol.1%] dałoby to błąd względny wynoszący ok.0.33%

Nie udało się również zrealizować regulacji poziomu wzmacnienia w II stopniu wzmacnienia. Przyczyną tego był brak komunikacji z potencjometrem MCP40D18T. Zastosowanie napisanych funkcji do obsługi komend „Read/Write” nie spowodowało zmiany wartości pozycji suwaka oraz nie umożliwiło odebrania informacji o jego pozycji. Stąd potencjometr pracował na domyślnym ustawieniu.

Otrzymany wynik parametru CMRR jest mniejszy od zakładanego. Może być to związane z brakiem oddzielnego niskoszumnego bufora na referencje wzmacniacza pomiarowego. Ponadto ze względu na błąd w realizacji projektu PCB wzmacniacz został zastąpiony układem AD8542ARZ o słabszych parametrach prądowych oraz niższym współczynnikiem CMRR(48dB). W drugiej wersji urządzenia do realizacji obu układów wykorzystano pojedyncze wzmacniacze operacyjne MCP6002.

Wykonano układ PCB, lecz jego rozmiary łącznie z obudową nie są na tyle kompaktowe aby umożliwić akwizycję sygnału EMG z mniejszych powierzchniowo mięśni np. twarzy. Ponadto w procesie tworzenia prototypu dokonano pomyłek takich jak wstawienie nieodpowiednich footprintów, niepodciagnięcie ścieżek SCL oraz SDA do stanu wysokiego. W drugiej wersji układu dokonano korekcji wymienionych błędów oraz zredukowano wymiary obwodu PCB (rysunek 6.2.). W pierwszej iteracji urządzenia dobrano również komponenty nieoptymalnie pod względem kosztowym co skutkowało dużym budżetem wydanym na realizację fizyczną, jak na przedstawioną charakterystykę urządzenia w rozdziale 2. W drugim stopniu wzmacnienia użyto stosunkowo drogi, jak na ten blok wzmacniacz operacyjny AD8538ARZ. W trakcie realizacji kolejnej wersji układu zwrócono uwagę na dobór komponentów o parametrach podobnych do tych wykorzystanych w prototype, jednocześnie zastępując je tańszymi układami np. wzmacniacz operacyjny AD8538ARZ kosztujący 14.99zł zastąpiono wzmacniaczem operacyjnym MCP6002 za którego należy zapłacić mniej niż 3 zł. Dzięki temu znacząco zredukowano budżet przeznaczony na realizację urządzenia.

Zademonstrowano możliwość działania układu w sieci Mesh, jednakże ze względu na brak drugiego urządzenia, na podstawie którego można byłoby wykonać kolejny węzeł nie

zaprezentowano ograniczeń, jak również możliwości BlueNRGMESH. W drugiej wersji systemu do akwizycji EMG planowane jest wykonanie co najmniej trzech wzmacniaczy, dzięki czemu będzie można stworzyć w pełni funkcjonalną sieć czujników.

Maksymalny zasięg transmisji urządzenia wynosił 5,0 m. Jest on stosunkowo mały w porównaniu z możliwościami oferowanymi przez BLE (<100m). Prawdopodobnie wynikło to z doboru nieoptymalnych pod względem mocy transmisji komponentów do realizacji RF-Matching Network. W drugiej wersji prototypu wykorzystano układ BALFNRG, który powinien znacznie zwiększyć parametry transmisji (Tx, RSSI, zasięg).

Układ zasilany jest baterijnym źródłem zasilania, teoretyczny czas pracy wynosi ponad 6 h, niemniej jednak nie dokonano pomiaru czasu pracy układu.

Dalsze prace nad rozwojem wzmacniacza mogą dotyczyć zmiany obudowy urządzenia, polegającej na dodanie uchwytu na baterię.

Bibliografia

Publikacje zostały umieszczone zgodnie z kolejnością występowania.

1. How to create the best BOM for your pcb (2023) , <https://www.proto-electronics.com/blog/how-to-create-best-bom-for-pcb>
2. What is a BOM in PCB Designe (2020) <https://resources.altium.com/p/what-is-a-bom-in-pcb-design>
3. BLUETOOTH SPECIFICATION Version 4.0 (2010) [pdf], <https://www.bluetooth.com/specifications/specs/core-specification-4-0/>, dostęp na dzień 11.12.2022 r.
4. Bluetooth® Wireless Technology Owwerwiew [online], <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>, dostęp na dzień 11.12.2022 r.
6. What are binary and text file [online], <https://www.nayuki.io/page/what-are-binary-and-text-files> , dostęp na dzień 11.12.2022 r.
7. Daga A., Hex File Format [online], <https://www.engineersgarage.com/hex-file-format/> , dostęp na dzień 11.12.2022r.
8. BlueNRG-1 ST-LINK utility for BlueNRG-1, BlueNRG-2 MCU [pdf], <https://www.st.com/en/embedded-software/stsw-bnrg1stlink.html> , dostęp na dzień 11.12.2022r.
9. STSW-WISE-STUDIO Data brief [pdf], https://www.st.com/resource/en/data_brief/stsw-wise-studio.pdf , dostęp na dzień 11.12.2022r.
10. UM1075 User manual [pdf], https://www.st.com/resource/en/user_manual/um1075-stlinkv2-incircuit-debuggerprogrammer-for-stm8-and-stm32-stmicroelectronics.pdf , dostęp na dzień 11.12.2022r.
10. Understanding the I 2C Bus, Application Report SLVA704 (2015) [pdf], https://www.ti.com/lit/an/slva704/slva704.pdf?ts=1678544284220&ref_url=https%253A%252F%252Fwww.google.co.za%252F , dostęp na dzień 11.12.2022r.
11. Understanding RSSI [online], <https://www.metageek.com/training/resources/understanding-rssi/>, dostęp na dzień 11.12.2022r.
12. St BlueNRG-MESH APPLICATION OVERWIEV [PDF/Doxygen] <https://www.st.com/en/embedded-software/stsw-bnrg-mesh.html>; dostępny na dzień 11.12.2022r.
13. STMicroelectronics, Rozwiązania STMicroelectronics do projektowania sieci Bluetooth mesh <https://mikrokontroler.pl/2018/07/06/webinarium-rozwiazania-stmicroelectronics-do-projektowania-sieci-bluetooth-mesh/>, dostępny na dzień 11.12.2022r.
14. Bonina S., Radakichenane H. (2019), [pdf], BlueNRG-MESH Overview of the BlueNRG-MESH SDK for the ST Bluetooth Low Energy SOCs, https://www.st.com/content/dam/AME/2019/technology-tour-2019/anaheim/presentations/T4S3_Anaheim_BlueNRG-Mesh_H.Radakichenane-S.Bonina.pdf, dostępny na dzień 11.12.2022r.
15. KONDRAD P. (2006) [pdf], <https://www.noraxon.com/wp-content/uploads/2014/12/ABC-EMG-ISBN.pdf> , dostępny na dzień 11.12.2022r.

- 16 Cromwell L., Weibell F.J., Pfeiffer E. A. (1980), Biomedical Instrumentation and Measurements
- 17 Webster J.G.(red) (2009), MEDICAL INSTRUMENTATION Application and Design, fourth edition, John Wiley & Sons, INC. Hoboken, NJ
- 18 Merletti R, Farina D.(red) (2016) Surface Electromyography Physiology,Engineering and Applications, IEE PRESS Piscataway, NJ
- 19 Analog Devices, A Designer's Guide to Instrumentation Amplifiers, 3rd Edition <https://www.analog.com/media/en/training-seminars/design-handbooks/designers-guide-instrument-amplifiers-complete.pdf> , dostęp na dzień 11.12.2022r.
- 20 Analog Devices, A620 Data Sheet <https://www.analog.com/media/en/technical-documentation/data-sheets/adp7118.pdf>, dostęp na dzień 11.12.2022r.
- 21 Microchip MCP40D17/18/19 Data Sheet <https://www.mouser.pl/datasheet/2/268/22152b-71278.pdf> , dostęp na dzień 11.12.2022r.
- 22 Holistic character Designe <https://design.tutsplus.com/articles/holistic-character-design--vector-3538> , dostęp na dzień 11.12.2022r.

Spis rysunków

Rys. 1.1. Programator ST-Link V2.....	9
Rys. 2.1. Schemat projektowanego systemu do akwizycji sygnału EMG	13
Rys. 3.1. Schemat bloku przedwzmacniacza	22
Rys. 3.2. Schemat filtra górnoprzepustowego	24
Rys. 3.3. Schemat filtra dolnoprzepustowego	24
Rys. 3.4. Schemat zaprojektowanego bloku II stopnia wzmacnienia	28
Rys. 3.5. Fragment schematu zaprojektowanego bloku mikrokontrolera BlueNRG-2.....	32
Rys. 3.6. Schemat bloku mikrokontrolera	33
Rys. 3.7. Schemat zaprojektowanego bloku VGND	36
Rys. 3.8. Schemat zaprojektowanego bloku regulatora napięcia zasilania	37
Rys. 4.1. Schemat symulacyjny bloku zasilnia wykonany w programie LT spice	38
Rys. 4.2. Wyniki symulacji przejściowej w punkcie Vout (wyjście z układu LDO).....	38
Rys. 4.3. Wyniki symulacji przejściowej w punkcie Vref (wyjście z układu VGND).....	39
Rys. 4.4. Schemat symulacyjny bloku przedwzmacniacza	38
Rys. 4.5. Wyniki symulacji przejściowej w punkcie TP1 (wyjście z wzmacniacza instrumentalnego).....	40
Rys. 4.6. Sygnał wejściowy, używany w symulacji w punkcie sin	40
Rys. 4.7. Charakterystyka amplitudowo-częstotliwościowa na wyjściu filtru górnoprzepustowego (TP2)	41
Rys. 4.8. Charakterystyka amplitudowo-częstotliwościowa na wyjściu filtru dolnoprzepustowego (TP4)	41
Rys. 4.9. Schemat symulacyjny toru pomiarowego projektowanego wzmacniacza	42
Rys. 4.10. Sygnał wejściowy V2. Jego amplituda wynosi 20 μ V.....	42
Rys. 4.11. Sygnał w punkcie TP2	43
Rys. 4.12. Wykres amplitudy sygnału na wyjściu II stopnia wzmacnienia (TP3)	43
Rys. 4.13. Wykres amplitudy sygnału na wyjściu filtra dolnoprzepustowego (TP4)	44
Rys. 4.14. Wykres amplitudy sygnału na wyjściu II stopnia wzmacnienia (TP3)	44
Rys. 4.15. Wykres amplitudy sygnału na filtru dolnoprzepustowym (TP4).....	45
Rys. 4.16. Schemat symulacyjny pomiaru współczynnika CMRR dla toru pomiarowego projektowanego wzmacniacza.....	45
Rys. 4.17. Sygnał wejściowy V2 używany w symulacji	46
Rys. 4.18. Wyniki symulacji przejściowej w punkcie TP4.....	46
Rys. 5.1. Struktura ramki danych charakterystyki notyfikacyjnej.....	47
Rys. 5.2. Diagram protokołu komunikacji między aplikacją a urządzeniem	48
Rys. 5.3. Widok okna aplikacji.....	48
Rys. 5.4. Schemat przepływu danych w aplikacji komputerowej	49
Rys. 5.5. Widok z menu głównego aplikacji o rozszerzonej funkcjonalności.....	50
Rys. 5.6. Schemat przepływu aplikacji o rozszerzonej funkcjonalności	51
Rys. 5.7. Ekran główny aplikacji ST-BLE-MESH przedstawiający gotowy dołączenia w sieć MESH mikrokontroler BlueNRG-2 znajdujący się w projektowanym układzie	53
Rys. 5.8. Widok z aplikacji konsolowej, przedstawiający udane połączenie się z dwoma mikrokontrolerami BlueNRG oraz subskrypcję jednej z charakterystyk.....	54
Rys. 5.9. Schemat przepływu danych w aplikacji służącej do połączenia się z dwoma urządzeniami BlueNRG	53
Rys. 5.10. Schemat ramki komendy „Write”	56
Rys. 5.11. Schemat ramki komendy „Read”	56

Rys. 6.1. Pierwotna wersja prototypu wzmacniacza	57
Rys. 6.2. Finalna wersja prototypu wzmacniacza	58
Rys. 6.3. Prototyp układu	59
Rys. 6.4. Porównanie I wersji prototypu urządzenia oraz II wersji obwodu PCB	60
Rys. 6.5. Przedstawienie elementów obudowy	62
Rys. 6.6. Widok złożenia projektu obudowy urządzenia	63
Rys. 6.7. Widok części składowych obudowy w programie PrusaSlicer	63
Rys. 6.8. Rozłożone części obudowy z zamontowaną płytą obwodu drukowanego	64
Rys. 6.9. Złożona obudowa wraz z zamontowanym układem	64
Rys. 7.1. Schemat blokowy stanowiska pomiarowego	66
Rys. 7.2. Charakterystyka amplitudowo-częstotliwościowa filtra górnoprzepustowego	67
Rys. 7.3. Schemat blokowy stanowiska pomiarowego	68
Rys. 7.4. Charakterystyka amplitudowo-częstościowa na wyjściu filtra dolnoprzepustowego (wyjście z toru pomiarowego wzmacniacza)	69
Rys. 7.5. Schemat blokowy stanowiska pomiarowego do badania współczynnika CMRR w układzie	70
Rys. 7.6. Schemat blokowy stanowiska pomiarowego do badania szumów własnych układów	71
Rys. 7.7. Akwizycja sygnału EMG	73
Rys. 7.8. Przebieg zarejestrowanego sygnału EMG pobranego z mięśni przedramienia	73
Rys. 7.9. Przebieg zarejestrowanego sygnału EMG pobranego z dłoni	74
Rys. 7.10. Fragment pliku tekstowego zapisywanego przez stworzoną aplikację	75
Rys. 7.11. Fragment pliku wynikowego TXT	76
Rys. 7.12. Wykres przedstawiający kolejność ramek danych otrzymywanych w aplikacji	76

Spis tabel

Tab. 2.1. Porównanie podejść do procesu projektowego	13
Tab. 2.2. Porównanie dwóch rodzajów elektrod rozważanych do zastosowania w powierzchniowej akwizycji sygnału EMG w projektowanym wzmacniaczu.....	14
Tab. 2.3. Porównanie dwóch standardów protokołu Bluetooth.....	15
Tab. 2.4. Porównanie rozwiązań w zakresie przesyłu sygnału.....	15
Tab. 2.5. Porównanie dwóch możliwych realizacji cyfrowej części układu	16
Tab. 2.6. Porównanie głównych cech filtrów pasywnych i aktywnych	17
Tab. 2.7. Porównanie sposobu otrzymania napięcia symetrycznego w układzie wzmacniacza	17
Tab. 3.1. Proponowany podział budżetu	19
Tab. 3.2. Porównanie konstrukcji wzmacniacza instrumentalnego	20
Tab. 3.3. Charakterystyka porównawcza wzmacniaczy instrumentalnych	20
Tab. 3.4. Oznaczenie wartości komponentów bloku przedwzmacniacza	22
Tab. 3.4. Porównanie dwóch możliwych rozwiązań konfiguracji stopni filtracji	20
Tab. 3.5. Zestawienie wartości komponentów użytych do realizacji bloku filtracji	24
Tab. 3.6. Porównanie konfiguracji odwracającej i nieodwracającej wzmacniacza operacyjnego	25
Tab. 3.7. Zestawienie danych wzmacniaczy operacyjnych	25
Tab. 3.8. Porównanie cyfrowych sposobów regulacji wzmacnienia	26
Tab. 3.9. Tabela porównawcza potencjometrów cyfrowych.....	27
Tab. 3.10. Parametry potencjometru MCP40D18T-104E/LT	27
Tab. 3.11. Spis elementów wchodzących w skład bloku II stopnia wzmacnienia	28
Tab. 3.12. Zestawienie wybranych parametrów mikrokontrolerów rozważanych do zastosowania w projektowanym urządzeniu.....	30
Tab. 3.13. Porównanie rozwiązań dotyczących podkładu RF-Matching Network.....	31
Tab. 3.14. Spis komponentów wchodzących w skład bloku mikrokontrolera.....	32
Tab. 3.15. Porównanie regulatorów linowych do wykorzystania w bloku zasilania	34
Tab. 3.16. Zestawienie wzmacniaczy operacyjnych do użycia w układzie VGND	34
Tab. 3.17. Sumaryczny pobór prądu przez poszczególne elementy	35
Tab. 3.18. Porównanie źródeł zasilania do wykorzystania w bloku zasilania	35
Tab. 3.19. Spis elementów wchodzących w skład bloku zasilania	36
Tab. 6.1. Całkowite podsumowanie kosztu realizacji fizycznej	61
Tab. 7.1. Wykaz napięcia zasilania na poszczególnych komponentach aktywnych urządzenia	66
Tab. 7.2. Wartość amplitudy sygnału w zależności od jego częstotliwości w punkcie Test Point 3 (wyjście z filtra górnoprzepustowego).....	67
Tab. 7.3. Wartość amplitudy sygnału w zależności od jego częstotliwości w punkcie Test Point 2 (wyjście z filtra dolnoprzepustowego).....	69
Tab. 7.4. Wyniki pomiaru szumu własnego układu Grass Instrument CO.Model P511J.....	71
Tab. 7.5. Otrzymane wyniki pomiaru szumu własnego układu w punkcie TestPoint4	71
Tab. 7.6. Wartości RSSI w zależności od odległości.....	72

Spis załączników

Załącznik A. Oprogramowanie mikrokontrolera	86
Załącznik B:Aplikacje PC.....	92
Załącznik C:Skrypty stworzone w programie Matlab.....	103
Załącznik D. Schematy ideowy.....	107
Załącznik E. Projekt obudowy płytki.....	108
Załącznik F. Projekt obwodu drukowanego PCB	111

Załącznik A. Oprogramowanie mikrokontrolera

1

2.Funkcja DMA_Configuration()

```
*****
* Nazwa funkcji : DMA_Configuration.
* Opis      : Funkcja konfiguruająca DMA.
*****  
/  
void DMA_Configuration(void)  
{  
    DMA_InitType DMA_InitStructure;  
  
    SysCtrl_PeripheralClockCmd(CLOCK_PERIPH_DMA, ENABLE);  
  
    /* DMA_CH_UART_TX Initialization */  
    DMA_InitStructure.DMA_PeripheralBaseAddr = ADC_OUT_ADDRESS;  
    DMA_InitStructure.DMA_MemoryBaseAddr = (uint32_t)buffer_adc;  
    DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralSRC;  
    DMA_InitStructure.DMA_BufferSize = (uint32_t)LEN;  
    DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;  
    DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;  
    DMA_InitStructure.DMA_PeripheralDataSize = DMA_PeripheralDataSize_HalfWord;  
    DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_HalfWord;  
    DMA_InitStructure.DMA_Mode = DMA_Mode_Circular; //DMA_Mode_Circular;  
    DMA_InitStructure.DMA_Priority = DMA_Priority_High;  
    DMA_InitStructure.DMA_M2M = DMA_M2M_Disable;  
    DMA_Init(ADC_DMA_CH0, &DMA_InitStructure);  
  
    /* Enable DMA ADC CHANNEL 0 Transfer Complete interrupt */  
    DMA_FlagConfig(ADC_DMA_CH0, DMA_FLAG_TC, ENABLE);  
    DMA_FlagConfig(ADC_DMA_CH0, DMA_FLAG_HT, ENABLE);  
  
    /* Select DMA ADC CHANNEL 0 */  
    DMA_SelectAdcChannel(DMA_ADC_CHANNEL0, ENABLE);  
  
    /* Enable DMA ADC CHANNEL 0 */  
    DMA_Cmd(ADC_DMA_CH0, ENABLE);  
}
```

1.Funkcja ADC_Update1()

```
*****  
* Nazwa funkcji: ADC_Update1  
* Opis: Funkcja aktualizująca wartość charakterystyki zawierającej dane z przetwornika  
ADC  
* Pobiera: adc_send_buff - bufor do wysłania, len - długość bufora  
* Zwraca: Status.  
*****  
tBleStatus ADC_Update1 (uint8_t * adc_send_buff, uint8_t len)  
{  
  
    tBleStatus ret;
```

```

    // Wywołanie funkcji aci_gatt_update_char_value_ext i aktualizacja wartości
    charakterystyki
    ret =
        aci_gatt_update_char_value_ext (connection_handle, envSensServHandle,
                                         adcCharHandle, 1, 40, 0,
                                         len, (uint8_t *) adc_send_buff);

    // Sprawdź, czy aktualizacja charakterystyki zakończyła się sukcesem
    if (ret != BLE_STATUS_SUCCESS)
    {

        // Jeśli wystąpił błąd zwróć jego wartość
        PRINTF ("Error while updating Acceleration characteristic: 0x%02X\n",
                ret);

        // Zwraca kod błędu
        return BLE_STATUS_ERROR;
    }

    // Jeśli aktualizacja zakończyła się sukcesem zwróć odpowiedni status
    return BLE_STATUS_SUCCESS;
}

```

2.funkcja APP_Tick1

```
*****

```

* Nazwa funkcji: APP_Tick1

* Opis: Wywoływana cyklicznie funkcja, przetwarzająca dane otrzymywane z przetwornika ADC techniką DMA, oraz zapisująca wartość „00” w dwóch ostatnich pozycjach wysyłanego przez charakterystykę bufora

*Pobiera: Brak

* Zwraca: Brak

```
*****

```

void APP_Tick1(void) {

// Sprawdź, czy flaga półtransferu DMA jest ustawiona

if (DMA_GetFlagStatus(DMA_FLAG_HT)) {

// Przetwórz pierwszą połowę bufora buffer_adc

for (uint8_t i = 0; i < LEN / 2; i++) {

// Umieść starszy bajt wartości ADC w buforze buffer_adc_send

buffer_adc_send[2 * i] = (buffer_adc[i] >> 8) & 0xFF;

// Umieść młodszy bajt wartości ADC w buforze buffer_adc_send

buffer_adc_send[2 * i + 1] = buffer_adc[i] & 0xFF;

}

// Zapisz wartość 0x00 w dwóch ostatnich pozycjach bufora buffer_adc_send

buffer_adc_send[LEN + 1] = 0x00;

buffer_adc_send[LEN + 2] = 0x00;

```

// Wyczysć flagę półtransferu DMA
DMA_ClearFlag(DMA_FLAG_HT);
// Zaktualizuj dane ADC
ADC_Update1((uint8_t *)buffer_adc_send, LEN1);
}

// Sprawdź, czy flaga zakończenia transferu DMA jest ustawiona
if (DMA_GetFlagStatus(DMA_FLAG_TC)) {
    // Przetwórz drugą połowę bufora buffer_adc
    for (uint8_t i = LEN / 2; i < LEN; i++) {
        // Umieść starszy bajt wartości ADC w buforze buffer_adc_send1
        buffer_adc_send1[2 * (i - LEN / 2)] = (buffer_adc[i] >> 8) & 0xFF;
        // Umieść młodszy bajt wartości ADC w buforze buffer_adc_send1
        buffer_adc_send1[2 * (i - LEN / 2) + 1] = buffer_adc[i] & 0xFF;
    }
}

// Zapisz wartość 0x00 w dwóch ostatnich pozycjach bufora buffer_adc_send1
buffer_adc_send1[LEN + 1] = 0x00;
buffer_adc_send1[LEN + 2] = 0x00;

// Wyczysć flagę zakończenia transferu DMA
DMA_ClearFlag(DMA_FLAG_TC);
// Zaktualizuj dane ADC
ADC_Update1((uint8_t *)buffer_adc_send1, LEN1);
}

```

3.Funkcja Testowe_Wartosci()

```
*****
* Nazwa funkcji: Testowe_Wartosci
* Opis: Wywoływana cyklicznie funkcja, uzupełniająca wysyłany bufor znanimi wartościami.
* Pobiera: brak
* Zwraca: brak
*****
```

```
void Testowe_Wartosci(void){
for(uint8_t i=0;i< 10 ;i++){
    buffer_adc_send1[2*i]=buffer_test[i]>>8;
    buffer_adc_send1[2*i+1]=buffer_test[i]&0xFF;}
    ADC_Update1((uint8_t*)buffer_adc_send1,LEN);
}
```

3.Funkcja APP_TICK2()

```
*****
* Nazwa funkcji: APP_Tick2
* Opis: Wywoywana cyklicznie funkcja, uzupełniająca wysyłany bufor znanimi wartościami
oraz wartością iteracji wywołania
* Pobiera: brak
```

```

* Zwraca:brak
*****
void APP_Tick2(void) {
    if (DMA_GetFlagStatus(DMA_FLAG_HT)) {
        // Przetwarzaj pierwsza połowe bufora
        for (uint8_t i = 0; i < LEN2-2; i++) {
            buffer_adc_send[2 * i] = (buffer_test[i] >> 8) & 0xFF;
            buffer_adc_send[2 * i + 1] = buffer_test[i] & 0xFF;
        }

        // Zwięksź wartość iteration_count i zapisz ją w dwóch ostatnich pozycjach
        iteration_count++;
        buffer_adc_send[LEN1 - 1] = (iteration_count >> 8) & 0xFF;
        buffer_adc_send[LEN1 - 2] = iteration_count & 0xFF;

        // Wyczysć flagę półtransferu DMA
        DMA_ClearFlag(DMA_FLAG_HT);
        // Zaktualizuj dane ADC
        ADC_Update1((uint8_t *)buffer_adc_send, LEN1);
    }

    // Sprawdź, czy ustawiona jest flaga zakończenia transferu DMA
    if (DMA_GetFlagStatus(DMA_FLAG_TC)) {
        // Przetwarzaj druga połowe bufora
        for (uint8_t i = LEN / 2; i < LEN; i++) {
            buffer_adc_send1[2 * (i - LEN / 2)] = (buffer_test[i] >> 8) & 0xFF;
            buffer_adc_send1[2 * (i - LEN / 2) + 1] = buffer_test[i] & 0xFF;
        }

        // Zwięksź wartość iteration_count i zapisz ją w dwóch ostatnich pozycjach
        iteration_count++;
        buffer_adc_send1[LEN1 - 1] = (iteration_count >> 8) & 0xFF;
        buffer_adc_send1[LEN1 - 2] = iteration_count & 0xFF;

        // Wyczysć flagę zakończenia transferu DMA
        DMA_ClearFlag(DMA_FLAG_TC);
        // Zaktualizuj dane ADC
        ADC_Update1((uint8_t *)buffer_adc_send1, LEN1);
    }
}
*****

```

* Nazwa funkcji: **I2C_READ_MCP40D18TV2**

* Opis: funkcja obsługująca komendę Read potencjometru

* Pobiera: bufor danych do zapisu , adres urządzenia , ilość danych do zapisu, ilość danych do odczytu , bufor na dane do odczytu

* Zwraca:

```

*****
int I2C_READ_MCP40D18TV2(uint8_t* pBuffer, uint8_t DeviceAddr,uint8_t
NumByteToWrite,uint8_t NumByteToRead,uint8_t* xBuffer)
{
    I2C_TransactionType t;

    // Ustawienie parametrów transakcji I2C dla operacji zapisu
    t.Operation = I2C_Operation_Write;
    t.Address = DeviceAddr;
    t.StartByte = I2C_StartByte_Enable;

```

```

t.AddressType = I2C_AddressType_7Bit;
t.StopCondition = I2C_StopCondition_Disable;
t.Length = NumByteToWrite;

// Wyczyszczenie bufora Tx
I2C_FlushTx(I2C1);
while (I2C_WaitFlushTx(I2C1) == I2C_OP_ONGOING);

// Rozpoczęcie transakcji
I2C_BeginTransaction(I2C1, &t);
for (uint8_t i = 0; i < NumByteToWrite; i++) {
    I2C_FillTxFIFO(I2C1, pBuffer[i]);
}

// Czekanie na zakończenie operacji
do {
    if (I2C_GetStatus(I2C1) == I2C_OP_ABORTED) {
        // Tutaj zwracany jest błąd
        return ERROR;
    }
} while (I2C_GetITStatus(I2C1, I2C_IT_MTDWS) == RESET);

// Wyczyszczenie
// Wyczyszczenie bitów oczekiwania
I2C_ClearITPendingBit(I2C1, I2C_IT_MTDWS);

// Odczyt danych
t.Operation = I2C_Operation_Read;
t.Address = DeviceAddr;
t.StartByte = I2C_StartByte_Enable;
t.AddressType = I2C_AddressType_7Bit;
t.StopCondition = I2C_StopCondition_Disable;
t.Length = NumByteToRead;
I2C_BeginTransaction(I2C1, &t);

// Petla oczekiwania
do {
    if(I2C_OP_ABORTED == I2C_GetStatus(I2C1))
        return ERROR;
}
while (RESET == I2C_GetITStatus(I2C1, I2C_IT_MTD));

// Wyczyszczenie bitów oczekiwania
I2C_ClearITPendingBit(I2C1,I2C_IT_MTD | I2C_IT_MTDWS);

// Pobranie danych z bufora RX
while(NumByteToRead--) {
    *xBuffer = I2C_ReceiveData(I2C1);
    xBuffer++;
}

return SUCCESS;
}
}
*****  

* Nazwa funkcji: I2C_WRITE_MCP40D18T
* Opis: funkcja obsługująca komendę Wite potencjometru

```

```

* Pobiera: bufor danych , adres urządzenia, ilość bajtów do komunikacji
* Zwraca: status
*****
int I2C_WRITE_MCP40D18T(uint8_t* pBuffer, uint8_t DeviceAddr, uint8_t
NumByteToWrite){
    I2C_TransactionType t;

    // Zapisz adres urządzenia I2C
    t.Operation = I2C_Operation_Write;
    t.Address = DeviceAddr;
    t.StartByte = I2C_StartByte_Enable;
    t.AddressType = I2C_AddressType_7Bit;
    t.StopCondition = I2C_StopCondition_Enable;
    t.Length = NumByteToWrite;

    // Wyczyszczenie bufora TX
    I2C_FlushTx(I2C1);

    // Zapisz adres urządzenia I2C i umieść wartość send_val w buforze TX FIFO
    I2C_BeginTransaction(I2C1, &t);

    // Wypełnienie bufora TX FIFO danymi do wysłania
    for(uint8_t i=0; i<NumByteToWrite;i++) {
        I2C_FillTxFIFO(I2C1, pBuffer[i]);
    }

    // Pętla oczekiwania
    do {
        if(I2C_OP_ABORTED == I2C_GetStatus(I2C1))
            return I2C_OP_FAILED;
    }
    while (I2C_GetITStatus(I2C1, I2C_IT_MTD) == RESET);

    // Wyczyszczenie bitów oczekiwania
    I2C_ClearITPendingBit(I2C1, I2C_IT_MTD | I2C_IT_MTDWS);

    return SUCCESS;
}

```

Załącznik B: Aplikacje PC

Notyfie.cpp

```
#include <chrono>
#include <fstream>
#include <iomanip>
#include <iostream>
#include <sstream>
#include <thread>
#include <vector>
#include "../common/utils.hpp"

#include <atomic>
#include <chrono>
#include <condition_variable>
#include <mutex>
#include <string>
#include "simpleble/SimpleBLE.h"

using namespace std::chrono_literals;

std::atomic<bool> done(false);

bool is_done() { return done; }

int main() {
    auto adapter_optional = Utils::getAdapter();

    if (!adapter_optional.has_value()) {
        return EXIT_FAILURE;
    }

    auto adapter = adapter_optional.value();

    std::vector<SimpleBLE::Peripheral> peripherals;

    adapter.set_callback_on_scan_found([&](SimpleBLE::Peripheral peripheral) {
        std::cout << "Znaleziono urzadzenia: " << peripheral.identifier() << " ["
<< peripheral.address() << "]"
        << std::endl;
        ;
        peripherals.push_back(peripheral);
    });

    adapter.set_callback_on_scan_start([]() { std::cout << "Rozpoczeto skanowanie." << std::endl; });
    adapter.set_callback_on_scan_stop([]() { std::cout << "Zakonczono skanowanie." << std::endl; });
    // Scan for 5 seconds and return.
    adapter.scan_for(5000);

    std::cout << "Znaleziono nastepujace urzadzenia:" << std::endl;
    for (size_t i = 0; i < peripherals.size(); i++) {
        std::cout << "[" << i << "] " << peripherals[i].identifier() << " [" <<
peripherals[i].address() << "]"
        << std::endl;
    }
}

auto selection = Utils::getUserInputInt("Wybierz urzadzenie z ktorym chcesz nawiazac polaczenie", peripherals.size() - 1);
if (!selection.has_value()) {
```

```

        return EXIT_FAILURE;
    }

    auto peripheral = peripherals[selection.value()];
    std::cout << "laczenie z " << peripheral.identifier() << " [" <<
peripheral.address() << "]" << std::endl;
    peripheral.connect();

    std::cout << "Udalo sie polaczyc z urzadzeniem, posiada ono nastepuace
charakterystyki oraz seriwsy" << std::endl;

    // Store all service and characteristic uuids in a vector.
    std::vector<std::pair<SimpleBLE::BluetoothUUID, SimpleBLE::BluetoothUUID>>
uuids;
    for (auto service : peripheral.services()) {
        for (auto characteristic : service.characteristics()) {
            uuids.push_back(std::make_pair(service.uuid(),
characteristic.uuid()));
        }
    }

    std::cout << "Nastepujace charakterystyki zostaly znalezione:" << std::endl;
    for (size_t i = 0; i < uuids.size(); i++) {
        std::cout << "[" << i << "] " << uuids[i].first << " " << uuids[i].second
<< std::endl;
    }

    selection = Utils::getUserInputInt("Wybierz charakterystyke do subskypcji",
uuids.size() - 1);

    if (!selection.has_value()) {
        return EXIT_FAILURE;
    }
    size_t buffer_size = 5000;
    std::cout << "Podaj liczbe buforow do wypełnienia (kazdy posiada 5k sampli) :
";
    size_t num_buffers_to_fill;
    std::cin >> num_buffers_to_fill;

    size_t total_bytes_received = 0;
    size_t total_notifications_received = 0;
    std::string file_name;
    std::cout << "Wprowadz nazwe pliku, w ktorym zosana zapisane dane: ";
    std::cin.ignore();
    std::getline(std::cin, file_name);

    // zrob bufor
    std::vector<SimpleBLE::ByteArray> received_data_buffer;

    std::chrono::high_resolution_clock::time_point start_time, end_time;
    std::atomic<bool> done(false);
    std::mutex data_mutex;
    std::condition_variable done_cv;

    received_data_buffer.reserve(num_buffers_to_fill);

    peripheral.notify(uuids[selection.value()].first,
uuids[selection.value()].second, [&](SimpleBLE::ByteArray bytes) {
        std::unique_lock<std::mutex> lock(data_mutex);
        std::cout << "Otrzymano: ";
        Utils::print_byte_array(bytes);
        if (total_notifications_received == 0) {

```

```

        // start_time = std::chrono::high_resolution_clock::now();
    }

    // wpisz dane do buffora
    received_data_buffer.push_back(bytes);

    total_bytes_received += bytes.size();
    total_notifications_received++;
    if (total_bytes_received >= num_buffers_to_fill * buffer_size) {
        peripheral.unsubscribe(uuids[selection.value()].first,
        uuids[selection.value()].second);
        end_time = std::chrono::high_resolution_clock::now(); // Store the
    end time
        done = true;
        done_cv.notify_one();
    }
});

std::unique_lock<std::mutex> lock(data_mutex);
done_cv.wait(lock, [&]() { return done.load(); });

peripheral.unsubscribe(uuids[selection.value()].first,
uuids[selection.value()].second);
peripheral.disconnect();

//zapisz dane do pliku
std::ofstream output_file(file_name, std::ios::out | std::ios::binary);
for (const auto& bytes : received_data_buffer) {
    for (const auto& byte : bytes) {
        output_file << std::hex << std::setw(2) << std::setfill('0')
            << static_cast<unsigned int>(static_cast<unsigned
char>(byte)) << " ";
    }
    output_file << std::endl;
}
output_file.close();

std::chrono::duration<double> elapsed_time = end_time - start_time;
std::cout << "Czas notyfikacji wyniosl : " << elapsed_time.count() << " s" <<
std::endl;

```

Funkcja o rozszerzonej funkcjonalności

```

    return EXIT_SUCCESS;
}#include <atomic>
#include <chrono>
#include <condition_variable>
#include <fstream>
#include <fstream>
#include <iomanip>
#include <iostream>
#include <mutex>
#include <thread>
#include <sstream>
#include <vector>
#include "../common/utils.hpp"
#include <thread>
#include <future>
#include <memory>
#include <functional>

```

```

#include "simpleble/SimpleBLE.h"
using namespace std;

using namespace std::chrono_literals;
void cls() { std::cout << "\033[2J\033[1;1H"; }
void writerThreadFunc(const std::vector<char>& buffer, std::ofstream& fout) {
    for (size_t i = 0; i < buffer.size(); i++) {
        fout << std::hex << std::setw(2) << std::setfill('0') << (int)(unsigned
char)buffer[i] << " ";
    }
}

int main() {
    auto adapter_optional = Utils::getAdapter();

    if (!adapter_optional.has_value()) {
        return EXIT_FAILURE;
    }

    auto adapter = adapter_optional.value();

    std::vector<SimpleBLE::Peripheral> peripherals;

    adapter.set_callback_on_scan_found([&](SimpleBLE::Peripheral peripheral) {
        std::cout << "Znaleziono: " << peripheral.identifier() << " [" <<
peripheral.address() << "]" << std::endl;
        peripherals.push_back(peripheral);
    });

    adapter.set_callback_on_scan_start([]() { std::cout << "Skanowanie." <<
std::endl; });
    adapter.set_callback_on_scan_stop([]() { std::cout << "Skanowanie
zakonczono." << std::endl; });
    // Scan for 5 seconds and return.
    adapter.scan_for(3000);

    std::cout << "Znaleziono urzadzenie:" << std::endl;
    for (size_t i = 0; i < peripherals.size(); i++) {
        std::cout << "[" << i << "] " << peripherals[i].identifier() << " [" <<
peripherals[i].address() << "]"
        << std::endl;
    }

    auto selection = Utils::getUserInputInt("WYBIERZ URZADZENIE Z KTORYM CHCESZ
SIE POLOCZYC", peripherals.size() - 1);

    if (!selection.has_value()) {
        return EXIT_FAILURE;
    }

    auto peripheral = peripherals[selection.value()];
    std::cout << "LACZENIE Z " << peripheral.identifier() << " [" <<
peripheral.address() << "]" << std::endl;
    peripheral.connect();
}

```

```

        std::vector<std::pair<SimpleBLE::BluetoothUUID, SimpleBLE::BluetoothUUID>>
uuids;
    for (auto service : peripheral.services()) {
        for (auto characteristic : service.characteristics()) {
            uuids.push_back(std::make_pair(service.uuid(),
characteristic.uuid()));
        }
    }
    cls();
}

peripheral.identifier() << std::endl;
std::cout << "NAZWA:" <<
std::endl;
std::cout << "MENU" <<
std::endl;
int pick = 0;
bool quit = false;

do {
    std::cout << "WYBIERZ OPCJE:" << endl
    << "[1]" Subskrypcja
    "charakterystyki noatyfikacyjnej" << endl
    << "[2]" Rozloancz z
    "urzadzeniem" << endl
    << "[3]" Subskrypcja
    "charakterystyki write" << endl
    << "[4]" Subskrycja
    "charakterystki read" << endl
    << "[5]" Pokaz serwisy oraz
    "charakterystyki" << endl
    << "[6]" Pokaz parametry
    "transmisijs urzadzenia" << endl;
    std::cin >> pick;
    int bufferCount;
    if (pick == 1) {

        std::vector<std::pair<SimpleBLE::BluetoothUUID,
SimpleBLE::BluetoothUUID>> uuids;
        for (auto service : peripheral.services()) {
            for (auto characteristic : service.characteristics()) {
                uuids.push_back(std::make_pair(service.uuid(),
characteristic.uuid()));
            }
        }
    }

    std::cout << "Nastepujace charakterystyki zostaly znalezione:" <<
std::endl;
    for (size_t i = 0; i < uuids.size(); i++) {
        std::cout << "[" << i << "] " << uuids[i].first << " " <<
uuids[i].second << std::endl;
    }

    selection = Utils::getUserInputInt("Wybierz charakterystyke do
subskrypcji", uuids.size() - 1);

    if (!selection.has_value()) {

```

```

        return EXIT_FAILURE;
    }
    size_t buffer_size = 5000;
    std::cout << "Podaj liczbe buforow do wypełnienia (kazdy posiada 5k
sampli) : ";
    size_t num_buffers_to_fill;
    std::cin >> num_buffers_to_fill;

    size_t total_bytes_received = 0;
    size_t total_notifications_received = 0;
    std::string file_name;
    std::cout << "Wprowadz nazwe pliku, w którym zosąñ zapisane dane:
";
    std::cin.ignore();
    std::getline(std::cin, file_name);

    std::vector<SimpleBLE::ByteArray> received_data_buffer;

    std::chrono::high_resolution_clock::time_point start_time,
end_time;
    std::atomic<bool> done(false);
    std::mutex data_mutex;
    std::condition_variable done_cv;

    received_data_buffer.reserve(num_buffers_to_fill);

    peripheral.notify(
        uuids[selection.value()].first,
        uuids[selection.value()].second, [&](SimpleBLE::ByteArray bytes) {
            std::unique_lock<std::mutex> lock(data_mutex);
            std::cout << "Otrzymano: ";
            Utils::print_byte_array(bytes);
            if (total_notifications_received == 0) {

                start_time = std::chrono::high_resolution_clock::now();
            }

            received_data_buffer.push_back(bytes);

            total_bytes_received += bytes.size();
            total_notifications_received++;
            if (total_bytes_received >= num_buffers_to_fill * buffer_size) {
                peripheral.unsubscribe(uuids[selection.value()].first,
                uuids[selection.value()].second);
                end_time = std::chrono::high_resolution_clock::now();
// Store the end time
                done = true;
                done_cv.notify_one();
            }
        });
    std::unique_lock<std::mutex> lock(data_mutex);

```

```

        done_cv.wait(lock, [&]() { return done.load(); });

        peripheral.unsubscribe(uuids[selection.value()].first,
uuids[selection.value()].second);
        peripheral.disconnect();

        std::ofstream      output_file(file_name,      std::ios::out      |
std::ios::binary);
        for (const auto& bytes : received_data_buffer) {
            for (const auto& byte : bytes) {
                output_file << std::hex << std::setw(2) << std::setfill('0')
                << static_cast<unsigned
int>(static_cast<unsigned char>(byte)) << " ";
            }
            output_file << std::endl;
        }
        output_file.close();

        std::chrono::duration<double> elapsed_time = end_time - start_time;
        std::cout << "Czas notyfikacji wyniosl : " << elapsed_time.count()
<< " s" << std::endl;
    }

    else if (pick == 2) {
        std::cout << "\t ROZLONCZONO Z URZADZENIEM" << endl;

        peripheral.disconnect();
        break;
    } else if (pick == 3) {
        std::vector<std::pair<SimpleBLE::BluetoothUUID,
SimpleBLE::BluetoothUUID>> uuids;
        for (auto service : peripheral.services()) {
            for (auto characteristic : service.characteristics()) {
                uuids.push_back(std::make_pair(service.uuid(),
characteristic.uuid()));
            }
        }

        std::cout << " Nastepujace charakterystyki zostaly znalezione : " <<
std::endl;
        for (size_t i = 0; i < uuids.size(); i++) {
            std::cout << "[" << i << "] " << uuids[i].first << " " <<
uuids[i].second << std::endl;
        }
    }

    selection = Utils::getUserInputInt("Wybierz charakterystyke: ",
uuids.size() - 1);

    std::string contents;
    std::cout << "Wpisz dane do charakterystyki: ";
    std::cin >> contents;

    if (!selection.has_value()) {
        return EXIT_FAILURE;
    }
}

```

```

// NOTE: Alternatively, 'write_command' can be used to write to a
characteristic too.
    // 'write_request' is for unacknowledged writes.
    // 'write_command' is for acknowledged writes.
    peripheral.write_request(uuids[selection.value()].first,
uuids[selection.value()].second, contents);

    if (!selection.has_value()) {
        return EXIT_FAILURE;
    } else if (pick == 4) {
        std::vector<std::pair<SimpleBLE::BluetoothUUID,
SimpleBLE::BluetoothUUID>> uuids;
        for (auto service : peripheral.services()) {
            for (auto characteristic : service.characteristics()) {
                uuids.push_back(std::make_pair(service.uuid(),
characteristic.uuid()));
            }
        }
    }

        std::cout << "Nastepujace charakterystyki zostaly
znalezione:" << std::endl;
        for (size_t i = 0; i < uuids.size(); i++) {
            std::cout << "[" << i << "] " << uuids[i].first << " "
<< uuids[i].second << std::endl;
        }

    selection = Utils::getUserInputInt("Wybierz charakterystyke
do odczytu", uuids.size() - 1);

    if (!selection.has_value()) {
        return EXIT_FAILURE;
    }

    // Attempt to read the characteristic 5 times in 5 seconds.
    for (size_t i = 0; i < 5; i++) {
        SimpleBLE::ByteArray rx_data = peripheral.read(uuids[selection.value()].first,
uuids[selection.value()].second);
        std::cout << "Dane z charakterystyki: ";
        Utils::print_byte_array(rx_data);
        std::this_thread::sleep_for(1s);
    }

}

} else if (pick == 5) {
    std::cout << "\t      POLOACZONO Z BLUE-NRG2." << std::endl;
    std::cout << "\t      MTU: " << peripheral.mtu() << std::endl;
    for (auto& service : peripheral.services()) {
        std::cout << "\t      SERWIS: " << endl << service.uuid() <<
std::endl;
}
}

```

```

        for (auto& characteristic : service.characteristics()) {
            std::cout << "\t     CHARAKTERYSTYKA NALEZACA DO SERWISU
: " << endl
                << characteristic.uuid() << std::endl;

            std::cout << "\t ZDOLNOSCI : ";
            for (auto& capability : characteristic.capabilities())
{
                std::cout << capability << " ";
}
            std::cout << std::endl;

            for (auto& descriptor : characteristic.descriptors()) {
                std::cout << "\t Descriptor: " << descriptor.uuid()
<< std::endl;
}
}
}

else if (pick == 6) {
    std::cout << "\t NAZWA :" << peripheral.identifier() <<
std::endl;
    std::cout << "\t TX POWER: " << peripheral.tx_power() <<
std::endl;
    std::cout << "\t RSSI: " << peripheral.rssi() << std::endl;
    std::cout << "\t ADDRESS: " << peripheral.address() << std::endl
<< std::endl;
}

else {
    std::cout << "WYBIERZ POPRAWNA OPCJE " << endl;
}
}

while (!quit);

return EXIT_SUCCESS;
}

```

Notyfie_multiple.cpp

```
#include <atomic>
#include <chrono>
#include <fstream>
#include <iomanip>
#include <iostream>
#include <thread>
#include <vector>
#include <sstream>
#include "../common/utils.hpp"
#include "simpleble/SimpleBLE.h"

using namespace std::chrono_literals;

std::string bytes_to_string(const SimpleBLE::ByteArray& bytes) {
    std::stringstream ss;
    for (const auto& byte : bytes) {
        ss << static_cast<char>(byte);
    }
    return ss.str();
}

int main() {
    std::vector<SimpleBLE::Peripheral> aux_p; // Helper vector to hold scan results.

    // Fetch the adapter object that will be used for this test.
    auto adapter_optional = Utils::getAdapter();
    if (!adapter_optional.has_value()) {
        return EXIT_FAILURE;
    }
    auto adapter = adapter_optional.value();

    // Configure adapter callbacks.
    adapter.set_callback_on_scan_found([&](SimpleBLE::Peripheral peripheral) {
        std::cout << "Found device: " << peripheral.identifier() << " [" <<
peripheral.address() << "]" << std::endl;
        aux_p.push_back(peripheral);
    });
    adapter.set_callback_on_scan_start([]() { std::cout << "Scan started." <<
std::endl; });
    adapter.set_callback_on_scan_stop([]() { std::cout << "Scan stopped." <<
std::endl; });

    // Create two uninitialized peripherals that will hold our connections for this example.
    SimpleBLE::Peripheral peripherals[2];

    // Variable that enables printing per peripheral.
    std::atomic_bool print_allowed{false};

    std::vector<std::vector<SimpleBLE::ByteArray>> received_data_buffers(2);

    for (size_t iter = 0; iter < 2; iter++) {
        aux_p.clear();

        adapter.scan_for(7000);

        std::cout << "The following devices were found:" << std::endl;
```

```

        for (size_t i = 0; i < aux_p.size(); i++) {
            std::cout << "[" << i << "]" " << aux_p[i].identifier() << " [" <<
aux_p[i].address() << "]" << std::endl;
        }

        auto selection = Utils::getUserInputInt("Please select a device to
connect to", aux_p.size() - 1);
        if (!selection.has_value()) {
            return EXIT_FAILURE;
        }

        peripherals[iter] = aux_p[selection.value()];

        std::cout << "Connecting to " << peripherals[iter].identifier() << " [" <<
peripherals[iter].address() << "]"
                << std::endl;
        peripherals[iter].connect();

        std::cout << "Successfully connected, printing services and
characteristics.." << std::endl;

        // Store all service and characteristic uuids in a vector.
        std::vector<std::pair<SimpleBLE::BluetoothUUID,
SimpleBLE::BluetoothUUID>> uuids;
        for (auto service : peripherals[iter].services()) {
            for (auto characteristic : service.characteristics()) {
                uuids.push_back(std::make_pair(service.uuid(),
characteristic.uuid()));
            }
        }

        std::cout << "The following services and characteristics were found:" <<
std::endl;
        for (size_t i = 0; i < uuids.size(); i++) {
            std::cout << "[" << i << "]" " << uuids[i].first << " " <<
uuids[i].second << std::endl;
        }

        selection = Utils::getUserInputInt("Please select a characteristic to
read", uuids.size() - 1);

        if (!selection.has_value()) {
            return EXIT_FAILURE;
        }

        peripherals[iter].notify(uuids[selection.value()].first,
uuids[selection.value()].second,
        [&, iter](SimpleBLE::ByteArray bytes) {
            if (print_allowed) {
                std::cout << "Peripheral " << iter << "
received: ";
                Utils::print_byte_array(bytes);
            }
        });

        received_data_buffers[iter].push_back(bytes);
    });
}

print_allowed = true;

std::this_thread::sleep_for(5s);

```

```

print_allowed = false;

std::vector<std::thread> processing_threads;
for (size_t iter = 0; iter < 2; iter++) {
    peripherals[iter].disconnect();

    std::string output_file_name;
    if (iter == 0) {
        output_file_name = "pierwsze_urzadzenie.txt";
    } else {
        output_file_name = "drugie_urzadzenie.txt";
    }

    processing_threads.emplace_back([&received_data_buffers, iter,
output_file_name] {
        std::ofstream output_file(output_file_name, std::ios::out |
std::ios::binary);
        for (const auto& bytes : received_data_buffers[iter]) {
            for (const auto& byte : bytes) {
                output_file << std::hex << std::setw(2) << std::setfill('0')
                << static_cast<unsigned int>(static_cast<unsigned
char>(byte)) << " ";
            }
            output_file << std::endl;
        }
        output_file.close();
    });
}

for (auto& t : processing_threads) {
    t.join();
}

return EXIT_SUCCESS;
}

```

Załącznik C:Skrypty stworzone w programie Matlab

Skrypt matlab sprawdzający miejsca znaczące

```
*****  
* Opis: skrypt przetwarzający plik z danymi odbieranymi z mikrokontrolera przez aplikację,  
oraz sprawdzający miejsca znaczące.  
*****
```

```
% Otwórz plik tekstowy do odczytu  
plik = fopen('wyniki_25_04.txt', 'r');  
  
% Wczytaj dane z pliku tekstowego jako ciąg znaków  
data_str = fscanf(plik, '%s');  
  
% Zamknij plik  
fclose(plik);  
  
% Konwertuj ciąg znaków na wartości szesnastkowe  
data_hex = reshape(data_str, 2, []).' ;  
  
% Przekształć wartości szesnastkowe na wartości  
dziesiętne  
data_dec = hex2dec(data_hex);  
  
% Sprawdź, czy wartości na pozycjach 19 i 20 są równymi  
"00"  
czy_19_20_wynosza_zero = data_dec(19) == 0 &&  
data_dec(20) == 0;  
  
% Wyświetl wynik  
if czy_19_20_wynosza_zero  
    disp('Wartości na pozycjach 19 i 20 są równe "00"');  
else  
    disp('Wartości na pozycjach 19 i 20 NIE są równe  
"00"');  
end
```

Skrypt matlab porównujący znane dane wraz z zapisanymi do pliku

```
*****  
* Opis: skrypt przetwarzający plik z danymi odbieranymi z mikrokontrolera przez aplikację,  
oraz porównujący je z danymi referencyjnymi  
*****
```

```
% Odczyt pliku tekstowego  
filename = 'Zanane_dane_notyfikacja.txt';  
fileID = fopen(filename, 'r');  
raw_data = textscan(fileID, '%s');
```

```

fclose(fileID);
raw_data = raw_data{1};

% Konwersja danych na wartości szesnastkowe o długości 4
znaków
hex_data = reshape([raw_data{:}], 4, [])';

% Konwersja wartości szensatkowe na wartości dziesiętne
dec_data = hex2dec(hex_data);

% Wartości referencyjne
ref_values = [12000, 15000, 33800, 5400, 24500, 6700,
7500, 16000, 7800, 9600];

% Sprawdzenie, czy wartości dec_data są równe wartościom
referencyjnym
czy_sa_rowne = true;
for i = 1:length(ref_values)
    for j = i:10:length(dec_data)
        if dec_data(j) ~= ref_values(i)
            czy_sa_rowne = false;
            break;
        end
    end
    if ~czy_sa_rowne
        break;
    end
end

% Wyświetlenie wyniku
if czy_sa_rowne
    disp('Wszystkie wartości dec_data są równe wartościom
referencyjnym');
else
    disp('Wszystkie wartości dec_data NIE są równe
wartościom referencyjnym');
end
Skrypt matlab odczytujące dane zapisane do pliku
*****
* Opis: skrypt przetwarzający plik z danymi odbieranymi z mikrokontrolera przez aplikację,
oraz porównujący je z danymi referencyjnymi
*****
% Odczytaj plik tekstowy
filename = 'dane_przed_ramie_v2.txt';
fileID = fopen(filename, 'r');
raw_data = fscanf(fileID, '%s');
fclose(fileID);

```

```

% Usuń spacje i podziel dane na wartości szesnastkowe o 4
znakach
raw_data_no_spaces = erase(raw_data, ' ');
hex_data = reshape(raw_data_no_spaces, 4, []);

% Konwersja danych szesnastkowych na dziesiętne
dec_data = hex2dec(hex_data);
full_data=0.6+(dec_data/41260)21.2;

% Wykres danych
figure;
plot(full_data);
title('Wykres danych dziesiętnych');
xlabel('Indeks');
ylabel('Wartość');

Skrypt matlab odczytujące dane zapisane do pliku
/****************************************************************************
 * Opis: skrypt przetwarzający plik z danymi odbieranymi z mikrokontrolera przez aplikację,
****************************************************************************/
% wczytaj plik
filename = 'file.txt';
fileID = fopen(filename, 'r');
lines = textscan(fileID, '%s', 'Delimiter', '\n');
fclose(fileID);
lines = lines{1};

combined_hex_data = strings(length(lines), 1);

for i = 1:length(lines)
    % wyodrębnij 19. i 20. bajt i zamień ich kolejność
    hex_data = split(lines{i});
    extracted_hex_data = [hex_data(20); hex_data(19)];

    % połącz zamienione wartości szesnastkowe
    combined_hex_data(i) = strcat(extracted_hex_data(1),
extracted_hex_data(2));
end

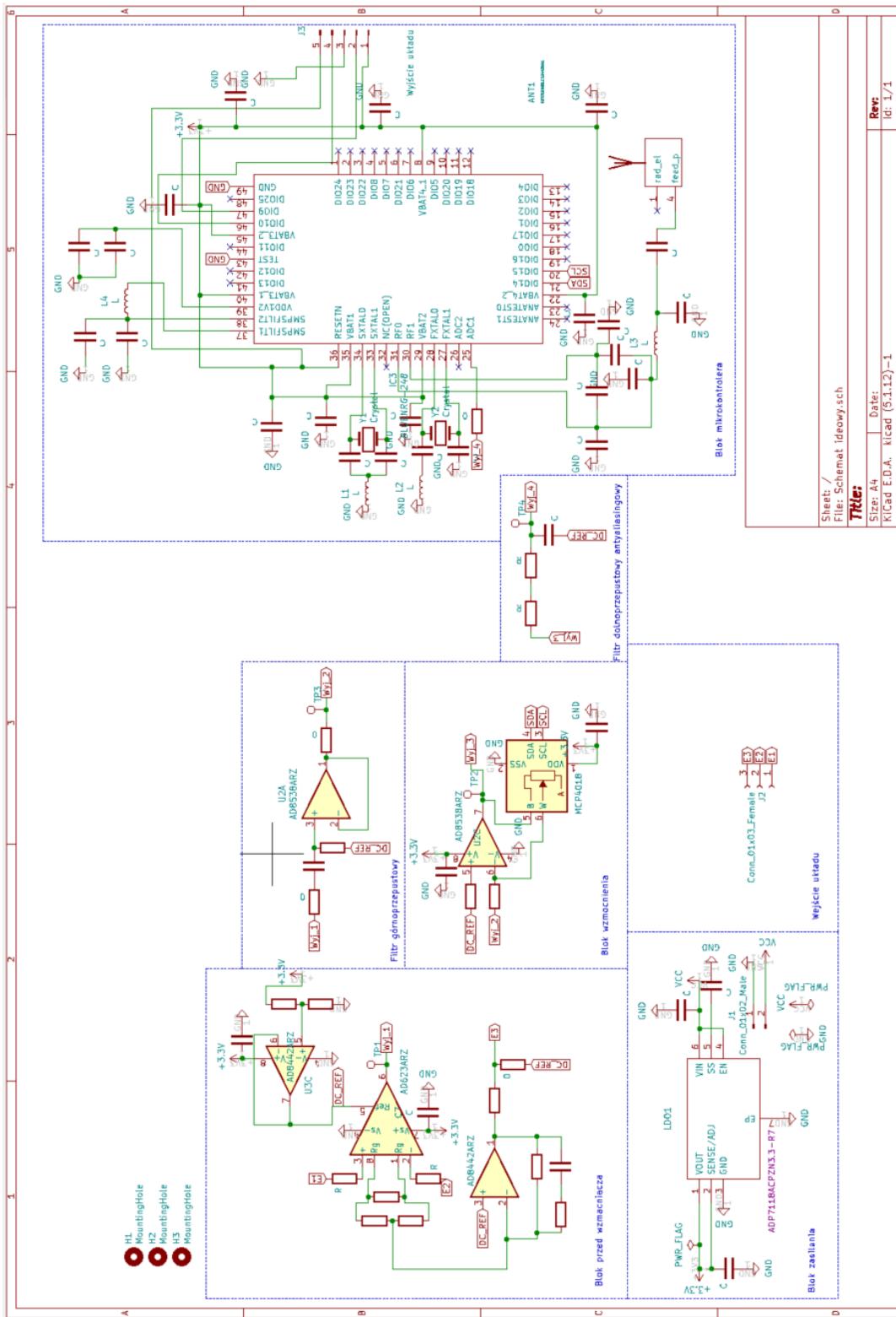
% konwersja szesnastkowych na dziesiętne
dec_data = hex2dec(combined_hex_data);

figure;
plot(dec_data);

```

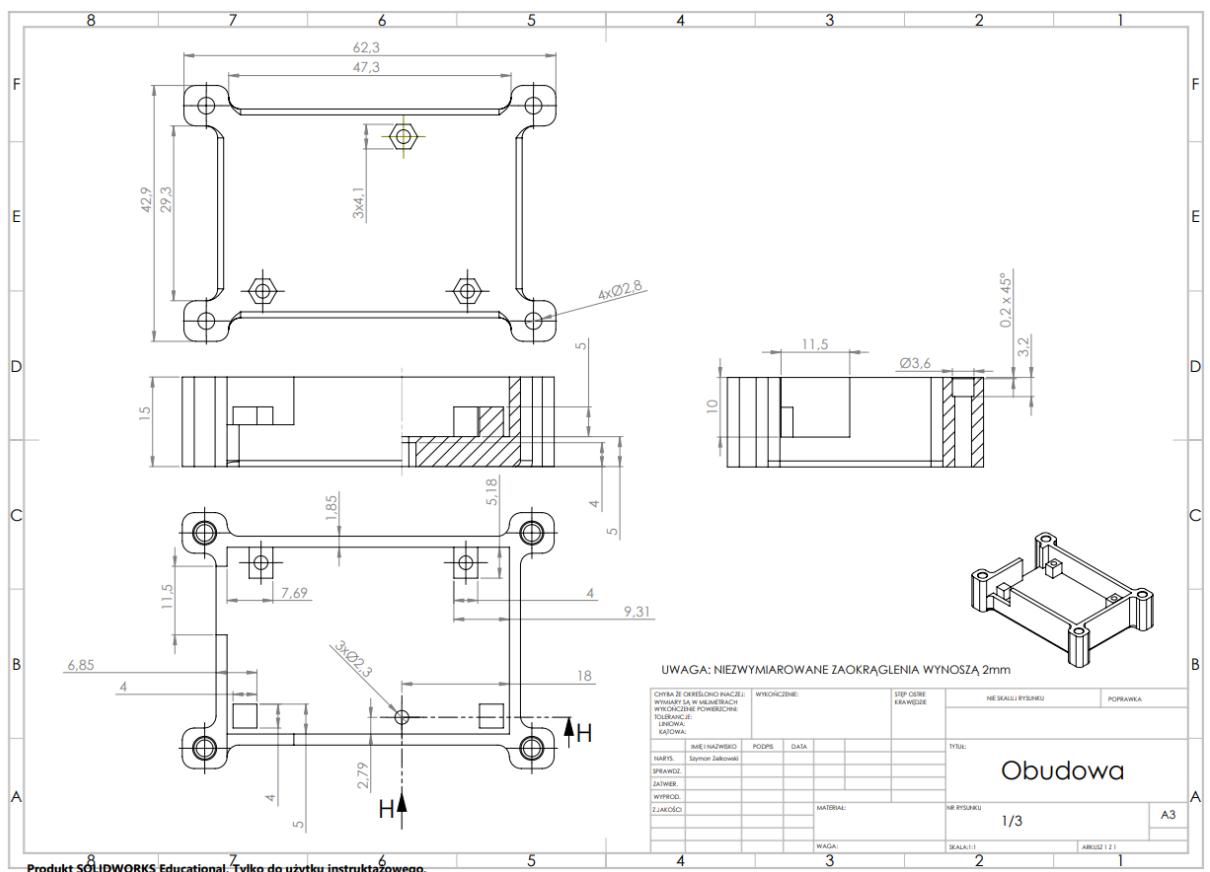
```
title('Wykres danych dziesiętnych');  
 xlabel('Indeks');  
 ylabel('Wartość');
```

Załącznik D. Schematy ideowy

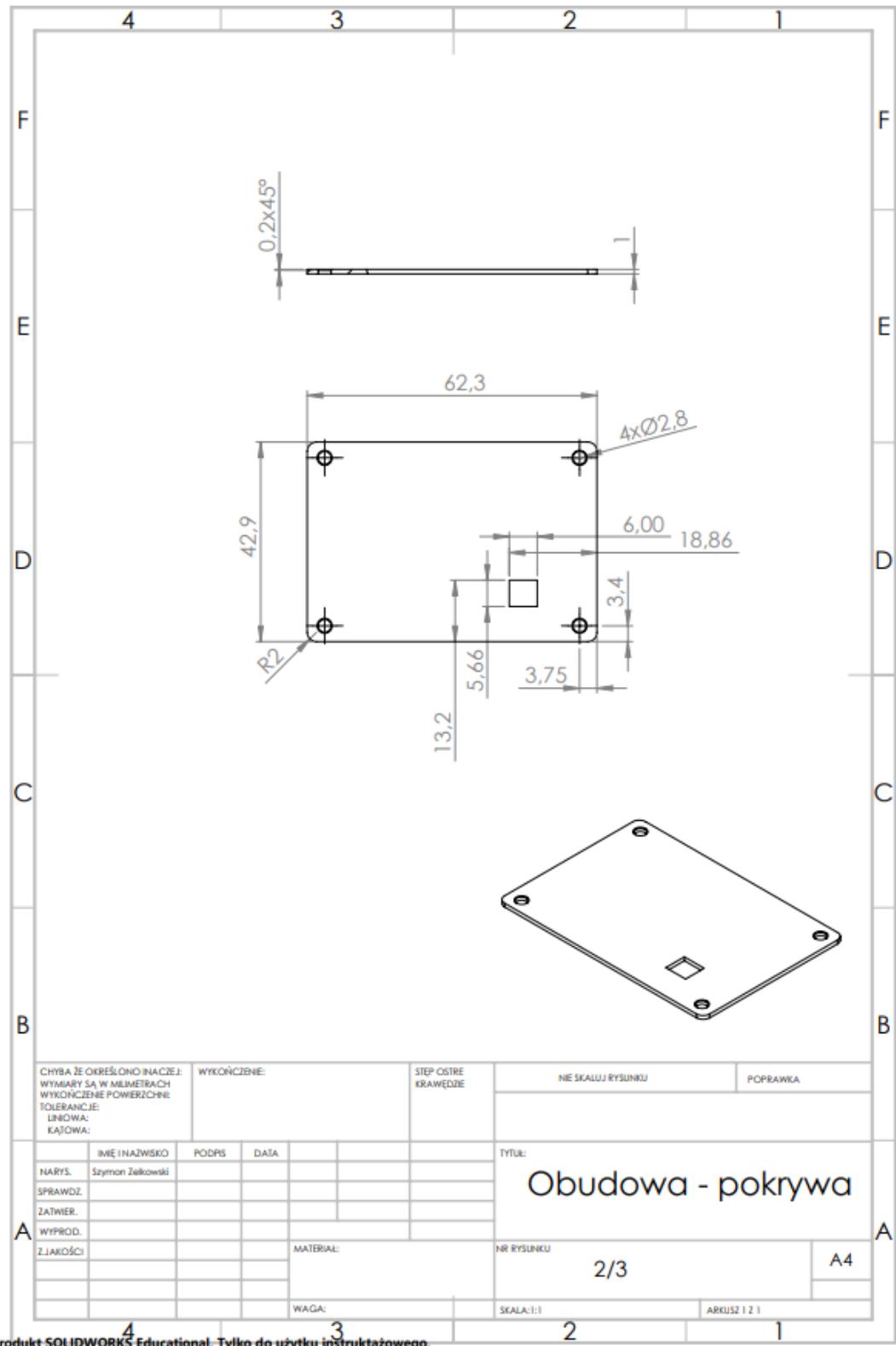


Schemat układu wzmacniacza

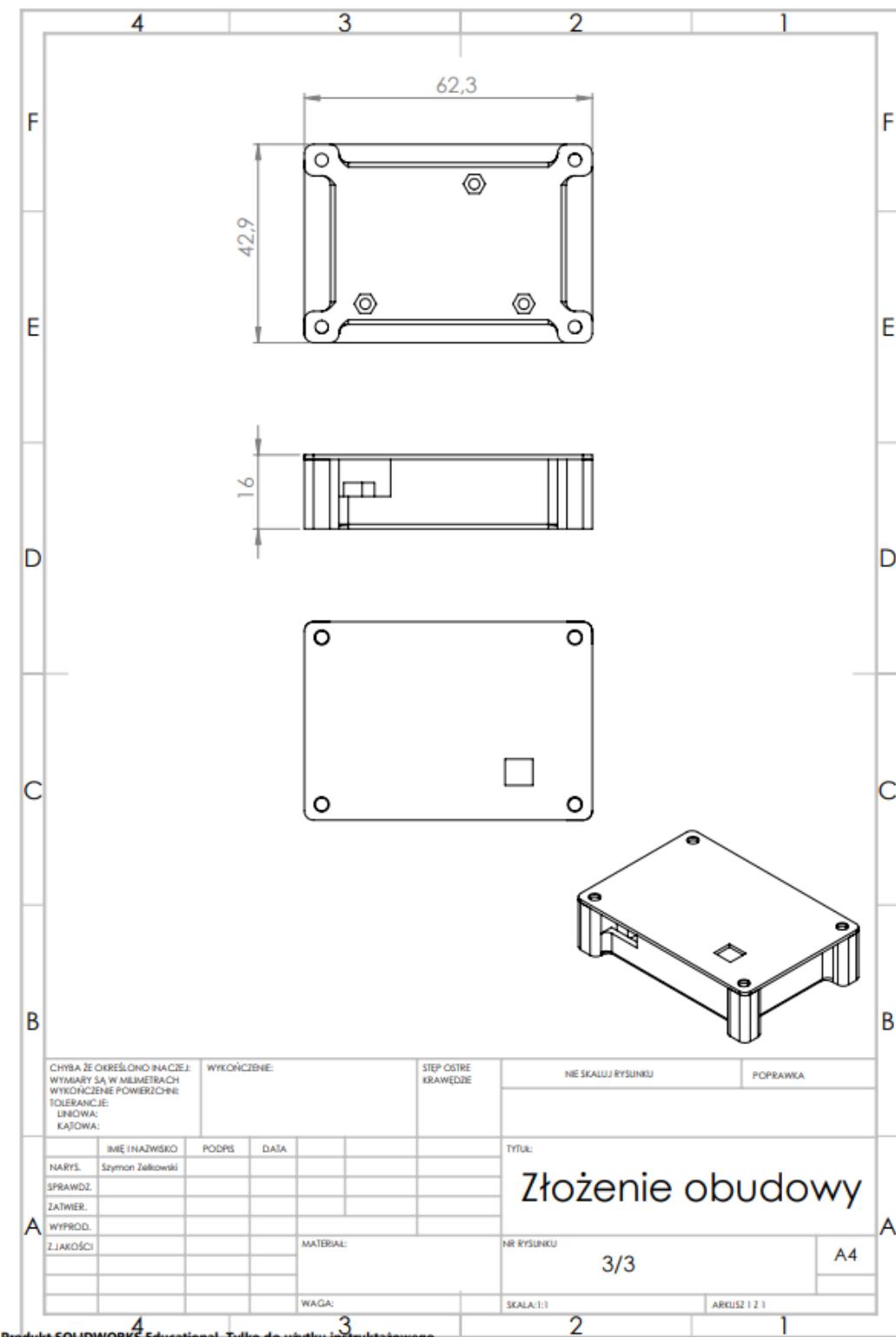
Załącznik E. Projekt obudowy płytki



Rysunek wykonawczy części obudowy

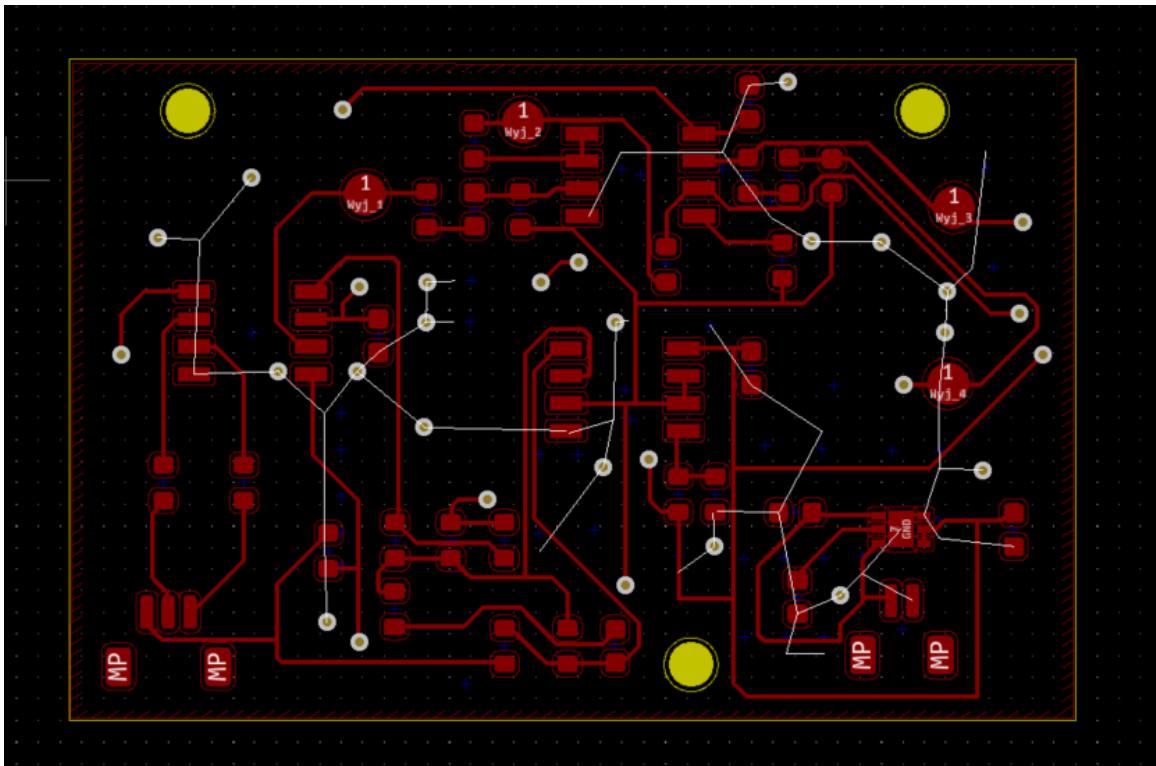


Rysunek wykonawczy części obudowy

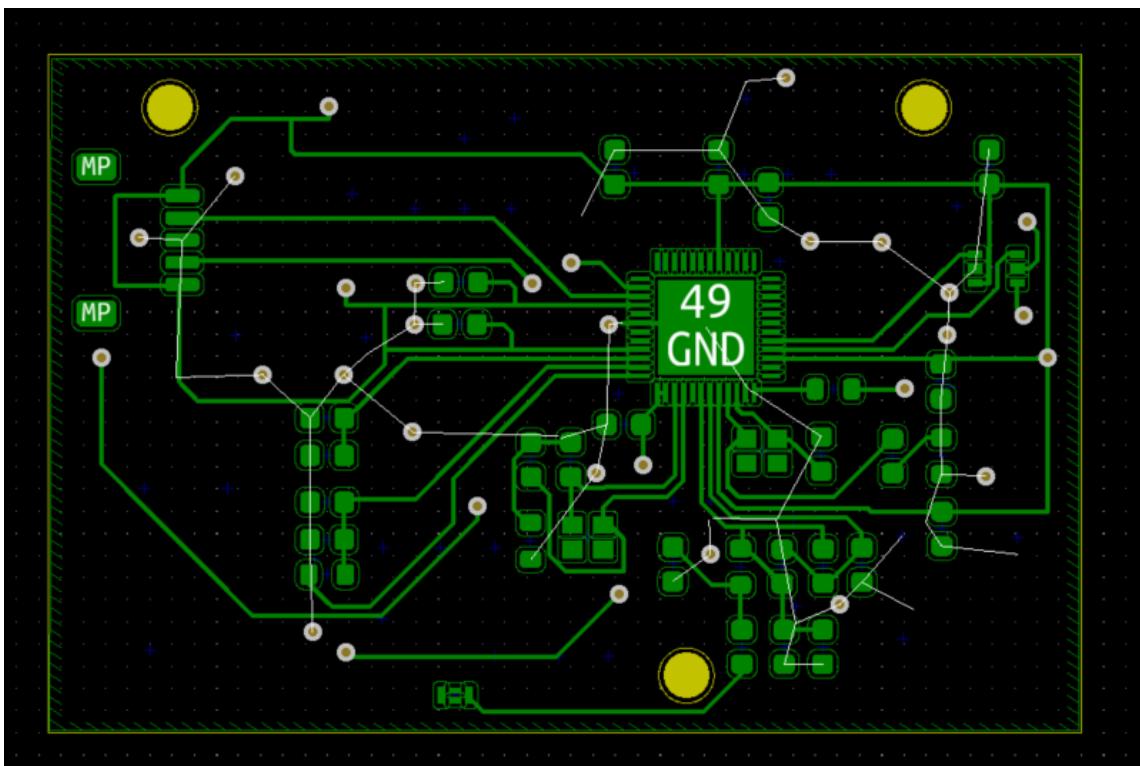


Rysunek złożeniowy obudowy

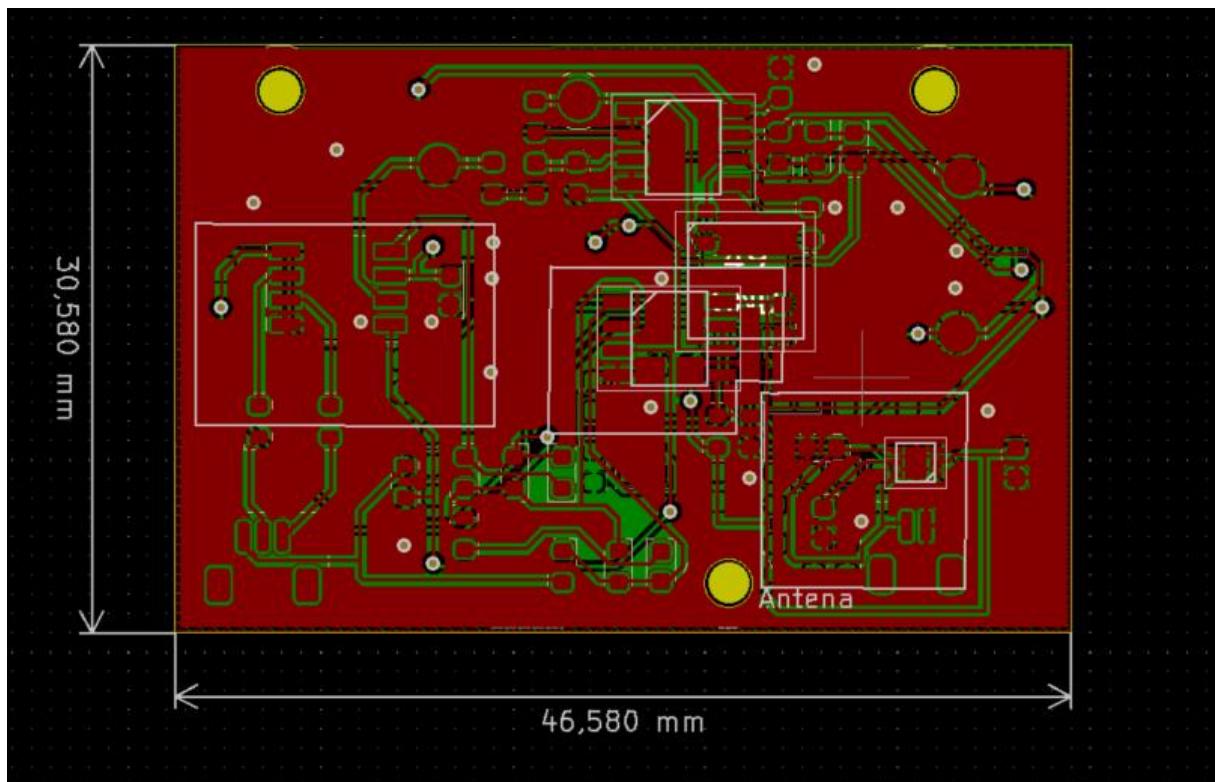
Załącznik F. Projekt obwodu drukowanego PCB



Widok górnej warstwy F.CU projektu płytki PCB



Widok dolnej warstwy F.CU projektu płytki PCB



Widok wszystkich warstw płytki PCB wraz z wymiarami