



**Politechnika
Śląska**

PRACA MAGISTERSKA

Aktywne obciążenie do testowania akumulatorów

Marek GAŁECZKA
Nr albumu: 296799

Kierunek: Automatyka i Robotyka
Specjalność: SPiL

PROWADZĄCY PRACĘ
dr inż. Wojciech Oliwa

KATEDRA Elektroniki, elektrotechniki i mikroelektroniki
Wydział Automatyki, Elektroniki i Informatyki

Gliwice 2025

Tytuł pracy

Aktywne obciążenie do testowania akumulatorów

Streszczenie

Praca przedstawia projekt i realizację aktywnego obciążenia przeznaczonego do testowania akumulatorów 12 V, działającego w trybach stałego prądu (CC), stałej rezystancji (CR) oraz stałej mocy (CP). Układ został oparty na tranzystorze MOSFET IRLZ44N sterowanym przetwornikiem DAC8830 i regulatorem PID z kompensacją temperaturową, natomiast pomiary napięcia i prądu realizują 24-bitowe przetworniki ADC MCP3561. System obejmuje lokalny panel użytkownika, moduł zapisu danych na kartę SD oraz aplikację komputerową komunikującą się z urządzeniem przez interfejs UART. Urządzenie umożliwia prowadzenie testów prądami do 14 A i jest przystosowane do pracy ciągłej. Projekt został zwieńczony badaniami, które potwierdziły stabilne utrzymywanie wartości zadanych, wysoką precyzję pomiarów oraz spełnienie założeń projektowych.

Słowa kluczowe

Obciążenie aktywne, MOSFET, regulator PID, elektronika, akumulator

Thesis title

Active load for battery testing

Abstract

The thesis presents the design and implementation of an active load intended for testing 12 V batteries, operating in constant-current (CC), constant-resistance (CR), and constant-power (CP) modes. The circuit is based on an IRLZ44N MOSFET driven by a DAC8830 and a PID controller with temperature compensation, while voltage and current are measured by 24-bit MCP3561 ADCs. The system includes a local user interface, data logging to an SD card, and a PC application that communicates with the device via a UART interface. The device enables tests with currents up to 14 A and is designed for continuous operation. The project culminated in testing that confirmed stable tracking of setpoints, high measurement accuracy, and fulfillment of the design objectives.

Keywords

Active load, MOSFET, PID regulator, electronics, accumulator

Spis treści

1 Wstęp	1
2 Analiza tematu	3
2.1 Obciążenia pasywne i aktywne	3
2.1.1 Tryby pracy aktywnego obciążenia	4
2.2 Przegląd gotowych rozwiązań	5
2.3 Ogólny schemat blokowy aktywnego obciążenia	6
2.3.1 Układ główny obciążenia	6
2.3.2 Układ pomiarowy	7
2.3.3 Układ chłodzenia	7
2.3.4 Mikrokontroler	7
2.3.5 Wymagania pomiarowe i metrologiczne	8
2.3.6 Panel sterowania	8
2.3.7 Układ zapisu danych	9
2.3.8 Podsumowanie i cel pracy	9
3 Realizacja sprzętowa - Koncepcja i schemat elektryczny	11
3.1 Układ główny obciążenia	12
3.1.1 Tranzystor IRLZ44N	12
3.1.2 Rezystory mocy	15
3.1.3 Przekaźniki, tranzystory, rezystory SMD i diody	16
3.2 Układy pomiarowe	17
3.3 Układ zadawania napięcia sterującego	20
3.4 Układ chłodzenia i czujniki temperatury	21
3.5 Panel sterowania	22
3.6 Układ zapisu danych na zewnętrznym nośniku pamięci – karta SD	23
3.7 Układ zasilania i napięcia referencyjnego	24
4 Realizacja sprzętowa - Projekt PCB i budowa	27
4.1 Układ obciążenia - PCB	28
4.2 Układ pomiarowy i zadawania napięcia sterującego – PCB.	29

4.3	Moduł sterowania.	29
4.4	Panel Sterowania.	29
4.5	Układ zasilania i referencji.	30
4.6	Układ zapisu danych.	30
4.7	Czujniki temperatury i wentylator.	30
4.8	Obudowa.	31
5	Oprogramowanie mikrokontrolera i aplikacji	33
5.1	Struktura programu.	34
5.2	Główna pętla programu.	34
5.3	Struktura DataStruct.	35
5.4	Pomiary napięcia i prądu obciążenia.	36
5.5	Pomiar temperatury.	40
5.6	Regulator i kompensacja temperaturowa.	41
5.7	Zapis danych pomiarowych na kartę SD.	44
5.8	Panel sterowania i Aplikacja na PC.	45
5.9	Komunikacja z PC - UART.	48
5.10	Tryby pracy obciążenia.	50
5.11	Warunki STOP.	50
5.12	Aplikacja na komputer PC.	51
6	Badania i pomiary	53
6.1	Badanie charakterystyki w trybie CC (stały prąd).	53
6.1.1	Stan Przejściowy.	54
6.1.2	Analiza stanu ustalonego.	54
6.1.3	Analiza charakterystyk rozładowania.	57
6.2	Badanie charakterystyki w trybie CP (stała moc).	58
6.2.1	Stan przejściowy.	58
6.2.2	Analiza stanu ustalonego.	58
6.2.3	Analiza charakterystyk rozładowania.	58
6.3	Badanie charakterystyki w trybie CR (stała rezystancja).	59
6.3.1	Stan przejściowy.	59
6.3.2	Analiza stanu ustalonego.	59
6.3.3	Analiza charakterystyk rozładowania.	60
6.4	Badanie charakterystyki w trybie CC (akumulator 12 V – 286 Ah).	61
6.4.1	Stan przejściowy.	61
6.4.2	Analiza stanu ustalonego.	61
6.4.3	Analiza charakterystyk rozładowania.	62

7 Podsumowanie	63
7.1 Podsumowanie celu pracy.	63
7.2 Realizacja i osiągnięcia.	63
Bibliografia	65
Dokumentacja techniczna	69
Spis skrótów i symboli	73
Lista dodatkowych plików	75
Spis rysunków	79
Spis tabel	81

Rozdział 1

Wstęp

W XXI wieku trudno jest wyobrazić sobie życie bez prądu. Urządzenia elektroniczne służą do rozmaitych celów, a każde z nich potrzebuje zasilania. Wiele z nich korzysta z zasilania akumulatorowego. Najczęściej wybieranym rozwiązaniem są akumulatory – kwasowo-ołowiowe, litowo-polimerowe i litowo-jonowe. Ich poprawne użytkowanie oraz monitorowanie stanu technicznego jest warunkiem długotrwałej niezawodności.

Jedną z podstawowych metod oceny parametrów akumulatora jest rozładowanie go przy znanych warunkach obciążeniowych. W tym celu można wykorzystać proste obciążenia rezystancyjne takie jak: zestawy rezistorów lub żarówek. Jednak te rozwiązania są nieelastyczne i nie można ich odpowiednio kontrolować. Obecnie w laboratoriach elektronicznych coraz częściej korzysta się z aktywnych obciążen. Mają one możliwość precyzyjnego i dynamicznego sterowania parametrami rozładowania. Ponadto niektóre modele mogą także rejestrować przebiegi napięcia, prądu i mocy.

Celem tej pracy magisterskiej było zaprojektowanie oraz wykonanie aktywnego obciążenia elektronicznego, przeznaczonego do testowania akumulatorów 12 V, z możliwością pracy w trzech trybach: stałego prądu obciążenia, stałej rezystancji obciążenia i stałej mocy pobieranej. Kolejnym założeniem projektowym był szeroki zakres pracy urządzenia, który powinien był wynosić od około 150 mA, aż do 14 A. Dzięki temu urządzenie mogłoby zostać wykorzystane do pracy zarówno z małymi, jak i dużymi akumulatorami.

W projekcie wykorzystano płytę NUCLEO z mikrokontrolerem z rodziny STM32 – STM32-L152RE. Odpowiada on za realizację algorytmu sterowania przy pomocy pomiarów napięcia i prądu w urządzeniu. Panel sterowania składa się z wyświetlacza LCD, impulsatora z enkoderem, przycisków oraz interfejsu komunikacji UART z komputerem PC. Dzięki takiemu rozwiązaniu można sterować urządzeniem z poziomu komputera, jak i lokalnie. Pozwala to na elastyczność pracy.

Projekt zakładał także zapis danych pomiarowych na karcie SD, w formacie CSV. Umożliwia to późniejszą analizę charakterystyk rozładowania. Ponadto urządzenie ma możliwość pracy w trybie automatycznym – wyłącza się po osiągnięciu warunków końcowych. W tym przypadku takimi kryteriami STOP są: maksymalny czas rozładowania, niskie napięcie na akumulatorze, wysoka temperatura tranzystora obciążenia oraz maksymalny ładunek wyprowadzony z akumulatora.

Zakres pracy obejmował:

- analizę istniejących rozwiązań na rynku i w literaturze,
- opracowanie założeń projektowych oraz schematów blokowych,
- projekt układu elektronicznego w środowisku KiCad,
- oprogramowanie mikrokontrolera opartego na STM32 w środowisku STM32CubeIDE,
- system zapisu danych, panel sterowania i aplikację na PC,
- przeprowadzenie badań z użyciem rzeczywistych akumulatorów.

W części projektowej szczególnie zwrócono uwagę na:

- bezpieczeństwo eksploatacji urządzenia (kryteria zakończenia pracy),
- dokładność pomiarową,
- stabilność termiczną pracy tranzystora.

Rozwiązanie zaproponowane w pracy pozwala na wygodne i precyzyjne testowanie akumulatorów w warunkach laboratoryjnych. Intuicyjny interfejs zapewnia płynne przełączanie trybów oraz dostosowanie parametrów pracy do testowanego ogniwa.

Całość pracy – realizacja sprzętowa i oprogramowania została wykonana samodzielnie przez autora.

Rozdział 2

Analiza tematu

Współczesne urządzenia coraz częściej zasilane są przez akumulatory. Zarówno w zastosowaniu przemysłowym jak i konsumenckim. Począwszy od aut elektrycznych po systemy UPS (zasilania awaryjnego). Wraz ze wzrostem popularności takiego rodzaju zasilania rośnie także wiedza na temat konserwacji, diagnostyki i testów. Kluczowymi urządzeniami pozwalającymi na powtarzalny i kontrolowany proces rozładowania akumulatorów są obciążenia elektroniczne.

Tradycyjne metody testów akumulatorów opierają się o pasywne obciążenia rezystancyjne. Są one proste konstrukcyjnie, lecz nie dają możliwości dynamicznej zmiany warunków pracy, ani nie są w stanie rejestrować parametrów pracy akumulatora. Dlatego laboratoria elektroniczne wykorzystują aktywne obciążenia. Dzięki nim można dokładnie i precyjnie sterować parametrami rozładowania. Pozwalają na zapis danych oraz umożliwiają więcej trybów pomiarowych.

W tym rozdziale zamieszczono przegląd aktualnego stanu wiedzy na temat metod testowania akumulatorów, typów stosowanych obciążzeń, a także elementów elektronicznych wykorzystywanych do ich budowy.

2.1 Obciążenia pasywne i aktywne

Rodzaje obciążzeń elektronicznych jest dużo, ale można podzielić je na dwa główne typy:

- obciążenia pasywne,
- obciążenia aktywne.

Obciążenia pasywne - to zazwyczaj po prostu rezistory dużej mocy, które nie są zmienne dynamicznie w czasie i nie umożliwiają precyjnej regulacji zadanego obciążenia.

Obciążenia aktywne - to obciążenia, które można bardzo precyzyjnie regulować. Różnią się od obciążenia pasywnego tym, że do ich sterowania wykorzystywane są elementy półprzewodnikowe, najczęściej tranzystory. Wraz z większą wydajnością rośnie także złożoność i skomplikowanie układów pomiarowych oraz układów zadających napięcie.

2.1.1 Tryby pracy aktywnego obciążenia

- tryb pracy ze stałym prądem (CC)

Stały prąd (CC) jest najczęściej wykorzystywany trybem pracy obciążenia elektronicznego. W tej konfiguracji układ pobiera zadany prąd niezależnie od wartości napięcia wyjściowego badanego urządzenia testowego (DUT). Element półprzewodnikowy, np. MOSFET, jest sterowaną siłą prądomotoryczną o charakterystyce: $I_d = f(U_{gs})$. Tryb CC stosowany jest głównie do testowania urządzeń o ustalonym napięciu wyjściowym, takich jak akumulatory, baterie czy zasilacze.

- tryb pracy ze stałym napięciem (CV)

W trybie stałego napięcia (CV) utrzymuje stałą różnicę potencjałów na swoich zaciskach niezależnie od prądu wyjściowego urządzenia testowego. Obciążenie dynamicznie zmienia swoją rezystancję, aby osiągnąć zadane napięcie przy dowolnym natężeniu prądu. Zazwyczaj stałym napięciem obciąża się źródła prądowe, takie jak ładowarki lub sterowniki prądu.

- tryb pracy ze stałą rezystancją (CR)

W trybie stałej rezystancji (CR) obciążenie zachowuje się jak programowalny rezistor mocy. Można je wykorzystać do obciążania zarówno źródeł napięciowych, jak i prądowych. Pobierany prąd z urządzenia testowanego jest równy ilorazowi napięcia wyjściowego i zadanej rezystancji. Analogicznie, w przypadku podłączenia do obciążenia źródła prądowego, napięcie na zaciskach jest równe iloczynowi natężenia prądu wyjściowego z DUT i zadanej rezystancji, zgodnie z prawem Ohma. Wykorzystuje się je na przykład do obciążania baterii w celu określenia profilu baterii (charakterystyka działania baterii w różnych warunkach).

- tryb pracy ze stałą mocą (CP)

W trybie stałej mocy (CP) obciążenie reguluje pobierany prąd, aby wchłonąć za-programowany poziom mocy. Tak jak tryb CR, stałą mocą można obciążać zarówno źródła prądowe, jak i napięciowe. Obciążając źródło napięciowe, reguluje pobierany

prąd, dążąc do pochłonięcia zadanej mocy. W przypadku źródła pradowego regluowana jest wartość napięcia na zaciskach. Często wykorzystuje się ten tryb w celu uzyskania informacji o żywotności baterii.

2.2 Przegląd gotowych rozwiązań

Istnieje wiele gotowych rozwiązań aktywnych obciążeń na rynku. Dużo z nich umożliwia kontrolowane testowanie nie tylko akumulatorów, ale też zasilaczy. Posiadają one szeroki zakres parametrów pracy, wysoką dokładność, wiele trybów pomiarowych oraz możliwość zapisu danych. Niektóre z nich umożliwiają komunikację z komputerem PC, co pozwala na zdalne sterowanie parametrami pracy. Do najpopularniejszych producentów takich urządzeń należą firmy: Array, Siglent, BK Precision oraz Rigol. Rozwiązania tych firm są kosztowne – ceny sięgają kilku, bądź kilkunastu tysięcy złotych. Można jednak znaleźć tańsze alternatywy, takie jak: DLT150 - firmy ACS lub BTE-551 - firmy BTE.

Array 3722A - to programowalne obciążenie o mocy 200 W, maksymalnym prądzie obciążenia 20 A oraz zakresie napięcia wejściowego do 200 V. Wyposażone jest w wyświetlacz LCD, klawiaturę numeryczną, impulsator do nastawy parametrów pracy, interfejsy komunikacji RS-232 i USB. Posiada cztery tryby pracy: stały prąd (CC), stałą rezystancję (CR), stałą moc (CP) i stałe napięcie (CV). Jego interfejs jest prosty i wygodny w obsłudze. Parametry te sprawiają, że urządzenie jest wszechstronne i dobrze sprawdza się przy testowaniu baterii jak i zasilaczy. Ze względu na precyzję i stabilność pomiarową jest często spotykane w laboratoriach elektronicznych. Jego cena rynkowa wynosi około 2500 zł.

BTE-551 - to tanie obciążenie elektroniczne o maksymalnym prądzie obciążenia 20 A, mocy 150 W oraz zakresie napięcia wejściowego do 200 V. Posiada wielofunkcyjny wyświetlacz LCD, potencjometry do regulacji prądu obciążenia, 5 różnych zacisków wejściowych. Obciążenie oferuje pomiar napięcia, prądu, mocy, temperatury, posiada także timer do pomiaru czasu pracy urządzenia. Jego cena rynkowa to około 250 zł.

Tabela 2.1: Porównanie modeli obciążeń różnych firm o podobnych parametrach.

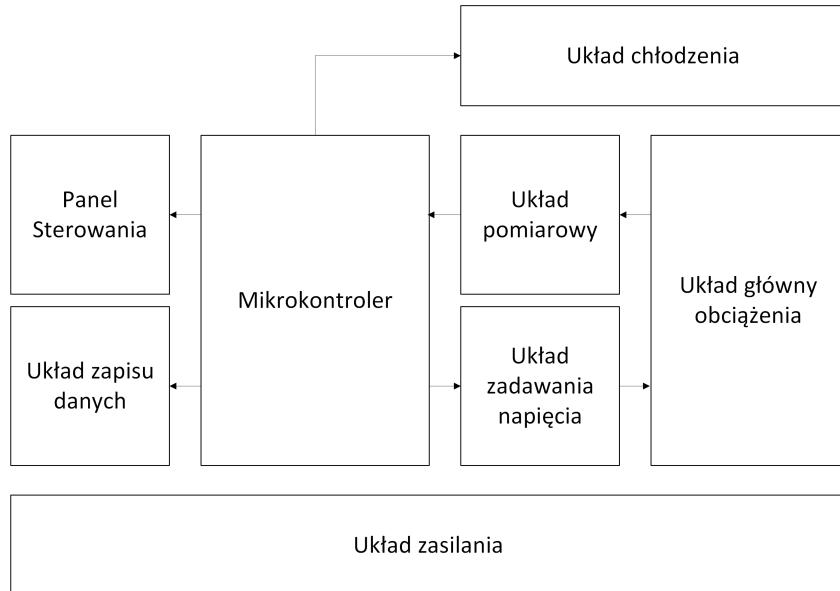
Firma	Model	Maks. Napięcie [V]	Maks. Prąd [A]	Maks. Moc [W]	Tryby Pracy
Array	3722A	200	20	200	CC/CV/CR/CP
Siglent	SDL1020X-E	150	30	200	CC/CV/CR/CP
BK Precision	8600	120	30	150	CC/CV/CR/CP
Rigol	DL3021A	150	40	200	CC/CV/CR/CP
ACS	DLT150	150	20	150	CC/CV/CR/CP
BTE	BTE-551	150	20	200	CC/CV/CR/CP

Niestety tańsze alternatywy cechują się niską jakością pomiarów, brakiem zabezpieczeń przeciążeniowych, wygodnego interfejsu, trybów pomiarowych oraz ograniczoną dokumentacją. Z drugiej strony droższe urządzenia nie będą raczej wykorzystywane w warunkach

domowych z racji na swoją cenę. Ponadto wiele z tych urządzeń oferuje parametry przewyższające potrzeby związane z testowaniem akumulatorów 12 V.

2.3 Ogólny schemat blokowy aktywnego obciążenia

Budowa aktywnego obciążenia wymaga integracji kilku układów, które wspólnie odpowiadają za kontrolę prądu obciążenia, parametrów pracy oraz obsługę interfejsu użytkownika. Na rysunku 2.1 schemat blokowy obciążenia aktywnego.



Rysunek 2.1: Schemat blokowy typowego obciążenia aktywnego.

2.3.1 Układ główny obciążenia

Sercem aktywnego obciążenia elektronicznego jest element odpowiadający za kontrolowany przepływ prądu obciążenia. Najczęściej stosowanym rozwiązaniem jest tranzystor MOSFET pracujący w trybie liniowym. Napięcie (U_{GS}) między bramką (G) a źródłem (S) ustawiane jest tak, aby tranzystor zachowywał się jak regulowana siła prądomotoryczna. W przeciwieństwie do pracy MOSFET-a w trybie PWM – (gdy tranzystor działa jako przełącznik) praca liniowa pozwala na uzyskanie ciągłego przepływu, precyzyjnie sterowanego prądu.

Do takiej pracy bardzo często wykorzystuje się tranzystory N-MOSFET (z kanałem typu n) o niskiej rezystancji RDS_{ON} , typu logic-level. Dzięki małej rezystancji oraz topologii logic-level tranzystor zaczyna przewodzić prąd przy napięciu U_{GS} około 2 V do 3 V oraz otwiera się w pełni, gdy U_{GS} osiąga poziom 5 V. Pozwala to na sterowanie takim komponentem przy pomocy przetwornika cyfrowo-analogowego (DAC) wbudowanego w mikrokontroler. Takie rozwiązanie cechuje się jednak ograniczoną dokładnością, co czyni je mniej opłacalnym w zastosowaniach wymagających precyzji.

2.3.2 Układ pomiarowy

Kolejnym kluczowym układem wchodząącym w skład obciążenia elektronicznego jest układ pomiarowy. Odpowiednia dokładność urządzeń wykorzystanych do pomiaru prądu i napięcia wejściowego decyduje o stabilności i precyzyji obciążenia.

Pomiar napięcia zazwyczaj realizowany jest za pomocą dzielnika napięcia oraz przetwornika analogowo-cyfrowego (ADC). Im większa rozdzielcość ADC, tym większa precyza pomiaru. Lepsze urządzenia korzystają z 24-bitowych przetworników. Przykładem takich komponentów są: MCP3561-E/ST, ADS1155, HX711. Odpowiednio dobrany przetwornik zapewnia precyzyję, stabilność oraz filtrację pomiarów.

Pomiar prądu w układzie często realizowany jest z użyciem przetwornika analogowo-cyfrowego i opornika pomiarowego o niskiej rezystancji (tzw. shunt resistor). Taka konfiguracja pozwala na pomiar spadku napięcia na rezystorze. Po odpowiednim przeliczeniu wartości otrzymanej z ADC można uzyskać wartość prądu przepływającego przez układ.

2.3.3 Układ chłodzenia

W budowie obciążenia aktywnego kolejnym ważnym aspektem jest chłodzenie komponentów. Im większy prąd płynie przez elementy takie jak rezystory mocy i tranzystory, tym więcej wydziela się na nich ciepła. Jest to problem każdego obciążenia. Trzeba pamiętać, że wraz z rosnącą temperaturą w wielu elementach, takich jak rezystory drutowe, rezystancja rośnie. Jednak niektóre komponenty, np. tranzystory, mogą zachowywać się jak NTC (z ang. Negative Temperature Coefficient, czyli elementy o ujemnym współczynniku temperaturowym) — w ich przypadku wzrost temperatury powoduje spadek rezystancji dynamicznej, co prowadzi do zwiększenia przepływu prądu i ryzyka przegrzania. Wiele urządzeń radzi sobie z tym za pomocą czujników temperatury, radiatorów oraz wentylatorów. Do najbardziej znanych czujników należą: DS18B20 oraz termistory. DS18B20 są popularne ze względu na cyfrową transmisję danych (one-wire) i łatwość w integracji z mikrokontrolerami.

2.3.4 Mikrokontroler

Kolejnym ważnym elementem aktywnego obciążenia jest mikrokontroler. Jego zadaniem jest odczyt danych pomiarowych z przetworników, realizacja algorytmu regulacji z pętlą sprzężenia zwrotnego oraz obsługa interfejsu użytkownika. Często realizuje także funkcje związane z komunikacją (RS-232, UART) oraz zapisem danych pomiarowych na zewnętrznych nośnikach pamięci. Ponadto odpowiada także za bezpieczeństwo pracy. Monitoruje przekroczenia parametrów granicznych (prąd, napięcie, temperatura), a w razie potrzeby może przerывать pracę urządzenia lub sterować wentylatorem. Bardzo popularnym wyborem wśród producentów są procesory z rodziny STM32. W ich ofercie znaj-

dują się mikrokontrolery wyposażone w wiele timerów, wbudowanych przetworników ADC i DAC, modułów PWM, interfejsy UART i USB, a także mechanizmy DMA (ang. direct memory access) oraz obsługę przerwań. Dodatkowo, środowisko STM32CubeIDE umożliwia sprawne tworzenie oprogramowania dzięki graficznemu konfiguratorowi peryferiów oraz bibliotekom HAL/LL. Wszystkie te cechy wpływają na wygodę oraz prostotę korzystania z procesorów STM32.

2.3.5 Wymagania pomiarowe i metrologiczne

Proces akwizycji danych pomiarowych w obciążeniach jest bardzo istotnym elementem. Od dokładności pomiaru prądu i napięcia zależy precyzja regulacji oraz jakość całego testu rozładowania. Jest to bardzo ważny aspekt w urządzeniach służących do stabilizacji prądu i testowania akumulatorów.

Przetworniki analogowo-cyfrowe znajdujące się w mikrokontrolerach często są nieefektywne przez swoją ograniczoną rozdzielczość – zwykle 10 lub 12 bitów. Przy napięciu odniesienia na poziomie 2.5 V daje to rozdzielczość rzędu kilku miliwoltów. Dla bardziej precyzyjnych pomiarów powinno wykorzystać się ADC o rozdzielczości minimum 16 bitów. Dzięki temu możliwe jest osiągnięcie znacznie wyższej dokładności.

2.3.6 Panel sterowania

Interfejs użytkownika jest podstawowym elementem funkcjonalnym obciążenia. Pozwala na ustawienie parametrów pracy urządzenia, wybór trybu, uruchamianie i zatrzymywanie testu, a także zapis danych. Dobrze zaprojektowany interfejs zapewnia intuicyjną i wygodną obsługę urządzenia. W komercyjnych urządzeniach najczęściej stosowane są wyświetlacze LCD. Umożliwiają one wyświetlanie aktualnych danych pomiarowych, takich jak napięcie, prąd zadany, prąd zmierzony. Wiele z nich posiada także proste menu, dzięki któremu można płynnie zmieniać parametry pracy. Do obsługi menu urządzenia można na rynku znaleźć rozwiązania wykorzystujące enkodery obrotowe (impulsatory), klawiatury lub przyciski. Impulsatory w połączeniu z przyciskami są bardzo wygodne w obsłudze. Umożliwiają sprawne poruszanie się po interfejsie urządzenia. Klawiatury przydają się do szybkiego i wygodnego wpisywania parametrów pracy. Zajmują jednak stosunkowo dużo miejsca, co zwiększa złożoność konstrukcji urządzenia. W związku z powyższym, dobrym rozwiązaniem jest interfejs komunikacji urządzenia z komputerem PC. Aplikacja komputerowa umożliwiająca wymianę danych z obciążeniem jest dużym udogodnieniem. Dzięki niej możliwe jest wprowadzenie danych z poziomu komputera, które następnie przesyłane są do mikrokontrolera. Dodatkowo interfejs wyświetlacza może zostać uproszczony.

2.3.7 Układ zapisu danych

Wiele urządzeń komercyjnych posiada funkcję zapisu danych pomiarowych. Jest to bardzo ważna kwestia, szczególnie w przypadku pobierania charakterystyk rozładowania akumulatorów, co często jest procesem długotrwałym. Dzięki takim danym możliwa jest wizualizacja charakterystyk w postaci wykresów. W gotowych rozwiązaniach na rynku popularny jest zapis danych do pliku tekstowego CSV. Jest to uniwersalny format zapisu danych. Jest on kompatybilny z wieloma programami takimi jak: Excel czy Libre Office Calc, co umożliwia łatwe przetwarzanie danych. Dane zapisywane są w tabeli, w której kolumny zawierają parametry takie jak: czas, napięcie, prąd obciążenia, moc oraz temperatura.

2.3.8 Podsumowanie i cel pracy

Celem pracy jest zaprojektowanie oraz wykonanie aktywnego obciążenia do testowania akumulatorów 12V poprzez ich kontrolowane rozładowanie według zadanych parametrów. Urządzenie ma posiadać trzy tryby pracy:

1. stały prąd płynący przez obciążenie,
2. stałą rezystancję obciążenia,
3. stałą moc obciążenia.

Maksymalny prąd obciążenia powinien mieć wartość kilkunastu amper. Urządzenie ma wykorzystywać mikrokontroler NUCLEO-L152RE, ma posiadać ekran LCD oraz impulsator do zadawania parametrów pracy. Zmiana parametrów powinna być również możliwa za pomocą łącznika szeregowego. Urządzenie ma mierzyć prąd i napięcie obciążenia.

Test akumulatora polega na zmierzeniu jego charakterystyki w czasie rozładowania (wybranym trybem) przez określony czas lub do spełnienia określonego warunku końcowego. Urządzenie powinno zapisywać zbieraną charakterystykę. Mierzona charakterystyka i inne parametry pracy powinny być zapisywane w pliku CSV na karcie SD.

Rozdział 3

Realizacja sprzętowa - Koncepcja i schemat elektryczny

W poniższym rozdziale przedstawiony został sposób realizacji sprzętowej zaprojektowanego aktywnego obciążenia elektronicznego. Zgodnie z celem projektu urządzenie zostało zaprojektowane jako obciążenie dedykowane do testowania akumulatorów 12 V w zakresie prądowym: 150 mA – 14 A, z możliwością rejestracji danych na nośniku pamięci, komunikacji z komputerem PC oraz możliwością autonomicznej pracy.

Projekt został podzielony na kilka bloków funkcjonalnych. Każdy z nich odpowiada za inny aspekt pracy obciążenia. W kolejnych podrozdziałach opisano szczegółowo budowę i działanie kolejnych bloków:

- układ główny obciążenia,
- układ pomiaru napięcia wejściowego
- układ pomiaru natężenia prądu obciążenia,
- układ zadawania napięcia sterującego,
- układ zasilania i napięcia referencyjnego,
- czujniki temperatury,
- panel sterowania (LCD, impulsator, przyciski),
- układ zapisu danych na zewnętrznym nośniku pamięci.

Urządzenie zostało zaprojektowane z wykorzystaniem oprogramowania KiCad 9.0 umożliwiającego tworzenie schematów elektrycznych oraz projektów płytki PCB. Dodatkowo, do analizy wybranych fragmentów układu skorzystano z pomocy symulatora typu SPICE: Ltspice.

Opisane rozwiązania mają na celu zapewnienie pełnej funkcjonalności urządzenia zgodnie z założonymi wymaganiami, przy zachowaniu niskich kosztów budowy.

3.1 Układ główny obciążenia

Elementy wykorzystane do budowy podukładu (Rys. 3.1):

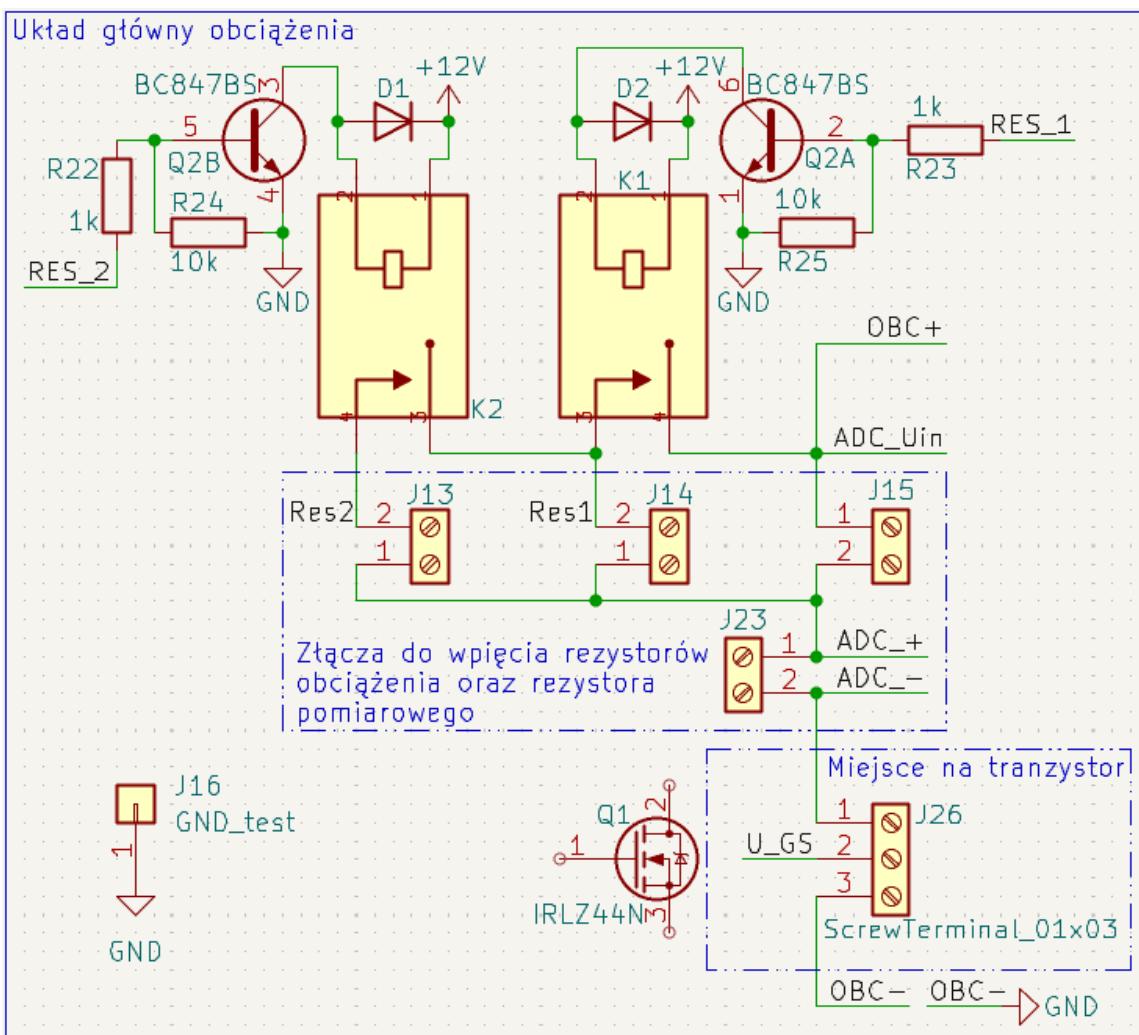
- tranzystor - IRLZ44N (Q1, J26 - terminal do podpięcia elektrod tranzystora),
- rezystory mocy (J13, J14, J15 - terminale do podpięcia końcówek rezystorów),
- rezistor pomiarowy (J23),
- przekaźniki PRE29 (K1, K2),
- tranzystory BC847 (Q2A, Q2B),
- rezystory SMD 0805,
- diody.

3.1.1 Tranzystor IRLZ44N

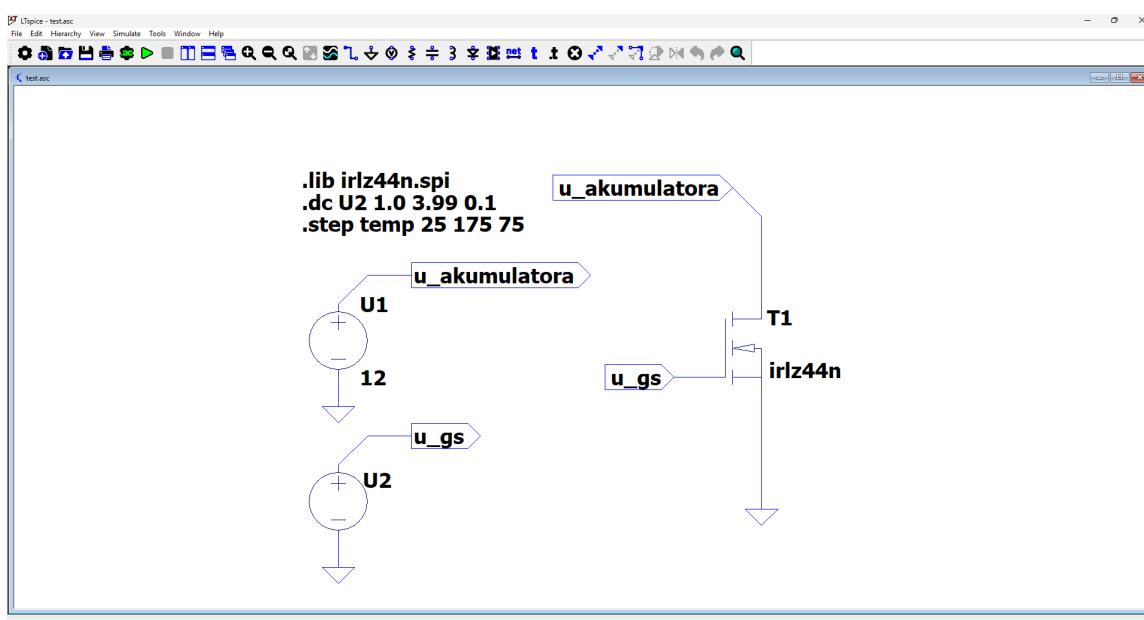
Najważniejszym elementem układu obciążenia jest tranzystor IRLZ44N typu MOSFET (Q1). Dzięki niemu możliwe jest odpowiednie ograniczanie prądu przepływającego przez pętle pomiędzy zaciskiem wejściowym a wyjściowym obciążenia. Sterując napięciem U_{GS} zmienia się prąd, który przepływa między drenem a źródłem, zgodnie z zasadą działania MOSFET-a. Tranzystor, jaki został wybrany do tego projektu, to IRLZ44N. Jest to tranzystor o następujących właściwościach:

- typ: N-MOSFET, logic-level,
- maksymalny prąd: 33 A,
- maksymalna moc: 110 W,
- napięcie progowe: około 2 V, pełne przewodzenie przy 5 V.

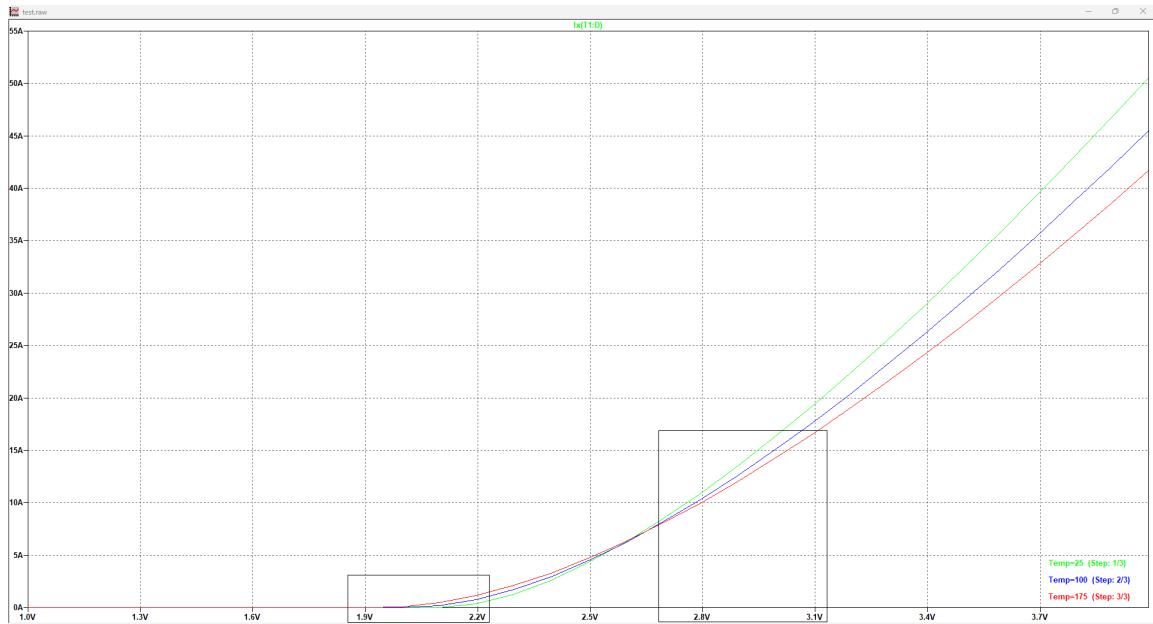
Tranzystor IRLZ44N został wybrany ze względu na to, że spełnia wszystkie założenia projektu. Maksymalny prąd przepływający przez obciążenie powinien wynosić 14 A. Wybrany MOSFET zapewnia około dwukrotnie większy prąd, który stale przepływa przez dren-źródło. Ponadto jest to tranzystor typu logic-level, co oznacza, że można nim sterować napięciem od ok. 2 V do 5 V. Wybrany model zaczyna przewodzić w okolicy 2 V (Rys. 3.4). Pokazuje to symulacja w programie LTspice. Model tranzystora wykorzystanego w aplikacji został ściągnięty ze strony producenta. Symulacja została pokazana na rysunkach: 3.2, 3.3, 3.4 i 3.5.



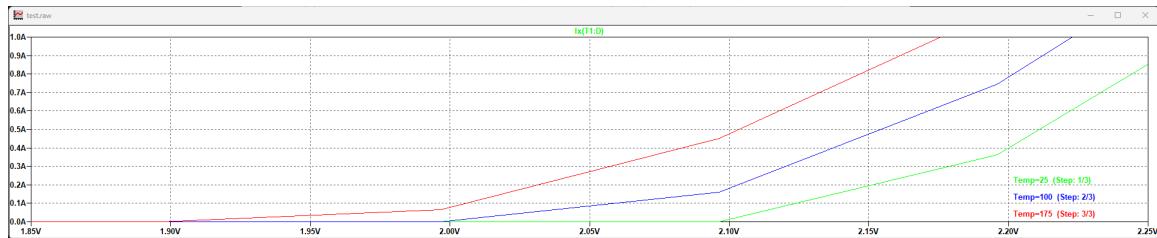
Rysunek 3.1: Fragment schematu elektrycznego przedstawiający układ główny obciążenia.



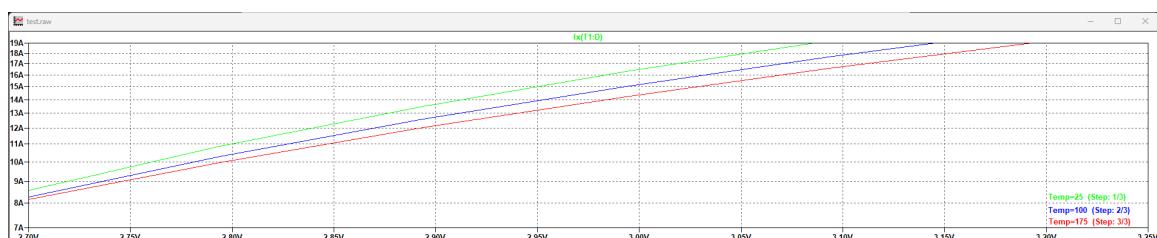
Rysunek 3.2: Zrzut ekranu przedstawiający symulację pracy tranzystora IRLZ44N.



Rysunek 3.3: Wykres przedstawiający napięcie tranzystora IRLZ44N bramka-źródło od prądu na drenie.



Rysunek 3.4: Fragment wykresu $U_{GS}(I_D)$ tranzystora w początku pracy obciążenia.



Rysunek 3.5: Fragment wykresu $U_{GS}(I_D)$ tranzystora w granicach planowanej pracy obciążenia.

3.1.2 Rezystory mocy

W układzie obciążenia znajdują się także rezystory mocy (J13, J14, J15). Zostały one dodane po to, żeby mogła się na nich rozpraszać część mocy pobieranej z akumulatora. Odciążają one tranzystor i nie pozwalają, aby moc przekroczyła na nim 110 W. Układ ten został zaprojektowany z trzech rezystorów mocy, które są podłączane i odłączane w zależności od zadanych przez użytkownika parametrów (prądu obciążenia lub rezystancji obciążenia lub pobieranej mocy). Wraz ze wzrostem prądu obciążenia rezystancja na wejściu powinna maleć, dlatego oporniki są połączone między sobą równolegle. Wspólnie tworzą one trzy różne rezystancje: 1.5Ω , 0.857143Ω oraz 0.6Ω . Wyliczenia dotyczące wydzielanej mocy w układzie, na rezystorach i na tranzystorze można znaleźć w tabeli Tab.3.1.

Tabela 3.1: Tabela przedstawiająca obliczenia rezystancji obciążenia.

U_{we} R_p	12 V 0.1Ω			1.5Ω		0.857Ω		0.6Ω		
R_{obc}		$I[A]$	$P_{tot}[W]$	$P_{rp}[W]$	$P_{rob}[W]$	$P_{tranz}[W]$	$P_{rob}[W]$	$P_{tranz}[W]$	$P_{rob}[W]$	$P_{tr}[W]$
Zakres 1	1	12	0.1	1.5	10.4	0.86	11.04	0.6	11.3	
	2	24	0.4	6.0	17.6	3.43	20.17	2.4	21.2	
	3	36	0.9	13.5	21.6	7.71	27.39	5.4	29.7	
	4	48	1.6	24.0	22.4	13.71	32.69	9.6	36.8	
	5	60	2.5	37.5	20.0	21.43	36.07	15.0	42.5	
Zakres 2	6	72	3.6	54.0	14.4	30.86	37.54	21.6	46.8	
	7	84	4.9	73.5	5.6	42.00	37.10	29.4	49.7	
	8	96	6.4	96.0	*-6, 4	54.86	34.74	38.4	51.2	
	9	108	8.1	121.5	*-21.6	69.43	30.47	48.6	51.3	
Zakres 3	10	120	10.0	150.0	*-40.0	85.71	24.29	60.0	50.0	
	11	132	12.1	181.5	*-61.6	103.71	16.19	72.6	47.3	
	12	144	14.4	216.0	*-86.4	123.43	6.17	86.4	43.2	
	13	156	16.9	253.5	*-114.4	144.86	*-5.76	101.4	37.7	
	14	168	19.6	294.0	*-145.6	168.00	*-19.60	117.6	30.8	

* wartości pomijalne – niefizyczne warunki pracy.

W wierszach, gdzie P_{tranz} przyjmuje wartość ujemną, suma spadków napięć na rezystorach (R_p i R_{obc}) przekracza napięcie zasilania 12 V. Oznacza to, że tranzystor musiałby oddawać moc, co jest niemożliwe przy pracy liniowej. Te punkty wykraczają poza dopuszczalny zakres dla danej kombinacji rezystorów i zostały oznaczone jako niefizyczne.

Jak można zauważyć, rezystancje zostały dobrane tak, aby nie przekroczyć mocy krytycznej zarówno na tranzystorze, jak i opornikach. Każdy z nich ma ograniczenie 50 W. W zakresie nr 1, kiedy podłączony do układu jest tylko jeden opornik, może się rozproszyć na nim tylko 50 W, natomiast w zakresach nr 2 i 3 moc wydzielana na rezystancji obciążenia może wzrosnąć do kolejno: 100 i 150 W.

Obliczenia potrzebne do stworzenia tabel:

Moc całkowita w układzie (P_{tot}):

$$P_{tot} = U \cdot I[W] \quad (3.1)$$

Moc na rezystorze pomiarowym:

$$P_{rp} = R_p \cdot I^2[W] \quad (3.2)$$

Moc na rezystancji obciążenia:

$$P_{rob} = R_{rob} \cdot I^2[W] \quad (3.3)$$

Moc na tranzystorze:

$$P_{tranz} = P_{tot} - (P_{rp} + P_{rob})[W] \quad (3.4)$$

gdzie: U - Napięcie, R_p - Rezystancja opornika pomiarowego i R_{rob} - Rezystancja obciążenia.

3.1.3 Przekaźniki, tranzystory, rezystory SMD i diody

Do sterowania dołączaniem i odłączaniem rezystorów mocy wykorzystano przekaźniki PRE29 (K1 i K2), których cewki zasilane są napięciem 12 V (Rys. 3.1). Z uwagi na brak możliwości bezpośredniego sterowania przekaźnikami z mikrokontrolera, zastosowano proste przełączniki tranzystorowe zbudowane na bazie BC847 (Q2A i Q2B) z odpowiednimi rezystorami bazowymi (R22, R23, R24 i R25). Do zacisków cewek przekaźników równolegle zostały dołączone diody zabezpieczające (D1 i D2), których zadaniem jest ochrona przed przepięciami powstałymi w wyniku dołączania i odłączania rezystorów mocy. Takie zabezpieczenie chroni układ przełączników tranzystorowych i mikrokontroler przed uszkodzeniem.

3.2 Układy pomiarowe

Elementy wykorzystane do budowy układu:

- przetworniki analogowo-cyfrowe - MCP3561-E/ST (U4 i U5),
- wzmacniacze operacyjne - LMV932 (U8, U10 i U2B),
- rezystory SMD 0805,
- kondensatory SMD 0805,
- cewki i koraliki ferrytowe SMD 0805.

Układy pomiarowe zostały zaprojektowane w oparciu o przetworniki analogowo-cyfrowe – MCP3561-E/ST (U4 i U5). Są to 24-bitowe ADC typu sigma-delta, rail-to-rail wykorzystujące magistralę SPI do komunikacji. Nadpróbkowanie, kształtowanie szumu, filtr cyfrowy i decymacje przetwornika, w połączeniu z rozdzielczością 24-bit, pozwalają na uzyskanie wysokiej precyzji pomiaru przy stosunkowo niskim koszcie komponentu.

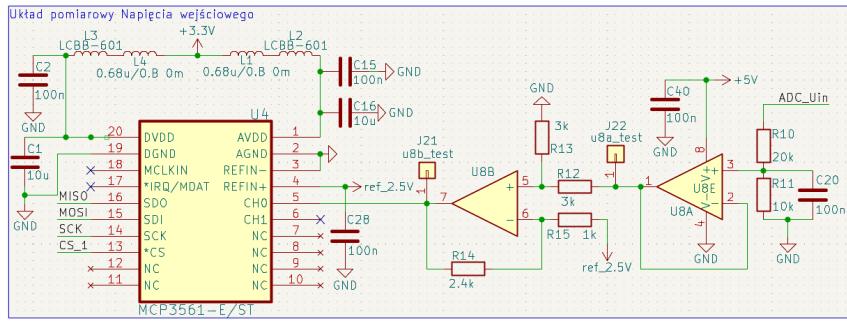
Układ pomiarowy składa się także z rezystorów i wzmacniaczy operacyjnych. Napięcie referencyjne dla przetwornika A/C ma wartość 2.5 V, w związku z tym zakres przetwarzania przetwornika A/C wynosi od 0 V do 2.5 V. Dzięki zaprojektowanym dzielnikom napięcia współpracującym ze wzmacniaczami, sygnał wejściowy przenoszony jest z zakresu od 10.5 V do 15 V do zakresu przetwarzania przetwornika A/C od 0 do 2.5 V (Rys. 3.7). W skład całego urządzenia wchodzą dwa układy pomiarowe:

- układ pomiarowy napięcia wejściowego (Rys. 3.6),
- układ pomiarowy natężenia prądu obciążenia (Rys. 3.8).

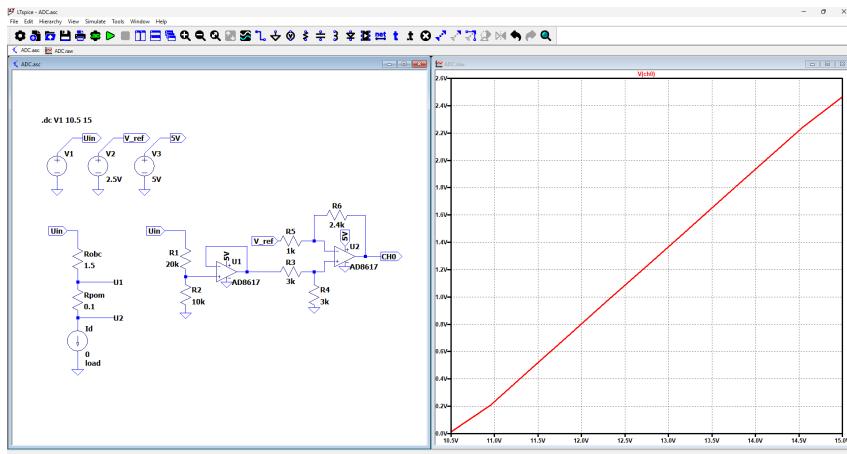
Wejście przetwornika (U4) zostało skonfigurowane do pracy z jednym wejściem z sygnałem niesymetrycznym względem masy. Napięcie do wejścia CH0 podawane jest przez dzielik napięcia złożony z rezystorów R10 i R11 (o wartościach odpowiednio $20\text{ k}\Omega$ i $10\text{ k}\Omega$) a następnie wzmacniacz operacyjny (U8B) w konfiguracji wzmacniacza różnicowego. Zakres napięć wejściowych od 10.5 V do 15 V transformowany jest przez dzielik napięcia, wtórnik napięciowy (U8A) i wzmacniacz operacyjny do zakresu od 0 V do 2.5 V.

Cewki, koraliki ferrytowe (L1, L2, L3, L4) oraz kondensatory (C1, C2, C15, C16) zostały wykorzystane w celu filtracji napięcia zasilającego przetwornik analogowo-cyfrowy.

Układ pomiaru prądu obciążenia został zaprojektowany w bardzo podobny sposób do tego powyżej. Opiera się o przetwornik analogowo-cyfrowy (U5) pracujący w konfiguracji niesymetrycznej lub różnicowej. Projekt przewiduje możliwość wyboru trybu pomiarowego. Pozwala to na porównywanie jakości sygnału w obu konfiguracjach. Sygnał mierzony z rezystora pomiarowego, przez który przepływa prąd obciążenia, został podzielony na dwie alternatywne ścieżki.



Rysunek 3.6: Fragment schematu przedstawiający układ pomiarowy napięcia wejściowego.



Rysunek 3.7: Symulacja pracy układu skalującego napięcie (z 10.5/15 V na 0/2.5 V) oraz wykres z programu LTspice.

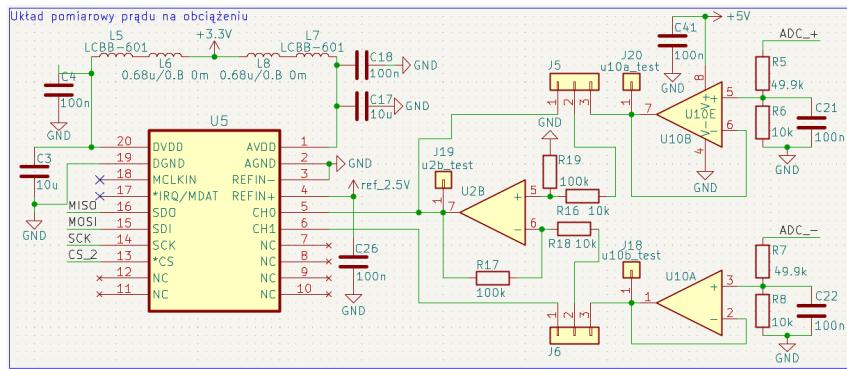
Pierwszy tor pomiarowy (niesymetryczny) wykorzystuje dzielniki napięcia (R5-R6 i R8-R9), wzmacniacze operacyjne: U10A i U10B w konfiguracji wtórnika napięciowego, oraz U2B w konfiguracji wzmacniacza różnicowego. Taki sygnał jest wprowadzany na wejście CH0 do U5.

Drugi tor pomiarowy (różnicowy) pomija wzmacniacz różnicowy (U2B). W tym przypadku sygnały z obu końców rezystora pomiarowego są transformowane przez dzielniki napięcia (R5-R6 i R8-R9), wtórniki napięciowe, a następnie wprowadzane bezpośrednio do dwóch kanałów ADC (CH0 i CH1).

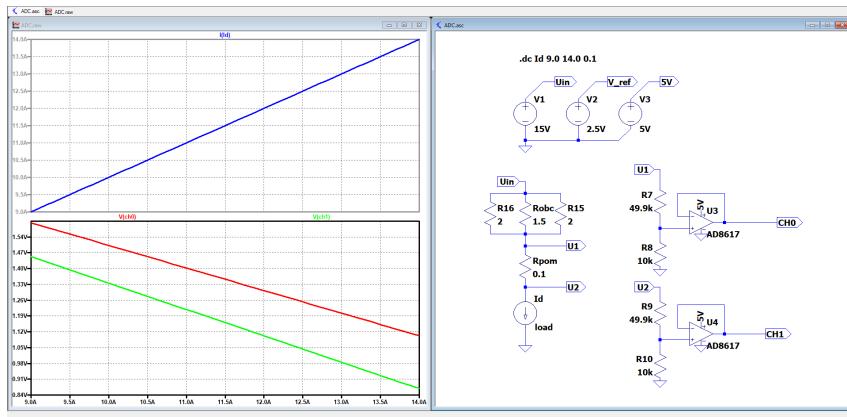
W obu ścieżkach wykorzystano kondensatory odsprzęgające (C21, C22) oraz zastosowano wtórniki napięciowe (U10A, U10B). Przełączanie między ścieżkami możliwe jest przy pomocy złączy J5 i J6.

W trakcie pierwszych testów urządzenia zdecydowano się na wykorzystanie przetwornika analogowo-cyfrowego w konfiguracji różnicowej.

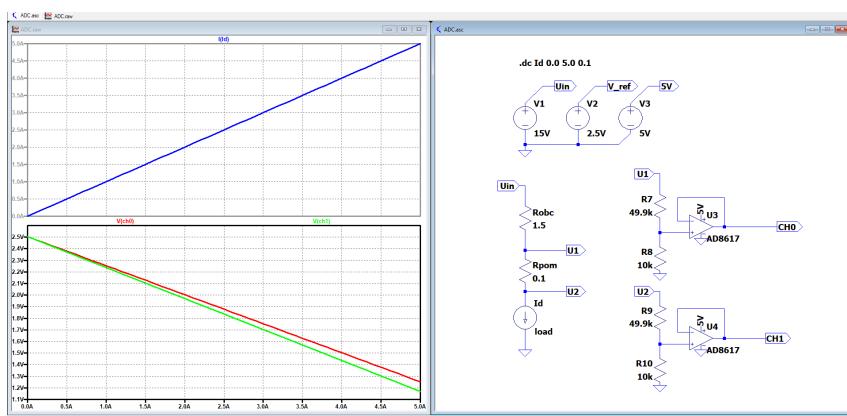
Jak można zauważyć na Rys. 3.9 i Rys. 3.10 napięcia z torów pomiarowych są przeskalowane do napięć od 0 V do 2.5 V.



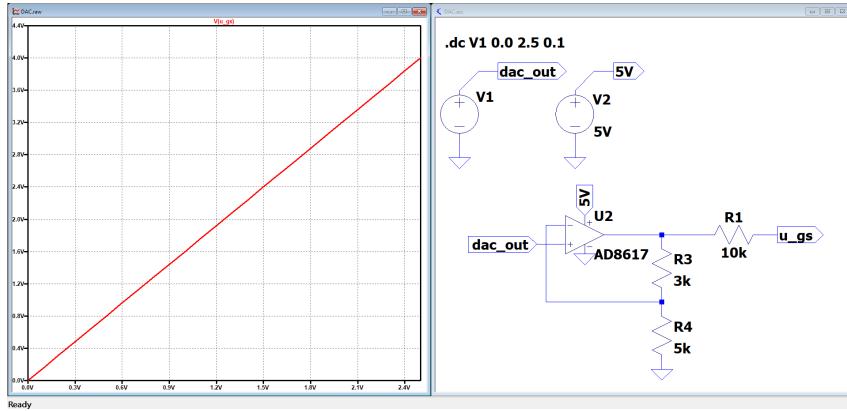
Rysunek 3.8: Fragment schematu przedstawiający układ pomiarowy natężenia prądu w obciążeniu.



Rysunek 3.9: Symulacja układu pomiarowego dla maksymalnego prądu i napięcia wejściowego.



Rysunek 3.10: Symulacja układu pomiarowego dla minimalnego prądu i maksymalnego napięcia wejściowego.



Rysunek 3.11: Symulacja pracy LMV932 w konfiguracji wzmacniacza nieodwracającego oraz wykres zależności napięcia wyjściowego od wejściowego wzmacniacza.

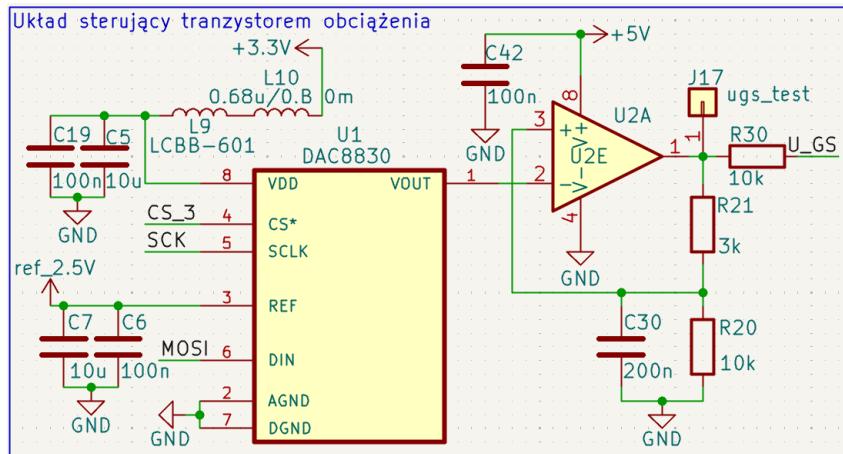
3.3 Układ zadawania napięcia sterującego

Elementy wchodzące w skład układu wytwarzającego napięcie sterujące Q1:

- przetwornik cyfrowo-analogowy - DAC8830 (U1),
- wzmacniacz operacyjny - LMV932 (U2A),
- rezystory SMD 0805,
- kondensatory SMD 0805,
- cewki i koraliki ferrytowe SMD 0805.

Układ zadający napięcie sterujące został wykonany w oparciu o przetwornik cyfrowo-analogowy DAC8830 (U1) oraz wzmacniacz operacyjny (U2A) pracujący w konfiguracji wzmacniacza nieodwracalnego (Rys. 3.12). Celem tego układu jest precyzyjne sterowanie napięciem bramki tranzystora MOSFET odpowiedzialnego za regulację prądu obciążenia.

DAC8830 to 16-bitowy przetwornik cyfrowo-analogowy wykorzystujący magistralę SPI do zadawania napięcia wyjściowego. W celu filtracji napięcia zasilającego użyto cewek, koralików ferrytowych oraz kondensatorów (L9, L10, C5, C19). Do transformacji napięcia sygnału wychodzącego z U1 wykorzystano wzmacniacz operacyjny w konfiguracji wzmacniacza nieodwracającego (U2A). Dzięki temu zakres napięcia sterującego bramką tranzystora został przetransformowany z zakresu od 0 V do 2.5 V na zakres od 0 V do 4 V (Rys. 3.11).



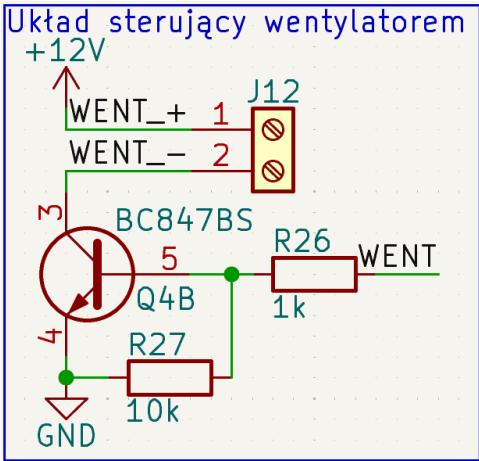
Rysunek 3.12: Fragment schematu elektrycznego układu sterującego tranzystorem obciążenia.

3.4 Układ chłodzenia i czujniki temperatury

Elementy wchodzące w skład układu i pomiaru temperatury to:

- radiator 12 cm x 12 cm,
- wentylator 12 V, 12 cm x 12 cm,
- czujniki DS18B20,
- tranzystor BC847BS (Q4B),
- rezystory SMD 0805.

Jednym z założeń projektowych była możliwość obciążania akumulatorów prądem rzędu kilkunastu amperów (14 A). W takiej sytuacji na rezystorach oraz tranzystorze obciążających wydziela się duża moc, co powoduje nagrzewanie się do wysokiej temperatury. Aby uniknąć nadmiernego przegrzewania się urządzenia, co mogłoby doprowadzić do zniszczenia komponentów, oraz dla zapewnienia bezpiecznej temperatury zdecydowano się na umieszczenie elementów obciążających na radiatorze, chłodzonym przez wentylator. Dodatkowo zastosowano czujniki temperatury tranzystora oraz otoczenia, w celu monitorowania poziomu temperatury. Wybrane czujniki to popularne układy DS18B20 w obudowie TO-92. Czujniki te mierzą temperaturę z rozdzielcością 12-bitów, komunikują się z mikrokontrolerem za pomocą one-wire, co ułatwia prowadzenie przewodów. Jeden z czujników został umieszczony bezpośrednio na tranzystorze obciążenia. Drugi zbiera dane na temat temperatury z otoczenia. Takie rozmieszczenie komponentów pozwala na precyzyjne monitorowanie stanu termicznego układu.



Rysunek 3.13: Fragment schematu elektrycznego przedstawiającego układ sterowania wentylatora.

Do zamocowania na płytce czujników temperatury oraz wentylatora wykorzystano terminale (J4, J7, J12).

Do sterowania wentylatorem zasilanym z napięcia 12 V wykorzystano tranzystor BC847BS (Q4B) w połączeniu z rezystorami bazowymi (R26 i R27).

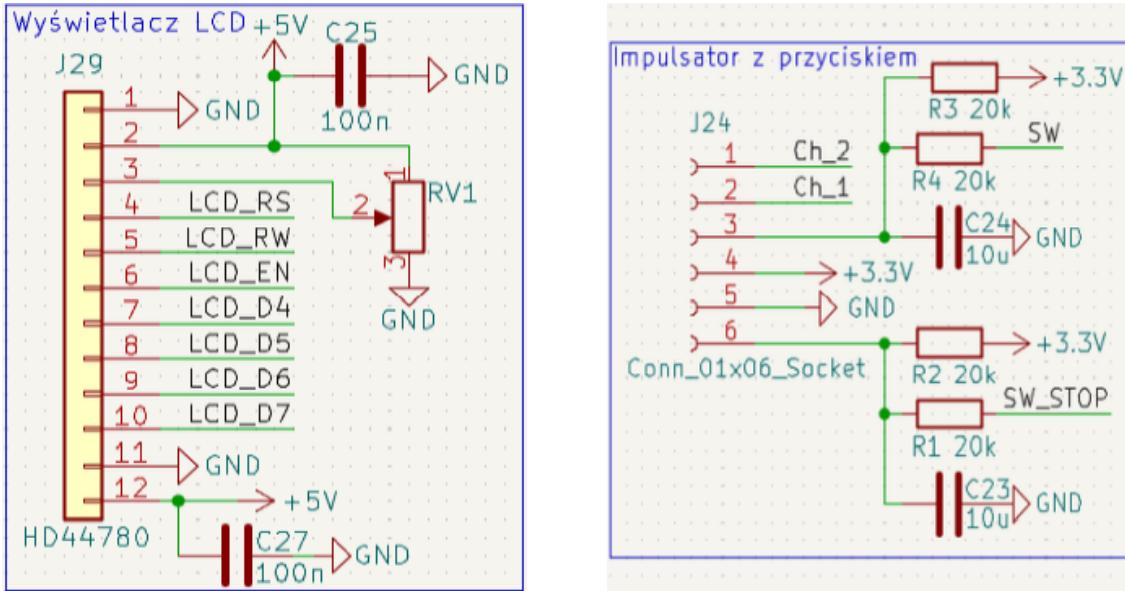
3.5 Panel sterowania

Elementy wchodzące w skład układu:

- wyświetlacz LCD 2x16, justPi (J29 - złącze do wpięcia wyświetlacza),
- potencjometr $10\text{ k}\Omega$ (RV1),
- impulsator z przyciskiem, Iduino SE055 (J24 - złącze do wpięcia impulsatora),
- przycisk monostabilny (J24),
- rezystory SMD 0805,
- kondensatory SMD 0805.

Układ panelu sterowania projektowanego obciążenia (Rys. 3.14) jest potrzebny do autonomicznej pracy urządzenia. Głównym elementem tego podukładu jest wyświetlacz LCD firmy justPi. Dzięki wbudowanemu sterownikowi HDD44780 możliwa jest bardzo prosta implementacja algorytmu wyświetlania komunikatów w dwóch liniach, mających łącznie trzydzieści dwa pola znakowe.

Z wyświetlaczem współpracuje układ złożony z gotowego impulsatora z wbudowanym przyciskiem firmy Iduino – (model SE055) oraz przycisku monostabilnego. Dodatkowo blisko przycisków zastosowano rezystory pull-up – R3 i R2 ($20\text{ k}\Omega$) oraz kondensatory

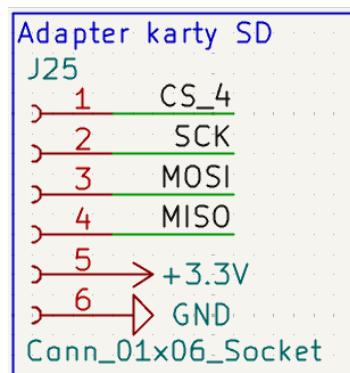


Rysunek 3.14: Fragmenty schematu elektrycznego przedstawiające układ panelu sterowania użytkownika.

odsprzęgające zakłócenia sygnału (C23 i C24). Dzięki takiemu połączeniu powyższych komponentów możliwe jest stworzenie prostego w obsłudze panelu sterowania, pozwalającego na łatwą zmianę parametrów pracy.

3.6 Układ zapisu danych na zewnętrznym nośniku pamięci – karta SD

Do zapisu danych wykorzystano gotowy moduł adaptera kart micro-SD wykorzystujący magistralę SPI do komunikacji. Charakteryzuje się niską ceną i prostotą obsługi. Ponadto występuje on w wersji z osobną płytą PCB, co pozwala na wyprowadzenie go poza obudowę urządzenia dla łatwego dostępu (J25 - złącze do wpięcia adaptera). Fragment schematu z adapterem znajduje się na rysunku 3.15.



Rysunek 3.15: Fragment schematu elektrycznego przedstawiający układ zapisu danych na zewnętrznym nośniku pamięci.

3.7 Układ zasilania i napięcia referencyjnego

Elementy układu zasilającego i napięcia referencyjnego:

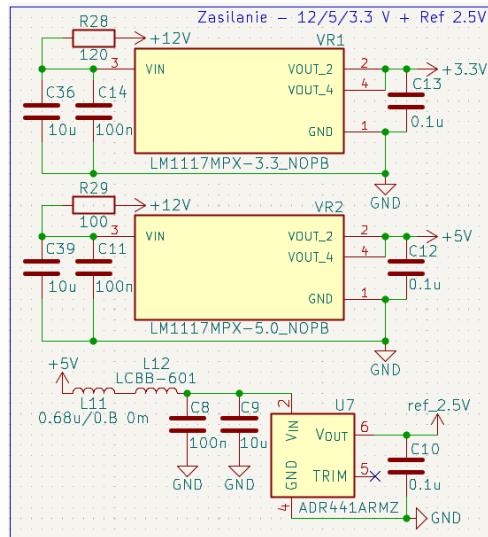
- zasilacz 12 V,
- stabilizator napięcia 5 V – LM117MPX-5.0NOPB (VR2),
- stabilizator napięcia 3.3 V – LM117MPX-3.3NOPB (VR1),
- napięcie referencyjne ADR441ARMZ (U7),
- rezystory SMD 1206,
- kondensatory SMD 0805,
- cewki i koraliki ferrytowe SMD 0805.

Układ zasilający (Rys. 3.16) odpowiedzialny jest za dostarczenie odpowiednich napięć do całego obciążenia, w tym przetworników ADC i DAC, wzmacniaczy operacyjnych, tranzystorów sterujących, przekaźników, wyświetlacza, wentylatora oraz czujników temperatury.

Założono, że całe urządzenie będzie zasilane z zewnętrznego napięcia 12 V DC. Napięcie to jest dostarczane do wentylatora, przekaźników oraz stanowi napięcie wejściowe stabilizatorów liniowych.

W celu uzyskania napięć 5 V oraz 3.3 V zastosowano stabilizatory liniowe z serii LM117PX o niskim poziomie szumów. Napięcie 5 V (VR2) używane jest do zasilenia wzmacniaczy operacyjnych, wyświetlacza LCD oraz stanowi wejście do napięcia referencyjnego. Dodatkowo stabilizator 3.3 V (VR1) wykorzystany został do dostarczenia napięcia wejściowego dla przetworników ADC i DAC, diody wyświetlacza LCD oraz czujników temperatury. Zastosowano również kondensatory filtrujące umieszczone blisko pinów stabilizatorów w celu tłumienia zakłóceń (C11, C12, C13, C14, C36, C39).

Źródło napięcia ADR441ARMZ (U7) zapewnia wysoką dokładność pomiarów przetworników. Jest to precyzyjny układ o stałym napięciu wyjściowym 2.5 V, przeznaczony do zastosowań pomiarowych. Charakteryzuje się on niskim poziomem szumów. Układ dostarcza napięcia zarówno do przetworników analogowo-cyfrowych, jak i cyfrowo-analogowego. Ponadto w celu filtracji napięcia zasilającego układu zastosowano kondensatory, cewki oraz koraliki ferrytowe (L11, L12, C8, C9 i C10).



Rysunek 3.16: Fragment schematu elektrycznego przedstawiającego układ zasilania.

Rozdział 4

Realizacja sprzętowa - Projekt PCB i budowa

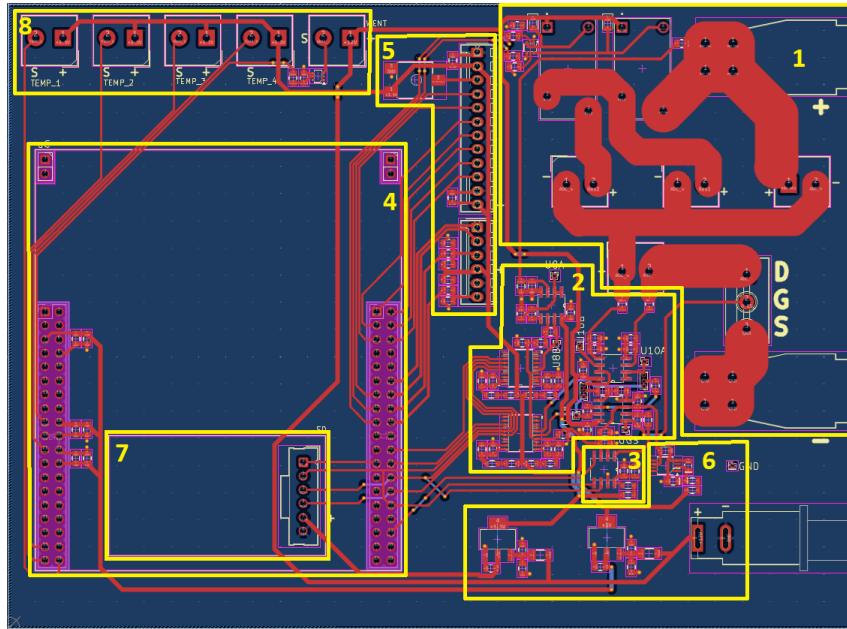
Poniższy rozdział opisuje projektowanie dedykowanej płytki drukowanej (PCB) oraz fizyczną realizację projektu. Projekt płytki został wykonany w programie KiCad 9.0. Układ zaprojektowano na dwuwarstwowej płytce o wymiarach 155.6 mm x 114.85 mm. Celem konstrukcji było zapewnienie stabilnej pracy obciążenia w pełnym zakresie prądowym od 150 mA do 14 A oraz zgodność z wymaganiami pomiarowymi i funkcjonalnymi przedstawionymi w poprzednich rozdziałach.

Struktura funkcjonalna płytki.

Płytkę drukowaną (Rys. 4.1) można podzielić na sekcje:

1. układ obciążenia,
2. układ pomiarowy,
3. układ zadawania napięcia sterującego,
4. moduł sterowania,
5. panel sterowania,
6. układ zasilania i referencji,
7. układ zapisu danych,
8. czujniki temperatury i wentylator.

Każda z powyższych sekcji otrzymała miejsca na płytce PCB z uwzględnieniem grubości ścieżek oraz zachowaniem odpowiednich odległości.



Rysunek 4.1: Projekt płytki drukowanej z podziałem na sekcje.

4.1 Układ obciążenia - PCB

Układ obciążenia został zaprojektowany zgodnie z wymaganiami elektrycznymi urządzenia. Według założeń projektowych, przez tę część płytki może płynąć prąd o natężeniu nawet 14 A, co wymagało odpowiedniego podejścia przy projektowaniu.

Zastosowano krótkie, ale szerokie ścieżki połączeniowe, aby zminimalizować ich rezystancję oraz ograniczyć nagrzewanie. W miejscach, gdzie przez pojedynczy tor prądowy płynie prąd rzędu kilkunastu amper, szerokość ścieżki została dobrana zgodnie z wytycznymi normy IPC-2221 dla grubości miedzi – 35 μm . Wejścia elementów mocy (terminale śrubowe – J13, J14, J15, J23 i J26) zostały umieszczone jak najbliżej siebie, tak aby zminimalizować długość ścieżki prądowej. W układzie tym znajdują się także przekaźniki sterujące rezystancją obciążenia (K1, K2). Wszystkie elementy zostały umieszczone na górnej warstwie PCB (top).

Do połączenia między rezystorami i tranzystorem obciążenia, a terminalami użyto przewody z linką miedzianą o przekroju 2 mm^2 . Zapewnia to odpowiednią przewodność oraz umożliwia bezpieczne przewodzenie natężenia rzędu kilkunastu amper. Zastosowanie przewodu zamiast ścieżki PCB wynika z konieczności umiejscowienia elementów obciążenia na radiatorze. Duża moc pochłaniana w układzie mocy skutkuje występowaniem wysokich temperatur. Przewody zostały odpowiednio przylutowane, a następnie zabezpieczone na końcówkach koszulkami termokurczliwymi.

4.2 Układ pomiarowy i zadawania napięcia sterującego – PCB.

Całość sekcji pomiarowej znajduje się w centralnej części płytki PCB między układem obciążenia a układem sterującym urządzeniem. Taka lokalizacja pozwala na zachowanie krótkich torów sygnałowych oraz ograniczenie wpływu zakłóceń na pomiary pochodzących z sekcji mocy.

Do układu pomiarowego doprowadzono linie sygnałowe z rezystora pomiarowego (J23) oraz z dzielnika napięcia wejściowego. Są one oddalone od sygnałów logicznych magistrali SPI oraz przełączających rezystancje obciążenia. Sekcje zawierają przetworniki analogowo-cyfrowe (U4, U5), cyfrowo-analogowe (U1) oraz wzmacniacze operacyjne (U2, U8, U10). Filtry zasilania oraz kondensatory odsprzęgające zostały umieszczone jak najbliżej pinów napięć wejściowych komponentów. Ponadto tory sygnałowe magistrali SPI zostały zaprojektowane z zachowaniem pewnej odległości od mikrokontrolera, w celu uniknięcia niepotrzebnych zakłóceń pochodzących z linii sygnałowych w torach pomiarowych.

4.3 Moduł sterowania.

Sekcja modułu sterowania obciążenia została umieszczona jak najdalej od elementów mocy, ograniczając wpływ zakłóceń na mikrokontroler oraz linie komunikacyjne.

W celu integracji druku PCB i mikrokontrolera zaprojektowano dedykowany shield, oparty na konektorach żeńskich (J1, J2, J27 i J28), kompatybilnych z rozstawem i układem pinów płytki Nucleo. Taki montaż zapewnia stabilną pozycję mikrokontrolera na płytce drukowanej oraz możliwość jego odłączenia lub wymiany, na przykład w przypadku awarii lub potrzeby programowania poza urządzeniem.

Dla linii sygnałowych ze wszystkich komponentów współpracujących z systemem mikroprocesorowym zapewniono odpowiednie prowadzenie, unikając krzyżowania się ścieżek.

4.4 Panel Sterowania.

Elementy z sekcji panelu sterowania zostały rozmieszczone w górnej części płytki drukowanej. Umożliwia to łatwe prowadzenie przewodów do frontowego panelu urządzenia. Na warstwie górnej płytki PCB znalazły się linie sygnałowe dla wyświetlacza LCD, impulsatora z wbudowanym przyciskiem oraz osobnego przycisku monostabilnego.

Linie sygnałowe łączące mikrokontroler z interfejsem użytkownika zostały poprowadzone w uporządkowany sposób z zachowaniem separacji od torów pomiarowych i komunikacji SPI. W miejscach narażonych na zakłócenia zdecydowano się na zamieszczenie kondensatorów odsprzęgających oraz rezistorów pull-up. Są one położone bardzo blisko li-

nii urządzeń, z którymi współpracują. Zapewniają odfiltrowanie zasilania, poprawną pracę enkodera oraz eliminację drgań styków.

Złącza użyte do wyprowadzeń linii sygnałowych wyświetlacza, impulsatora oraz przycisku monostabilnego to konektory męskie. Do połączenia elementów funkcjonalnych wykorzystano żeńsko-żeńskie przewody sygnałowe firmy justPi. Zapewniają solidne połączenie między drukiem PCB a komponentami interfejsu.

Cała sekcja została zaprojektowana z myślą o intuicyjnym montażu oraz niezawodnej pracy działania podukładu.

4.5 Układ zasilania i referencji.

W zgodzie z założeniami i wybranymi w rozdziale komponentami (Rozdział „Realizacja sprzętowa”), na płytce PCB powinny znaleźć się trzy tory zasilania - 12 V, 5 V oraz 3.3 V. Każda z linii doprowadzającej określone napięcie do elementów charakteryzuje się szerokością 30.0 mils (0.762 mm), co przy standardowej grubości ścieżek miedzianych: 35 μm pozwala na bezpieczne przewodzenie prądów do około 1 A do 1.5 A, bez znacznego wzrostu temperatury miedzi w liniach zasilających.

Napięcie 12 V doprowadzone zostało z zewnętrznego zasilacza o wydajności prądowej: 1.25 A. Elementem wejściowym łączącym druk PCB i zasilacz jest gniazdo zasilania DC. Od niego tory napięcia wejściowego rozchodzą się kolejno do stabilizatorów liniowych, wentylatora oraz przekaźników.

Najważniejszym elementem tego boku funkcjonalnego jest napięcie referencyjne (U7). Zostało ono umieszczone jak najbliżej przetworników zasilania wejściowego oraz przetworników, do których doprowadzony zostaje sygnał wyjściowy tego elementu. Dla zachowania niezakłóconej pracy napięcia referencyjnego wszystkie ścieżki wyjściowe mają szerokość 20.0 mils (0.508 mm).

4.6 Układ zapisu danych.

Blok funkcjonalny zapisu danych wyprowadzony został na zewnątrz obudowy. Przymocowano go za pomocą śrub M3 6 mm oraz nakrętek o średnicy 3 mm. Sygnały zasilające oraz magistrali SPI doprowadzone zostały za pomocą przewodów sygnałowych firmy justPi. Umieszczenie adaptera kart microSD na bocznej ściance urządzenia pozwala na szybki dostęp do nośnika pamięci, bez konieczności zdejmowania obudowy.

4.7 Czujniki temperatury i wentylator.

Dla odpowiedniego monitorowania stanu temperatury czujniki pomiarowe znajdują się wewnętrz i zewnętrz urządzenia. Jeden z nich umieszczony został bezpośrednio na

tranzystorze, za pomocą małej płytki miedzianej z otworem o średnicy 3 mm, która została przymocowana za pomocą nakrętki M3 do śruby trzymającej tranzystor na radiatorze. Czujnik pomiaru temperatury otoczenia został umieszczony na obudowie urządzenia.

4.8 Obudowa.

Całość urządzenia – płytka PCB oraz przewody sygnałowe zostały umieszczone wewnątrz poliwęglanowej obudowy uniwersalnej o wymiarach: 150 x 200 x 55 mm.

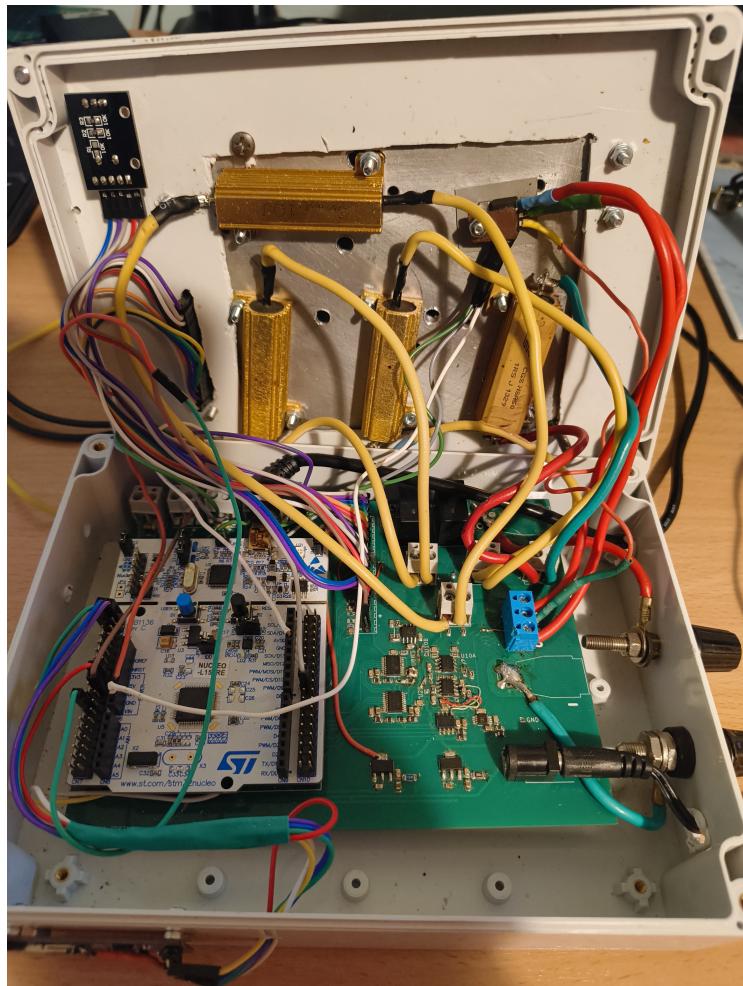
W celu montażu poszczególnych elementów w obudowie wykonano szereg otworów:

- kwadratowy otwór pod radiator oraz wyjścia sygnałowe wyświetlacza,
- okrągłe otwory, między innymi:
 - mocowanie radiatora (ϕ 4 mm),
 - przycisk monostabilny (ϕ 3 mm),
 - impulsator (ϕ 5 mm),
 - adapter kart microSD (3 x ϕ 3 mm),
 - mocowanie wyświetlacza LCD (2 x ϕ 3 mm),
 - gniazda bananowe (2 x ϕ 5 mm),
 - wejścia przewodów USB i zasilania (2 x ϕ 5 mm).

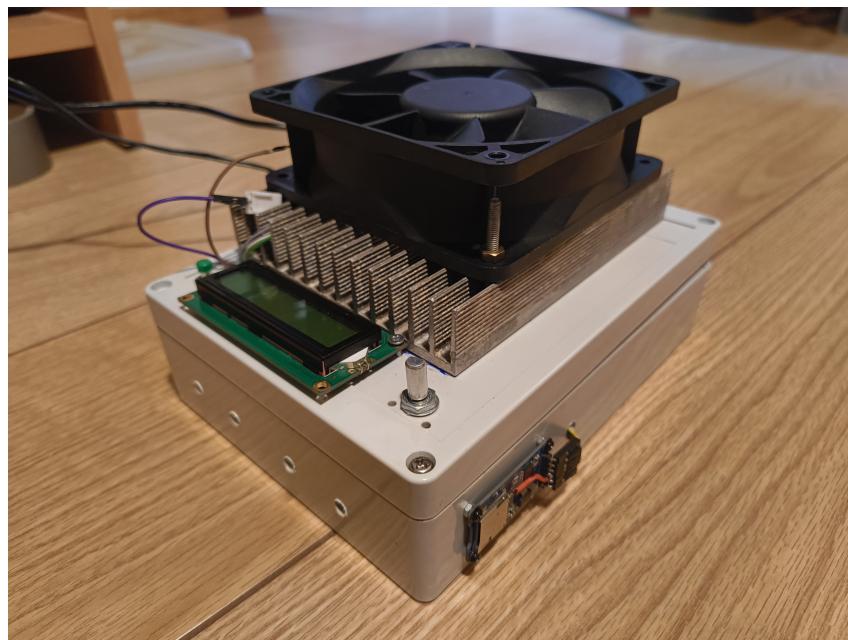
Ponadto dla zapewnienia odpowiedniej cyrkulacji powietrza wewnątrz urządzenia wykonano:

- 6 otworów wentylacyjnych (ϕ 5 mm) w radiatorze,
- 4 otwory wentylacyjne (ϕ 5 mm) w obudowie.

Wszystkie elementy zostały zamocowane za pomocą odpowiednich śrub, podkładek oraz nakrętek. Na rysunkach 4.2 i 4.3 zaprezentowano gotowe, w pełni złożone aktywne obciążenie.



Rysunek 4.2: Urządzenie po otwarciu obudowy.



Rysunek 4.3: Złożony projekt aktywnego obciążenia.

Rozdział 5

Oprogramowanie mikrokontrolera i aplikacji

Poniższy rozdział obejmuje tematykę realizacji oprogramowania mikrokontrolera, projektowanego obciążenia aktywnego. Program został stworzony z wykorzystaniem środowiska STM32CubeIDE 1.18.0, które umożliwia graficzną konfigurację peryferiów i jest dedykowanym środowiskiem służącym do projektowania układów z mikrokontrolerami STM32. Kod programu został napisany w języku C przy pomocy wbudowanych bibliotek HAL (Hardware Abstraction Layer). W dalszej części opisana zostanie także realizacja aplikacji służącej do komunikacji między mikrokontrolerem a komputerem PC w celu zadawania parametrów i wymiany danych.

Urządzenie pracuje całkowicie autonomicznie, jego działanie opiera się na działaniu w głównej pętli while(1) oraz reagowaniu na przerwania asynchroniczne. Do zadań realizowanych przez zaprojektowany system mikroprocesorowy należy:

- pomiary napięcia i prądu z przetworników analogowo-cyfrowych,
- regulacja napięcia zadającego sterowanie, zależna od trybu pracy (CC, CR, CP),
- sterowanie przetwornikiem cyfrowo-analogowym,
- zarządzanie i funkcjonalność interfejsu użytkownika,
- zapis danych na zewnętrzny nośnik pamięci (microSD),
- komunikacja z komputerem PC za pomocą interfejsu UART,
- nadzór nad bezpieczeństwem pracy (temperatura, napięcie, czas, ładunek wyprowadzony – obliczany na podstawie czasu i prądu).

Program został podzielony na części - biblioteki - odpowiedzialne za funkcje składowe. Takie podejście umożliwia proste i przejrzyste modyfikowanie kodu źródłowego.

W kolejnych podrozdziałach szczegółowo omówiono strukturę programu, działanie poszczególnych bloków funkcjonalnych kodu oraz implementację logiki sterującej obciążeniem.

5.1 Struktura programu.

Moduł sterowania obciążeniem aktywnym podzielony jest na podmoduły, które odpowiadają za poprawną pracę urządzenia. Takie podejście do pisania kodu nazywane jest architekturą modularną. Cechuje się tym, że kod staje się czytelniejszy, a wszystkie funkcje, obsługujące na przykład panel sterowania, mieszczą się w jednej bibliotece.

W ramach tego projektu stworzone zostały następujące moduły:

- główna pętla programu (main.c),
- struktura danych (mystruct.c),
- obsługa regulatora i kompensacji temperaturowej (regulator.c),
- panel sterowania,
- zapis danych na zewnętrznym nośniku pamięci,
- komunikacja z komputerem PC.

Wszystkie powyższe elementy tworzą cały kod zapewniający poprawną pracę obciążenia aktywnego.

5.2 Główna pętla programu.

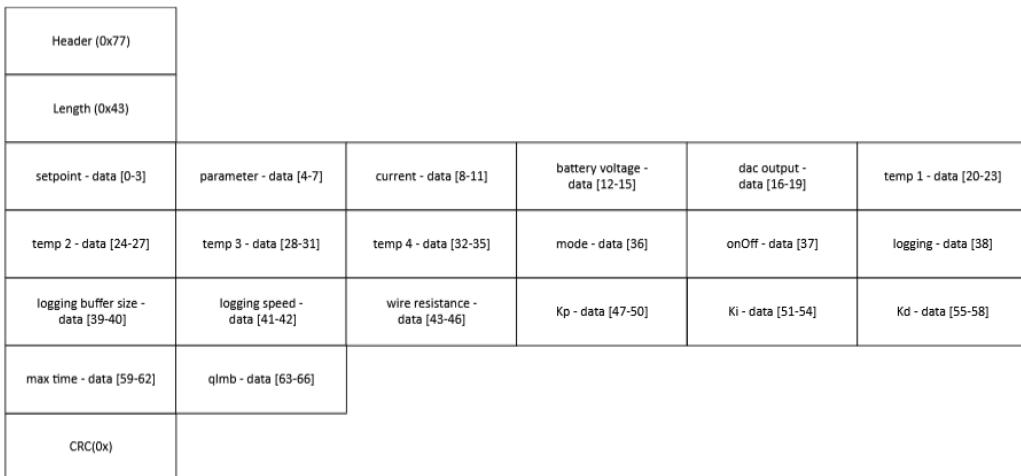
Kod w pliku main.c został odpowiednio uporządkowany. Na początku inicjalizowane są wykorzystywane biblioteki takie jak: "stdio.h", "string.h" oraz "math.h", zmienne globalne oraz peryferia. Następnie urządzenie komunikuje się z elementami takimi jak: przetworniki ADC w celu odpowiedniego ustawienia ich do pracy. Następnie inicjalizowane są timery i obsługa przerwań.

Główna pętla programu obsługuje bardzo ważne funkcje. Dotyczą one kryteriów stop obciążenia, pomiaru temperatury, aktualizacji danych w interfejsie użytkownika, zapisu danych pomiarowych na karcie microSD oraz obsługi komunikacji z komputerem PC za pośrednictwem UART.

W następnej części pliku main.c mieści się implementacja najważniejszej roli logiki sterującej - pomiar parametrów dotyczących napięcia wejściowego i prądu obciążenia, przeliczenie parametru zależnego od wybranego trybu pracy (CC - prąd, CR - rezystancja, CP - moc) oraz wywołanie funkcji regulatora. Wszystkie te elementy mieszczą się w przerwaniu, dla zapewnienia stałego czasu próbkowania.

5.3 Struktura DataStruct.

Bardzo ważnym elementem projektu jest struktura danych DataStruct, o którą opiera się działanie całego programu. Dane zamknięte są w strukturze złożonej z nagłówka – pełniącego funkcję identyfikatora mikrokontrolera, długości danych – zmiennej przechowującej długość całej struktury, tablicy danych – mieszącą się tam prawie wszystkie dane pomiarowe oraz ustawienia mikrokontrolera. Ostatnim elementem struktury jest suma kontrolna.



Rysunek 5.1: Graficzne przedstawienie struktury danych wykorzystywanej w programie obciążenia.

Jak przedstawiono na Rys. 5.1 dane w strukturze zapisane są jako pojedyncze bajty, które następnie są składane, na przykład do formatu float w celu dalszego przetwarzania.

Kod źródłowy dotyczący struktury danych znajduje się w module o nazwie “my-struct.c”. Mieści się tam implementacja globalnej struktury (Rys. 5.2), funkcje *set()* i *get()* (Rys. 5.3) oraz *define* (Rys. 5.4), pomagające w zarządzaniu danymi urządzenia.

```

18 typedef struct {
19     uint8_t header;
20     uint8_t length;
21     uint8_t data[67];
22     uint32_t crc;
23 } DataStruct;
24
25

```

Rysunek 5.2: Fragment kodu definiującego strukturę danych z Rys.5.1.

```

192 uint8_t getMode(void) {
193     return *(uint8_t*)&myData.data[36];
194 }
195
196 void setMode(uint8_t mode) {
197     memcpy(&myData.data[36], &mode, sizeof(uint8_t));
198 }
199
200

```

Rysunek 5.3: Przykładowe funkcje *set()* i *get()*.

```
87  
88  
89 #define wire_Res *(float*)myData.data[43]  
90 #define Kp *(float*)&myData.data[47]  
91 #define Ki *(float*)&myData.data[51]  
92 #define Kd *(float*)&myData.data[55]  
93 #define timeStop *(float*)&myData.data[59]  
94  
95
```

Rysunek 5.4: Przykładowe funkcje *define*.

5.4 Pomiary napięcia i prądu obciążenia.

Jednym z najważniejszych bloków funkcjonalnych obciążenia jest pomiar napięcia akumulatora, natężenia prądu obciążenia oraz temperatury. W poprzednich rozdziałach przedstawiono budowę torów sygnałowych oraz implementację sprzętową – w tej sekcji omówiono obsługę programową czujników.

Pomiar napięcia wejściowego oraz natężenia prądu realizowany jest z wykorzystaniem precyzyjnych przetworników komunikujących się z mikrokontrolerem za pośrednictwem magistrali SPI. Aby zapewnić stabilne i dokładne pomiary, konieczna jest odpowiednia inicjalizacja pracy ADC, która polega na ustawieniu szeregu parametrów zapisanych wewnętrznych rejestrach MCP3561. Szczegółowe informacje na temat treści rejestrów można znaleźć w dokumentacji katalogowej komponentu [3].

Przetworniki zostały ustawione w tryb pracy CONTINUOUS MODE. Oznacza to, że konwersja analogowo-cyfrowa odbywa się w sposób ciągły – każdy odczyt z rejestru zawierającego aktualną wartość pomiaru inicjuje kolejny cykl pracy. Dzięki temu możliwe jest bardzo szybkie odczytywanie napięcia lub natężenia prądu bez przymusu wyzwalania pojedynczych pomiarów.

Kolejnym ważnym parametrem jest OVERSAMPLING RATIO (OSR) – współczynnik nadpróbkowania. Zastosowanie wysokiego OSR pozwala na redukcję szumu. W projekcie zdecydowano się na nastawę najwyższej możliwej wartości współczynnika (98304). Dzięki temu pomiary mają największą rozdzielczość.

Jednym z najważniejszych ustawień przetwornika jest nastawa punktu odczytu danych oraz jego odniesienia. W przypadku pomiaru napięcia ADC ustawiony jest w konfiguracji niesymetrycznej. Oznacza to, że przetwornik mierzy napięcie na wejściu CH0 względem masy urządzenia. Przetwornik odpowiedzialny za pomiar prądu obciążenia ustawiony jest w konfiguracji różnicowej. MCP3561 mierzy różnicę napięć pomiędzy wejściami CH0 i CH1.

Do zmiany rejestrów przetwornika potrzebne jest wysłanie zestawu dwóch bajtów. Jeden z nich zawiera informację o kodzie wewnętrznym MCP3561 (opisanym na obudowie - „01”), numerze rejestru (np. „0001”) oraz kodzie zapisu („10”). Drugi bajt zawiera treść modyfikowanego rejestru. Inicjalizacja przetwornika została przedstawiona na rysunku 5.5.

Zbieranie danych pomiarowych (Rys. 5.6) odbywa się z wykorzystaniem tzw. „szybkiej komendy”. Jest to specjalny bajt wysyłany do przetwornika w celu odczytu danych z rejestru zawierającego wartość pomiaru. Wartość odsyłana przez MCP3561 (w postaci 3 bajtów) jest składana i przechowywana w zmiennej „raw_data”. Aby uzyskać prawdziwą wartość pomiaru, należy taką surową informację przeliczyć. W tym celu, dla znanych wartości parametrów, odczytano informację pochodzącą z przetwornika w maksymalnych i minimalnych warunkach pracy. Następnie obliczone zostały współczynniki równania liniowego powstałego z dwóch punktów pomiarowych. Dzięki takiej metodzie możliwe jest poprawne dopasowanie odczytu z MCP3561 do prawdziwych wartości prądu czy napięcia. W przypadku pomiaru natężenia prądu (Rys. 5.7) przeliczenie wartości, z surowych danych pochodzących z przetwornika na realną wartość, zostało wykonane dla każdej z trzech wartości rezystancji obciążenia (Rys. 5.8). Pomiar napięcia uwzględnia rezystancje kabli pomiarowych, gdyż dla dużych prądów oporność kabli pomiarowych jest na tyle duża, że spadek napięcia na przewodach jest niepomijalny. Aby to skorygować i poprawnie odczytywać napięcie akumulatora, do napięcia mierzonego dodawany jest iloczyn prądu i rezystancji kabli pomiarowych wpisywanych przez użytkownika jako parametr. Na podstawie informacji dotyczącej wielkości prądu ustawiane są rezystory obciążenia, zgodnie z Tab. 3.1.

Ponadto w funkcji odczytu prądu, (Rys. 5.8 - linia 128), realizowana jest dyskretna całka prądu. Iloczyn zmierzonego natężenia i okresu próbkowania (50 ms) dodawany jest do zmiennej *qlmb*, w której akumulowany jest całkowity ładunek wyprowadzony z akumulatora. Pozwala to na bieżące monitorowanie energii oddanej przez akumulator.

```

162= void NCP3561T_1_Init(GPIO_TypeDef *Port, uint16_t Pin) {
163
164     uint8_t config0_cmnd[2] = {0b01000110, 0b01100011};
165     uint8_t config1_cmnd[2] = {0b01001010, 0b00111100};
166     uint8_t config2_cmnd[2] = {0b01001110, 0b10001011}; // 110
167     uint8_t config3_cmnd[2] = {0b01010010, 0b11000000};
168     uint8_t MUX_cmnd[2] = {0b01011010, 0b00001000}; // 0b00001000
169     uint8_t IRQ_cmnd[2] = {0b01010110, 0b00000110};
170     uint8_t FAST_cmnd = 0b01101000;
171
172     HAL_Delay(100);
173
174     HAL_GPIO_WritePin(Port, Pin, GPIO_PIN_RESET);
175     HAL_SPI_Transmit(&hspi2, &config0_cmnd[0], 2, HAL_MAX_DELAY);
176     HAL_GPIO_WritePin(Port, Pin, GPIO_PIN_SET);
177
178     HAL_GPIO_WritePin(Port, Pin, GPIO_PIN_RESET);
179     HAL_SPI_Transmit(&hspi2, &config1_cmnd[0], 2, HAL_MAX_DELAY);
180     HAL_GPIO_WritePin(Port, Pin, GPIO_PIN_SET);
181
182     HAL_GPIO_WritePin(Port, Pin, GPIO_PIN_RESET);
183     HAL_SPI_Transmit(&hspi2, &config2_cmnd[0], 2, HAL_MAX_DELAY);
184     HAL_GPIO_WritePin(Port, Pin, GPIO_PIN_SET);
185
186     HAL_GPIO_WritePin(Port, Pin, GPIO_PIN_RESET);
187     HAL_SPI_Transmit(&hspi2, &config3_cmnd[0], 2, HAL_MAX_DELAY);
188     HAL_GPIO_WritePin(Port, Pin, GPIO_PIN_SET);
189
190     HAL_GPIO_WritePin(Port, Pin, GPIO_PIN_RESET);
191     HAL_SPI_Transmit(&hspi2, &IRQ_cmnd[0], 2, HAL_MAX_DELAY);
192     HAL_GPIO_WritePin(Port, Pin, GPIO_PIN_SET);
193
194     HAL_GPIO_WritePin(Port, Pin, GPIO_PIN_RESET);
195     HAL_SPI_Transmit(&hspi2, &MUX_cmnd[0], 2, HAL_MAX_DELAY);
196     HAL_GPIO_WritePin(Port, Pin, GPIO_PIN_SET);
197
198     HAL_GPIO_WritePin(Port, Pin, GPIO_PIN_RESET);
199     HAL_SPI_Transmit(&hspi2, &FAST_cmnd, 1, HAL_MAX_DELAY);
200     HAL_GPIO_WritePin(Port, Pin, GPIO_PIN_SET);
201
202     uint8_t byte = 0;
203     uint8_t tx_byte = {0b01000101};
204     HAL_GPIO_WritePin(Port, Pin, GPIO_PIN_RESET);
205     HAL_SPI_TransmitReceive(&hspi2, &tx_byte, &byte, 1, HAL_MAX_DELAY);
206     HAL_GPIO_WritePin(Port, Pin, GPIO_PIN_SET);

```

Rysunek 5.5: Fragment kodu pliku „mystruct.c” odpowiadający za inicjalizację przetwornika ADC.

```

137= void setBatteryVoltage(void) {
138
139     static const float a = 1.4978/2913086;
140     static const float b = 11.506 - (1838100*a);
141     uint8_t Voltage[3];
142     uint8_t readVoltage = 0b10000001;
143     HAL_GPIO_WritePin(CS_1_GPIO_Port, CS_1_Pin, GPIO_PIN_RESET);
144     HAL_SPI_Transmit(&hspi2, &readVoltage, 1, HAL_MAX_DELAY);
145     HAL_SPI_Receive(&hspi2, &Voltage[0], 3, HAL_MAX_DELAY);
146     HAL_GPIO_WritePin(CS_1_GPIO_Port, CS_1_Pin, GPIO_PIN_SET);
147     int32_t raw_data = (Voltage[0] << 16) | (Voltage[1] << 8) | Voltage[2];
148     float battery_voltage = 0;
149     float wire_resistance = *(float*)&myData.data[43];
150
151     if (getOnOff() != 0) {
152         battery_voltage = (((float)raw_data * a) + b) + getCurrent()*wire_resistance;
153     }
154     else {
155         battery_voltage = (((float)raw_data * a) + b);
156     }
157     memcpy(&myData.data[12], &battery_voltage, sizeof(float));
158 }
159

```

Rysunek 5.6: Fragment pliku „mystruct.c” przedstawiający realizacje pomiaru napięcia wejściowego.

```

65 void setCurrent(void) {
66
67     uint8_t Voltage[3];
68     uint8_t readVoltage = 0b01000001;
69     HAL_GPIO_WritePin(CS_2_GPIO_Port, CS_2_Pin, GPIO_PIN_RESET);
70     HAL_SPI_Transmit(&hspi2, &readVoltage, 1, HAL_MAX_DELAY);
71     HAL_SPI_Receive(&hspi2, &Voltage[0], 3, HAL_MAX_DELAY);
72     HAL_GPIO_WritePin(CS_2_GPIO_Port, CS_2_Pin, GPIO_PIN_SET);
73     int32_t raw_data = (Voltage[0] << 16) | (Voltage[1] << 8) | Voltage[2];
74
75     float current = 0.0;
76     float setpoint = 0.0;
77     if (getMode() == 'P')
78         setpoint = getSetpoint() / getBatteryVoltage();
79     else if (getMode() == 'R')
80         setpoint = getBatteryVoltage() / getSetpoint();
81     else
82         setpoint = getSetpoint();
83
84     if (setpoint >= 0.0f && setpoint <= 5.0) {
85         current = ((float)raw_data*(4.74f/294092.3333f) + (0.17-26377.6667*(4.74f/294092.3333f)));
86     }
87     else if (setpoint > 5.0f && setpoint <= 10.0) {
88         current = ((float)raw_data*(5.086667f/418925) + (5.066667f-333558.333f*(5.086667f/418925)));
89     }
90     else if (setpoint > 10.0f && setpoint <= 15.0) {
91         current = ((float)raw_data*(3.44f/461250.333f) + (11.04f-855380.333f*(3.44f/461250.333f)));
92     }
}

```

Rysunek 5.7: Fragment pliku „mystruct.c” przedstawiający odczyt natężenia prądu.

```

94     float value = 0.0;
95
96     if (getMode() == 'C') {
97
98         value = getSetpoint();
99     } else if (getMode() == 'R') {
100
101         value = getBatteryVoltage() / getSetpoint();
102     } else if (getMode() == 'P') {
103
104         value = getSetpoint() / getBatteryVoltage();
105     }
106
107     if (value <= 5.0f) {
108
109         HAL_GPIO_WritePin(RES_1_GPIO_Port, RES_1_Pin, GPIO_PIN_RESET);
110         HAL_GPIO_WritePin(RES_2_GPIO_Port, RES_2_Pin, GPIO_PIN_RESET);
111     }
112     else if (value > 5.0f && value <= 11.0f) {
113
114         HAL_GPIO_WritePin(RES_1_GPIO_Port, RES_1_Pin, GPIO_PIN_SET);
115         HAL_GPIO_WritePin(RES_2_GPIO_Port, RES_2_Pin, GPIO_PIN_RESET);
116     }
117     else if (value > 11.0f && value <= 15.0f) {
118
119         HAL_GPIO_WritePin(RES_1_GPIO_Port, RES_1_Pin, GPIO_PIN_SET);
120         HAL_GPIO_WritePin(RES_2_GPIO_Port, RES_2_Pin, GPIO_PIN_SET);
121     }
122
123     if (getOnOff() != 1)
124         current = 0.0f;
125
126     qlmb += current*0.05f;
127
128

```

Rysunek 5.8: Fragment funkcji *setCurrent()* przedstawiający realizacje zmiany rezystancji obciążenia oraz aktualizację parametru ładunku wyprowadzonego z akumulatora (linia 128).

5.5 Pomiar temperatury.

Komunikacja z czujnikami DS18B20 odbywa się za pośrednictwem protokołu one-wire, który pozwala na transmisję danych oraz zasilanie urządzeń przy użyciu tylko jednej linii sygnałowej. Czujniki zostały wykorzystane ze względu na prostotę protokołu oraz szeroką dostępność DS18B20.

Obsługa protokołu one-wire została w pełni zaimplementowana programowo w module „mystruct.c”. Do realizacji transmisji wykorzystywane są funkcje (Rys. 5.9):

- *wire_reset()* - inicjalizacja magistrali one-wire,
- *read_bit()*, *write_bit()* - niskopoziomowa obsługa bitów,
- *wire_read()*, *wire_write()* - wysyłanie i odbieranie bajtów.

Do generacji precyzyjnych opóźnień czasowych, wymaganych przez protokół (na poziomie mikrosekund) napisano funkcję *delay_us()* przy pomocy timera sprzętowego TIM6.

Program w pętli głównej inicjalizuje pomiar w czujniku, wysyłając do niego specjalną komendę. A po upływie 750 ms (czas konwersji temperatury z postaci analogowej na cyfrową przez komponent) temperatura jest odczytywana i zapisywana w strukturze danych, skąd dalej może być wykorzystywana przez system. Na przykład do kompensacji temperaturowej, warunków stop obciążenia oraz zapisu na karcie SD.

```

329= void delay_us(uint32_t us) {
330=     __HAL_TIM_SET_COUNTER(&htim6, 0);
331=     while (__HAL_TIM_GET_COUNTER(&htim6) < us) {}
332= }
333= HAL_StatusTypeDef wire_reset_1(void) {
334=
335=     int rc;
336=     HAL_GPIO_WritePin(temp_1_GPIO_Port, temp_1_Pin, GPIO_PIN_RESET);
337=     delay_us(480);
338=     HAL_GPIO_WritePin(temp_1_GPIO_Port, temp_1_Pin, GPIO_PIN_SET);
339=     delay_us(70);
340=     rc = HAL_GPIO_ReadPin(temp_1_GPIO_Port, temp_1_Pin);
341=     delay_us(410);
342=     if (rc == 0)
343=         return HAL_OK;
344=     else
345=         return HAL_ERROR;
346= }
347= void write_bit_1(int value) {
348=
349=     if (value) {
350=         HAL_GPIO_WritePin(temp_1_GPIO_Port, temp_1_Pin, GPIO_PIN_RESET);
351=         delay_us(6);
352=         HAL_GPIO_WritePin(temp_1_GPIO_Port, temp_1_Pin, GPIO_PIN_SET);
353=         delay_us(64);
354=     }
355=     else {
356=         HAL_GPIO_WritePin(temp_1_GPIO_Port, temp_1_Pin, GPIO_PIN_RESET);
357=         delay_us(60);
358=         HAL_GPIO_WritePin(temp_1_GPIO_Port, temp_1_Pin, GPIO_PIN_SET);
359=         delay_us(10);
360=     }
361= }
362= int read_bit_1(void) {
363=
364=     int rc;
365=     HAL_GPIO_WritePin(temp_1_GPIO_Port, temp_1_Pin, GPIO_PIN_RESET);
366=     delay_us(6);
367=     HAL_GPIO_WritePin(temp_1_GPIO_Port, temp_1_Pin, GPIO_PIN_SET);
368=     delay_us(9);
369=     rc = HAL_GPIO_ReadPin(temp_1_GPIO_Port, temp_1_Pin);
370=     delay_us(55);
371=     return rc;
372= }
```

Rysunek 5.9: Fragment kodu zawierający funkcje obsługujące DS18B20.

5.6 Regulator i kompensacja temperaturowa.

Modułem, który stanowi serce algorytmu sterowania, jest regulator PID. Jak wyżej wspomniano, aby zapewnić stały okres próbkowania i wyzwalania regulatora, jego wywołanie umieszczone w przerwaniu, które cyklicznie uruchamia się co 50 ms (Rys. 5.10).

```

294 /* USER CODE BEGIN 4 */
295 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) {
296
297     if (htim->Instance == TIM2) {
298
299         timer_flag = 1;
300         setCurrent();
301         setBatteryVoltage();
302         setParameter();
303         regulatorPI();
304     }
305 }
```

Rysunek 5.10: Fragment programu zawierający kod przerwania wywołującego się co 50 ms.

Funkcje, których nazwa rozpoczyna się od set – odpowiedzialne są za odbieranie wartości z przetworników pomiarowych lub przeliczenie ich na rezystancję, bądź moc. Funkcja *setParameter()* wykorzystywana jest w szczególności przy pracy w trybach pomiarowych – CR i CP, gdzie konieczne jest przeliczenie wartości parametrów takich jak rezystancja czy moc.

Typ regulatora wykorzystywanego w obciążeniu to PID. Zdecydowano się na niego ze względu na uniwersalność, prostotę implementacji oraz skuteczność w sterowaniu dynamicznie zmieniającymi się parametrami układu. Dzięki zastosowaniu wszystkich trzech członów – proporcjonalnego, całkującego i różniczkującego, możliwe jest szybkie osiągnięcie zadanej wartości pracy, przy dobrym ograniczeniu przeregulowania i zapewnieniu stabilności układu. Działanie regulatora PID opisuje równanie 5.1.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (5.1)$$

gdzie: $u(t)$ - sygnał sterujący, $e(t)$ - uchyb, K_p - wzmacnienie członu proporcjonalnego, K_i - wzmacnienie członu całkującego, K_d - wzmacnienie członu różniczkującego.

Funkcja *regulatorPI()* (Rys. 5.11) rozpoczyna się od wyznaczenia wewnętrznej zmiennej *setpoint*, której wartość zależy od wybranego trybu pracy. Dla trybów CP i CR wykonywane są odpowiednie obliczenia, pozwalające na przeliczenie wartości mocy lub rezystancji na odpowiadający im prąd odniesienia (wzory 5.1 i 5.2 z rozdziału 5.10). Dla trybu CC taka konwersja jest niepotrzebna – wartość jest pobierana bezpośrednio z interfejsu użytkownika.

Następnym krokiem w pracy regulatora jest obliczenie członów – proporcjonalnego (P),

całkującego (I) i różniczkującego (D). Regulator korzysta z mechanizmu anty-windup – aktualizuje całkę, tylko wtedy, gdy wartość mieści się w dopuszczalnym zakresie (0 A do 15 A przewidywanego prądu wyjścia). Dodatkowo w funkcji zaimplementowano mechanizm kompensacji temperaturowej, realizowanej przez funkcję *compensate()*, która uwzględnia prąd obciążenia oraz temperaturę tranzystora MOSFET i interpoluje wartości napięcia bramka-źródło zapisane w tabeli kalibracyjnej (dane do tabeli zostały pobrane z symulacji w Ltspice, patrz Rys. 3.3). Pozwala to na dynamiczne dostosowanie parametrów regulacji do zmian charakterystyki elementu mocy wynikającej ze wzrostu jego temperatury.

Ze względu na silnie nieliniową charakterystykę prądową tranzystora MOSFET, opisaną zależnością $I_D = f(V_{gs}, T)$, bezpośrednie zastosowanie wyjścia regulatora PID do sterowania napięciem bramki prowadziłoby do błędów regulacji. W celu zapewnienia liniowej odpowiedzi układu regulacji na zmiany wartości zadanej prądu oraz kompensacji temperaturowej, wyjście regulatora (będące wartością prądową) jest przeliczane na odpowiadające mu wymagane napięcie bramka-źródło V_{gs} poprzez odwzorowanie nieliniowej charakterystyki tranzystora.

Tabela $V_{gs_table}[i]$ (Rys. 5.12) zawiera wartości bramka-źródło wyznaczone za pomocą symulacji (Rys. 3.3) dla zadanych temperatur. W celu obliczenia napięcia $V_{gs}(I, T)$ w punkcie pośrednim I , zastosowano interpolację:

$$V_{gs}(I, T) = V_{gs}[i] + (I - i) \cdot (V_{gs}[i + 1] - V_{gs}[i]). \quad (5.2)$$

gdzie: I - prąd, i - część całkowita prądu, $V_{gs}[i]$ - wartość napięcia bramka-źródło z tablicy dla temperatury T i prądu i , $V_{gs}(I, T)$ - wynik interpolacji.

Po uzyskaniu V_{gs_low} dla niższej temperatury T_{low} i V_{gs_high} dla wyższej T_{high} (obie z poprzedniego kroku), należy dopasować wynik do rzeczywistej temperatury złącza T.

$$V_{gs}(I, T) = V_{gs_low} + \frac{T - T_{low}}{T_{high} - T_{low}} \cdot (V_{gs_high} - V_{gs_low}). \quad (5.3)$$

gdzie: T - temperatura wejściowa, T_{low} - niższa temperatura przedziału, w którym znajduje się T , T_{high} - wyższa temperatura przedziału, V_{gs_low} - wynik interpolacji prądowej przy T_{low} wg wzoru 5.2, V_{gs_high} - wynik interpolacji prądowej przy T_{high} , $V_{gs}(I, T)$ - wartość napięcia bramka-źródło po interpolacji w obu osiach (prądowej i temperaturowej).

Następnie obliczany zostaje końcowy współczynnik kompensacji:

$$\text{współczynnik_kompensacji} = \frac{V_{gs}(I, T)}{V_{gs}(I, 25^\circ C)} \quad (5.4)$$

Wartość ze wzoru 5.4 reprezentuje stosunek napięcia V_{gs} wymaganego do uzyskania prądu I w temperaturze T do napięcia wymaganego do uzyskania tego samego prądu I w temperaturze referencyjnej $25^\circ C$. Takie przeliczenie uwzględnia nieliniowy wpływ

```

23=void regulatorPI(void) {
24
25     static float integral = 0.0;
26     static float prev_error = 0.0;
27     float setpoint = 0.0;
28
29     if (getMode() == 'P')
30         setpoint = getSetpoint() / getBatteryVoltage();
31     else if (getMode() == 'R')
32         setpoint = getBatteryVoltage() / getSetpoint();
33     else
34         setpoint = getSetpoint();
35
36     if (getOnOff() != 0) {
37
38         float error = setpoint - getCurrent();
39         float derivative = (error - prev_error) / 0.05f;
40         prev_error = error;
41
42         float provisional_output = (Kp * error) + (Ki * integral) + (Kd * derivative);
43         provisional_output *= compensate(getCurrent(), getTemp_1());
44         if (provisional_output > 0.0f && provisional_output < 15.0f) {
45             integral += error * 0.05f;
46         }
47
48         float output = (Kp * error) + (Ki * integral) + (Kd * derivative);
49         output *= compensate(getCurrent(), getTemp_1());
50         if (output > 15.0f) output = 15.0f;
51         if (output < 0.0f) output = 0.0f;
52         uint16_t dac_output = (uint16_t)((output / 15.0) * 37683) + 27852;
53         setDacOutput(dac_output);
54     } else {
55
56         integral = 0.0;
57         prev_error = 0.0;
58         uint16_t dac_output = 0;
59         setDacOutput(dac_output);
60     }
61 }
62 }
```

Rysunek 5.11: Funkcja regulatora PID.

```

64=float compensate(float current, float temperature) {
65
66    const float Vgs_table[3][16] = {
67        {2.100, 2.266, 2.351, 2.420, 2.480, 2.534, 2.585, 2.632,
68         2.678, 2.721, 2.763, 2.804, 2.843, 2.882, 2.919, 2.956},
69        {2.002, 2.223, 2.321, 2.400, 2.466, 2.528, 2.585, 2.638,
70         2.690, 2.738, 2.786, 2.831, 2.875, 2.918, 2.960, 3.002},
71        {1.900, 2.176, 2.286, 2.372, 2.448, 2.517, 2.580, 2.639,
72         2.696, 2.749, 2.802, 2.852, 2.900, 2.948, 2.994, 3.039}
73    };
74
75    if (current < 0.0f)      current = 0.0f;
76    if (current >= 15.0f)    current = 14.999f;
77    if (temperature < 25.0f)  temperature = 25.0f;
78    if (temperature > 175.0f) temperature = 175.0f;
79
80    int i = (int)current;
81    int temp_l_idx = 0;
82    int temp_h_idx = 0;
83    float corr_temp = 0.0f;
84    if (temperature < 100) {
85        temp_l_idx = 0;
86        temp_h_idx = 1;
87        corr_temp = (temperature - 25) / 75.0f;
88    }
89    else {
90        temp_l_idx = 1;
91        temp_h_idx = 2;
92        corr_temp = (temperature - 100) / 75.0f;
93    }
94
95    float corr_25 = Vgs_table[0][i] + (current - (float)i) * (Vgs_table[0][i+1] - Vgs_table[0][i]);
96    if (corr_25 < 0.001f)  return 1.0f;
97
98    float corr_t_low = Vgs_table[temp_l_idx][i] + (current - (float)i) *
99        (Vgs_table[temp_l_idx][i+1] - Vgs_table[temp_l_idx][i]);
100   float corr_t_high = Vgs_table[temp_h_idx][i] + (current - (float)i) *
101       (Vgs_table[temp_h_idx][i+1] - Vgs_table[temp_h_idx][i]);
102
103  return (float)((corr_t_low + corr_temp * (corr_t_high - corr_t_low)) / corr_25);
104 }
```

Rysunek 5.12: Funkcja *compensate()*.

temperatury na charakterystykę tranzystora.

Wyjście regulatora po odpowiednim ograniczeniu jest przeliczane na wartość sterującą dla przetwornika DAC. Umożliwia to płynne sterowanie napięciem bramki tranzystora, a tym samym regulację prądu w obciążeniu.

Nastawy regulatora PID zostały wyznaczone metodą inżynierską na podstawie obserwacji przebiegu prądu. Regulując człon proporcjonalny i różniczkowy, dostosowywano krzywą w stanie nieustalonym, minimalizując przeregulowania i zwiększaając dynamikę układu. Człon całkujący został dobrany relatywnie wysoko, jednak eliminuje uchyb statyczny i umożliwia precyzyjne utrzymanie prądu, bez istotnego pogorszenia dynamiki układu.

5.7 Zapis danych pomiarowych na kartę SD.

W celu zapewnienia rejestracji parametrów w trakcie pracy urządzenia, zaimplementowano funkcje zapisu danych na zewnętrznym nośniku pamięci. Komunikacja z kartą SD realizowana jest za pomocą magistrali SPI oraz biblioteki FatFS – służącej do obsługi systemu plików FAT32, kompatybilnego z wieloma kartami SD.

W ramach startu procesu zapisu danych, przy przejściu urządzenia w stan rejestracji, automatycznie tworzony jest folder o nazwie odpowiadającej dacie i godzinie rozpoczęcia zapisu (Rys. 5.13). Jest to możliwe dzięki wykorzystaniu zegara czasu rzeczywistego – RTC (Real-Time Clock). W generowanym folderze tworzony jest plik CSV, do którego wpisywane są dane pomiarowe (Rys. 5.14).

Dane są buforowane w celu minimalizacji liczby operacji zapisu danych, tak aby nie zużyć przedwcześnie zewnętrznego nośnika pamięci. Po zapełnieniu bufora dane wysyłane są na kartę.

Format tabeli tworzonej na nośniku pamięci obejmuje parametry takie jak:

- czas pomiaru [s],
- wartość zadana - setpoint,
- parametr - zależny od trybu pracy (moc, rezystancja, prąd),
- prąd obciążenia,
- napięcie wejściowe akumulatora,
- wyjście przetwornika DAC,
- temperatura tranzystora - temp_1,
- temperatura otoczenia - temp_2,

- stan urządzenia - ON / OFF,
- ładunek wyprowadzony - qlmb.

Dodatkowo przy pomocy zmiennych znajdujących się w strukturze danych, która opisana jest na str. 33, możliwa jest zmiana prędkości pobierania danych pomiarowych oraz wielkości bufora, możliwych do zmiany poprzez komunikację z komputerem PC.

```

46= void getDateime(char *date_time_str) {
47
48     RTC_DateTypeDef sDate;
49     RTC_TimeTypeDef sTime;
50
51     HAL_RTC_GetDate(&hrtc, &sDate, RTC_FORMAT_BIN);
52     HAL_RTC_Gettime(&hrtc, &sTime, RTC_FORMAT_BIN);
53     sprintf(date_time_str, DATE_TIME_LENGTH, "%02d%02d%02d%02d",
54             sDate.Date, sTime.Hours, sTime.Minutes);
55 }
```

Rysunek 5.13: Kod funkcji pobierającej dane z RTC w celu stworzenia nazwy folderu.

```

154=void writeBufferToCsv(char buffer[][256], uint16_t size) {
155     sd_attach();
156
157     f_mount(&fs, "", 1);
158
159     res = f_open(&file, file_path, FA_WRITE | FA_OPEN_APPEND);
160     if (res == FR_OK) {
161         UINT bytes_written;
162
163         for (int i = 0; i < size; i++) {
164             res = f_write(&file, buffer[i], strlen(buffer[i]), &bytes_written);
165             if (res != FR_OK) {
166                 myprintf("f_write error (%i)\r\n", res);
167                 break;
168             }
169         }
170
171         myprintf("Data was added to .csv\r\n");
172         f_close(&file);
173
174         for (int i = 0; i < size; i++) {
175             memset(buffer[i], 0, 256);
176         }
177
178     } else {
179         myprintf("ERROR CSV (%i)\r\n", res);
180     }
181
182     f_mount(NULL, "", 0);
183     sd_detach();
184 }
```

Rysunek 5.14: Fragment funkcji realizującej zapis danych pomiarowych na kartę SD.

5.8 Panel sterowania i Aplikacja na PC.

Oprogramowanie urządzenia, służące do obsługi przez użytkownika, składa się z dwóch osobnych części:

1. obsługa panelu sterowania wbudowanego w urządzenie, w postaci wyświetlacza i jego menu, impulsatora oraz przycisku,
2. aplikacja na komputer PC realizująca komunikacje UART z obciążeniem.

Zarówno pierwszy, jak i drugi sposób pracy z urządzeniem pozwala na zmianę parametrów pracy obciążenia. Ten podrozdział poświęcony został obsłudze interfejsu wbudowanego w urządzenie, służącego do autonomicznej pracy.

```
40 void lcdEnablePulse(void) {
41     HAL_GPIO_WritePin(EN_GPIO_Port, EN_Pin, GPIO_PIN_SET);
42     HAL_Delay(5);
43     HAL_GPIO_WritePin(EN_GPIO_Port, EN_Pin, GPIO_PIN_RESET);
44     HAL_Delay(5);
45 }
46
47 void lcdSend4Bits(uint8_t data) {
48
49     HAL_GPIO_WritePin(DB4_GPIO_Port, DB4_Pin, (data & 0x01) ? GPIO_PIN_SET : GPIO_PIN_RESET);
50     HAL_GPIO_WritePin(DB5_GPIO_Port, DB5_Pin, (data & 0x02) ? GPIO_PIN_SET : GPIO_PIN_RESET);
51     HAL_GPIO_WritePin(DB6_GPIO_Port, DB6_Pin, (data & 0x04) ? GPIO_PIN_SET : GPIO_PIN_RESET);
52     HAL_GPIO_WritePin(DB7_GPIO_Port, DB7_Pin, (data & 0x08) ? GPIO_PIN_SET : GPIO_PIN_RESET);
53     lcdEnablePulse();
54 }
55
56 void lcdSendCmnd(uint8_t cmnd) {
57
58     HAL_GPIO_WritePin(RS_GPIO_Port, RS_Pin, GPIO_PIN_RESET);
59     HAL_GPIO_WritePin(RW_GPIO_Port, RW_Pin, GPIO_PIN_RESET);
60     lcdSend4Bits(cmnd >> 4);
61     lcdSend4Bits(cmnd & 0x0F);
62 }
63 }
```

Rysunek 5.15: Funkcje obsługujące komunikacje z wyświetlaczem LCD.

Wyświetlacz LCD wykorzystany w projekcie to popularny model oparty na sterowniku HD44780. Charakteryzuje się prostotą obsługi i umożliwia prezentację kluczowych parametrów pracy urządzenia, takich jak napięcie akumulatora, prąd obciążenia, tryb pomiarowy i ustawienie parametru zadanego. W celu uniknięcia zbyt dużej ilości przewodów sygnałowych, komunikacja z LCD odbywa się w konfiguracji połówkowej. Oznacza to, że komendy wysyłane do wyświetlacza są podzielone na dwie części.

Spis komend oraz sposób komunikacji z wyświetlaczem (Rys. 5.15) można znaleźć w dokumentacji katalogowej sterownika HD44780 [10].

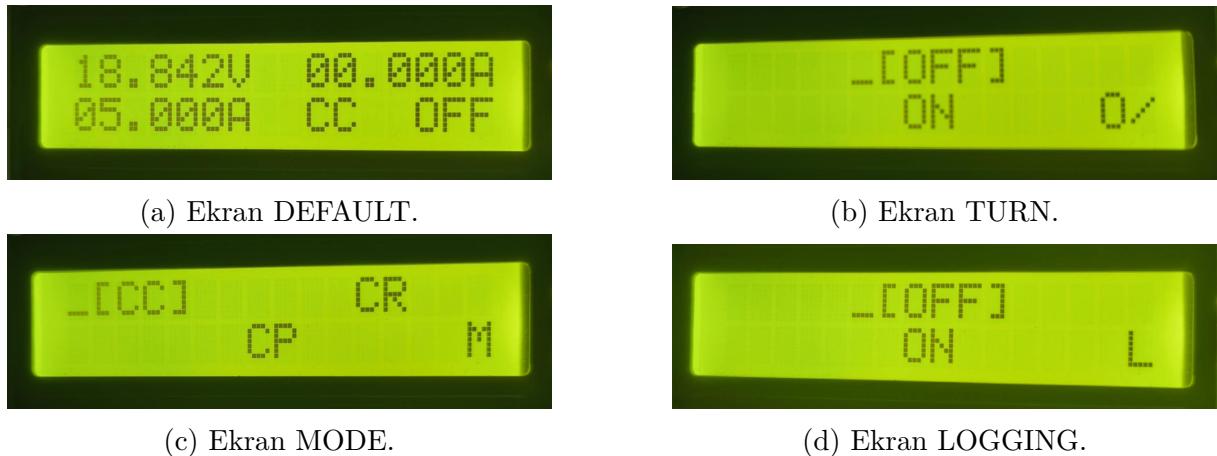
Menu składa się z czterech trybów:

- DEFAULT – główny wygląd menu,
- TURN – włączanie / wyłączanie obciążenia,
- MODE – ustawienie trybu pracy urządzenia – CC / CR / CP,
- LOGGING – włączanie / wyłączanie zapisu danych na kartę SD.

Do przemieszczania się po kolejnych ekranach menu wykorzystany został przycisk monostabilny umieszczony po lewej stronie wyświetlacza. Każde naciśnięcie powoduje rozpoczęcie przerwania, które wysyła szereg komend zarządzających wyświetlaczem LCD. Wygląd tych ekranów można zobaczyć na grafikach w Rys. 5.16.

Podkreślenie przed parametrami na ekranach Rys. 5.16b, 5.16c i 5.16d sygnalizuje aktywne właściwości. To znaczy, że urządzenie z Rys. 5.16 nie obciąża akumulatora (Rys. 5.16b), jest w trybie pomiarowym CC (Rys. 5.16c) i nie ma włączonej rejestracji danych (Rys. 5.16d).

Ekran główny – DEFAULT: - W pierwszej linii po lewej stronie wyświetlana jest wartość napięcia akumulatora, po prawej – aktualny pomiar natężenia prądu obciążenia. W drugiej linii mieści się wartość nastawiana prądu obciążenia / mocy / rezystancji (*setpoint*) – do której dąży układ w przypadku uruchomienia, a także informacja na temat



Rysunek 5.16: Ekrany wyświetlacza będącego częścią interfejsu użytkownika.

```

184 // FUNCTIONALITY OF CLK IMPULSATOR
185 if ((GPIO_Pin == GPIO_PIN_12) && (current_time - last_clk_change > 200)) {
186     switch(_display) {
187
188         case DEFAULT:
189             if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_11) != GPIO_PIN_RESET) {
190
191                 float new_setpoint = getSetpoint() + (0.001 * (float)multiplier);
192                 setSetpoint(new_setpoint);
193             }
194             else {
195
196                 float new_setpoint = getSetpoint() - (0.001 * (float)multiplier);
197                 setSetpoint(new_setpoint);
198             }
199             break;
200         case TURN:
201             lcd_menu_idx = (lcd_menu_idx >= ON) ? OFF : ON;
202             break;
203         case MODE:
204             lcd_menu_idx += 1;
205             if (lcd_menu_idx > 2)
206                 lcd_menu_idx = OFF;
207             break;
208         case LOGGING:
209             lcd_menu_idx = (lcd_menu_idx >= ON) ? OFF : ON;
210             break;
211         default:
212             break;
213     }
214     lcd_update_flag = ON;
215     last_clk_change = current_time;
216 }
217 }
```

Rysunek 5.17: Fragment kodu funkcji realizującej pracę impulsatora.

aktualnego trybu pomiarowego i sygnalizacja on/off – informująca o tym, czy urządzenie obciąża DUT.

Do poruszania się po parametrach wykorzystywany jest impulsator. W przypadku ekranu DEFAULT służy on do zmiany wartości parametru *setpoint*. Dodatkowo, każde naciśnięcie przycisku wbudowanego w enkoder umożliwia zmianę aktywnego rzędu edycji. Użytkownik może w ten sposób przesuwać kursor pomiędzy rzędami: częściami tysięcznymi, setnymi, dziesiętnymi oraz całkowitymi (Rys. 5.17). Takie rozwiązanie ułatwia ustawienie wartości parametru.

W przypadku ekranów Rys. 5.16b, 5.16c i 5.16d do wyboru ustawienia – ON/OFF lub CC/CR/CP, także wykorzystywany jest impulsator. Natomiast przycisk enkodera zatwierdza wartość parametru, na którym ustawiony jest kursor.

5.9 Komunikacja z PC - UART.

W urządzeniu zaimplementowano funkcje komunikacji z komputerem PC za pomocą interfejsu UART, w trybie asynchronicznym z wykorzystaniem przerwań. Obsługa protokołu została zawarta w module „komunikacjaPC2.c”.

Transmisja danych odbywa się w postaci ramek złożonych z nagłówka, długości danych, parametrów pracy urządzenia oraz sumy kontrolnej, która umożliwia weryfikację poprawnej transmisji. Odbierane dane z interfejsu UART umieszczone są w buforze kołowym. Dzięki takiemu rozwiązaniu możliwa jest obsługa asynchronicznych transmisji bez ryzyka utraty danych przy dużych prędkościach. Funkcja *checkForData()* (Rys 5.20) analizuje zawartość bufora i w momencie odebrania kompletnej ramki (Rys. 5.18 o poprawnej sumie kontrolnej wywołuje funkcję *saveDataToStruct()* (Rys 5.19), która aktualizuje zawartość wewnętrznej struktury danych oraz zapisuje nowe parametry ustawień do pamięci EEPROM (Rys. 5.21).

Header (0x77)	Length (0x1F)				
setpoint - data [0-3]	mode - data [4]	onOff - data [5]	logging - data [6]	logging buffer size - data [7-8]	logging speed - data [9-10]
wire resistance - data [11-14]	Kp - data [15-18]	Ki - data [19-22]	Kd - data [23-26]	max time - data [27-30]	
CRC(0x)					

Rysunek 5.18: Graficzne przedstawienie ramki danych służącej do komunikacji z komputerem PC.

```

169 void saveDataToStruct( uint16_t start_msg_idx, uint8_t length, DataStruct *data ) {
170
171     myprintf("I am in saving Data process...\r\n");
172
173     memcpy(&data->header, &rx_buffer[start_msg_idx], sizeof(uint8_t));
174     setSetpoint(*(float*)&rx_buffer[start_msg_idx + 2]);
175     setMode(*(uint8_t*)&rx_buffer[start_msg_idx + 6]);
176     setOnOff(*(uint8_t*)&rx_buffer[start_msg_idx + 7]);
177     setLogging(*(uint8_t*)&rx_buffer[start_msg_idx + 8]);
178     setLoggingBufferSize(*(uint16_t*)&rx_buffer[start_msg_idx + 9]);
179     setLoggingSpeed(*(uint16_t*)&rx_buffer[start_msg_idx + 11]);
180
181     memcpy(&data->data[43], &rx_buffer[start_msg_idx + 13], sizeof(float));
182     memcpy(&data->data[47], &rx_buffer[start_msg_idx + 17], sizeof(float));
183     memcpy(&data->data[51], &rx_buffer[start_msg_idx + 21], sizeof(float));
184     memcpy(&data->data[55], &rx_buffer[start_msg_idx + 25], sizeof(float));
185     memcpy(&data->data[59], &rx_buffer[start_msg_idx + 29], sizeof(float));
186
187     EEPROM_Save_Partial(data);
188 }
```

Rysunek 5.19: Funkcja *saveDataToStruct()* zapisująca parametry odebrane przez UART do głównej struktury.

```

122 void checkForData(DataStruct *data) {
123
124     static uint8_t message_start_idx = 0;
125     static uint16_t read_idx = 0;
126     static uint8_t buffer[37];
127     static uint8_t buf_pos = 0;
128     static uint8_t message_flag = 0;
129     static uint8_t length = 0;
130     static uint32_t crc = 0;
131
132     while (read_idx != uart_idx) {
133         uint8_t byte_ = rx_buffer[read_idx];
134
135         if (message_flag == 0) {
136             if (byte_ == 0x77) {
137                 message_flag = 1;
138                 buffer[0] = byte_;
139                 buf_pos = 1;
140                 crc = byte_;
141                 message_start_idx = read_idx;
142             }
143         } else {
144             buffer[buf_pos++] = byte_;
145
146             if (buf_pos == 2) {
147                 length = byte_;
148             }
149             if (buf_pos <= 33) {
150                 crc += byte_;
151             }
152             if (buf_pos == 36) {
153                 message_flag = 0;
154                 uint32_t rx_crc;
155                 memcpy(&rx_crc, &buffer[buf_pos - 3], sizeof(uint32_t));
156                 if (crc == rx_crc) {
157
158                     saveDataToStruct(message_start_idx, length, data);
159                     memcpy(&data->crc, &rx_crc, sizeof(float));
160
161                     buf_pos = 0;
162                     crc = 0;
163                 }
164             }
165             if (read_idx >= RX_BUFFER_SIZE) read_idx = 0;
166         }
167     }
}

```

Rysunek 5.20: Funkcja *checkForData()* - wywoływana w głównej pętli.

```

190 void EEPROM_Save_Partial(DataStruct* data) {
191
192     float value = 0;
193     HAL_FLASHEx_DATAEEPROM_Unlock();
194
195     HAL_FLASHEx_DATAEEPROM_Program(FLASH_TYPEPROGRAMDATA_WORD, EEPROM_ADDR_HEADER, data->header);
196     value = getSetpoint();
197     uint32_t value_u32 = *(uint32_t*)&value;
198     HAL_FLASHEx_DATAEEPROM_Program(FLASH_TYPEPROGRAMDATA_WORD, EEPROM_ADDR_SETPOINT, value_u32);
199     HAL_FLASHEx_DATAEEPROM_Program(FLASH_TYPEPROGRAMDATA_WORD, EEPROM_ADDR_MODE, getMode());
200     HAL_FLASHEx_DATAEEPROM_Program(FLASH_TYPEPROGRAMDATA_WORD, EEPROM_ADDR_ONOFF, getOnOff());
201     HAL_FLASHEx_DATAEEPROM_Program(FLASH_TYPEPROGRAMDATA_WORD, EEPROM_ADDR_LOGGING, getLogging());
202     HAL_FLASHEx_DATAEEPROM_Program(FLASH_TYPEPROGRAMDATA_WORD, EEPROM_ADDR_BUF_SIZE,
203                                     getLoggingBufferSize());
204     HAL_FLASHEx_DATAEEPROM_Program(FLASH_TYPEPROGRAMDATA_WORD, EEPROM_ADDR_SPEED, getLoggingSpeed());
205     value = *(float*)&data->data[43];
206     value_u32 = *(uint32_t*)&value;
207     HAL_FLASHEx_DATAEEPROM_Program(FLASH_TYPEPROGRAMDATA_WORD, EEPROM_ADDR_DATA43, value_u32);
208     value = *(float*)&data->data[47];
209     value_u32 = *(uint32_t*)&value;
210     HAL_FLASHEx_DATAEEPROM_Program(FLASH_TYPEPROGRAMDATA_WORD, EEPROM_ADDR_DATA47, value_u32);
211     value = *(float*)&data->data[51];
212     value_u32 = *(uint32_t*)&value;
213     HAL_FLASHEx_DATAEEPROM_Program(FLASH_TYPEPROGRAMDATA_WORD, EEPROM_ADDR_DATA51, value_u32);
214     value = *(float*)&data->data[55];
215     value_u32 = *(uint32_t*)&value;
216     HAL_FLASHEx_DATAEEPROM_Program(FLASH_TYPEPROGRAMDATA_WORD, EEPROM_ADDR_DATA55, value_u32);
217     value = *(float*)&data->data[59];
218     value_u32 = *(uint32_t*)&value;
219     HAL_FLASHEx_DATAEEPROM_Program(FLASH_TYPEPROGRAMDATA_WORD, EEPROM_ADDR_DATA59, value_u32);
220
221     HAL_FLASHEx_DATAEEPROM_Lock();
222 }

```

Rysunek 5.21: Funkcja *EEPROM_Save_Partial* służąca do zapisu ważnych parametrów w pamięci EEPROM.

5.10 Tryby pracy obciążenia.

Urządzenie ma możliwość pracy w jednym z trzech wybranych trybów pracy:

- CC - stały prąd,
- CR - stała rezystancja,
- CP - stała moc.

W tych trybach urządzenie utrzymuje wartość opisaną w strukturze danych jako zmienna *parameter* na poziomie, podyktyowanym przez wartość zadaną – *setpoint*. W funkcji regulatora (Rys. 5.11) można zauważyć, że na bieżąco przeliczana jest lokalna wartość zadana w trybach CR i CP. Pozwala to na regulację rezystancji lub mocy obciążenia. Dodatkowo *parameter* zapisywany w strukturze danych także jest przeliczany za pomocą wartości napięcia i prądu (Rys. 5.22):

- CR

$$\frac{U}{I}, [\Omega] \quad (5.5)$$

- CP

$$U \cdot I, [W] \quad (5.6)$$

```

175 float getParameter(void) {
176     return *(float*)&myData.data[4];
177 }
178
179 void setParameter(void) {
180
181     float parameter = 0.0;
182     if (getMode() == 'C')
183         parameter = getCurrent();
184     if (getMode() == 'R')
185         parameter = getBatteryVoltage() / getCurrent();
186     if (getMode() == 'P')
187         parameter = getBatteryVoltage() * getCurrent();
188
189     memcpy(&myData.data[4], &parameter, sizeof(float));
190 }
```

Rysunek 5.22: Funkcje *set()* i *get()* obsługujące zmienną *parameter* w strukturze danych.

5.11 Warunki STOP.

Urządzenie posiada szereg warunków kończących pracę obciążenia (Rys. 5.23). Należą do nich:

- niskie napięcie na akumulatorze,
- wysoka temperatura tranzystora obciążenia,

- zbyt wysoki ładunek wyprowadzony,
- czas końcowy testu.

Dzięki tylu warunkom kończącym pracę urządzenia, jeśli użytkownik poprawnie wpisze warunki końcowe, dopasowując je do badanego akumulatora, obciążenie nie uszkodzi badanego akumulatora.

```

191     float minutesPassed = timeStop - (float)((HAL_GetTick() - time_base) / (1000.0*60.0));
192     if (minutesPassed < 0) {
193         setOnOff(0);
194         setLogging(0);
195     }
196
197     if (getBatteryVoltage() <= 11.2)
198         setOnOff(0);
199
200     if (qlmb/3600 > 250)
201         setOnOff(0);
202
203     if (getTemp_1() >= 170 && prev_temp >= 170) {
204
205         setDacOutput(0);
206         setOnOff(0);
207     }
208

```

Rysunek 5.23: Realizacja warunków kończących pracę urządzenia.

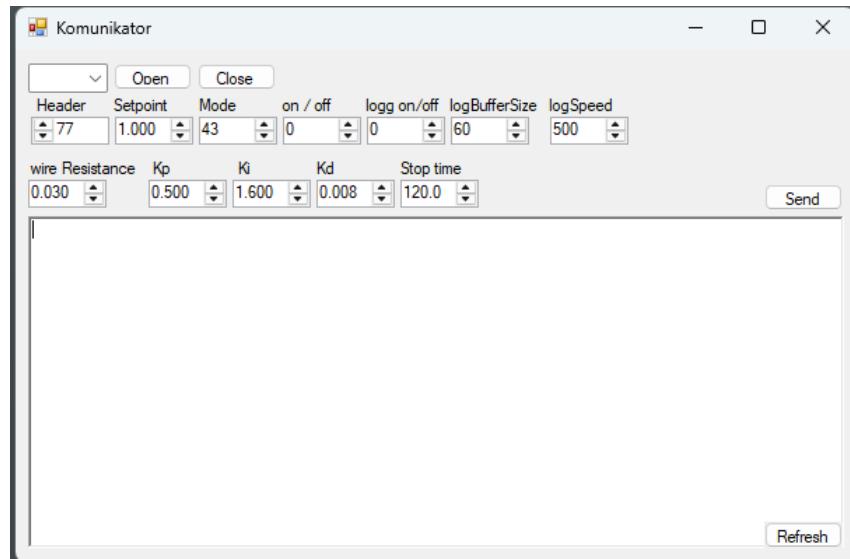
5.12 Aplikacja na komputer PC.

Aplikacja udostępnia graficzny interfejs użytkownika służący do sterowania obciążeniem aktywnym, z którym komunikuje się poprzez łącze UART. Umożliwia ona zdalną konfigurację parametrów pracy urządzenia oraz monitorowanie bieżących komunikatów debug. Aplikacja została opracowana w środowisku „Visual Studio 2022” z wykorzystaniem języka programowania „C#”.

Poniżej znajdują się parametry pracy urządzenia możliwe do konfiguracji przez aplikację:

- wartość zadana (setpoint),
- tryb pracy (mode) – CC / CR / CP,
- włączenie / wyłączenie obciążenia (On/Off),
- włączenie / wyłączenie rejestracji danych pomiarowych (logg On / Off),
- konfiguracja długości bufora i prędkości rejestracji danych,
- ustawienie rezystancji przewodów pomiarowych,
- nastawy regulatora PID (K_p, K_i i K_d),
- czas zakończenia testu (stop time).

Komunikacja między urządzeniem a komputerem PC odbywa się zgodnie z protokołem opisany w powyższym podrozdziale. Interfejs aplikacji z rysunku 5.24 został zaprojektowany z myślą o prostocie i przejrzystości, umożliwiając wygodną konfigurację parametrów pracy obciążenia bez konieczności korzystania z interfejsu lokalnego.



Rysunek 5.24: Prezentacja aplikacji do konfigurowania parametrów obciążenia.

Rozdział 6

Badania i pomiary

Celem przeprowadzonych badań było zweryfikowanie poprawności działania zaprojektowanego aktywnego obciążenia oraz analiza charakterystyk rozładowania akumulatorów 12 V w różnych trybach pracy urządzenia. Testy miały na celu sprawdzenie:

- czy układ poprawnie realizuje tryby CC, CR i CP,
- czy zaprojektowany regulator PID z kompensacją temperaturową zapewnia stabilne utrzymanie zadanych parametrów,
- czy urządzenie prawidłowo reaguje na warunki graniczne.

Badania zostały przeprowadzone w warunkach domowych, ale zasymulowano realną pracę urządzenia. Testy odbywały się w temperaturze pokojowej (ok. 22°C). Do testów wykorzystano:

- dwa akumulatory żelowe o różnych pojemnościach nominalnych (7.2 Ah i 286 Ah),
- różne wartości prądu obciążenia, zgodnie z trybami pracy.

Pomiary i proces obciążenia były realizowane przy pomocy zaprojektowanego obciążenia aktywnego. Dane były zapisywane w pliku CSV, a następnie analizowane w środowisku MATLAB.

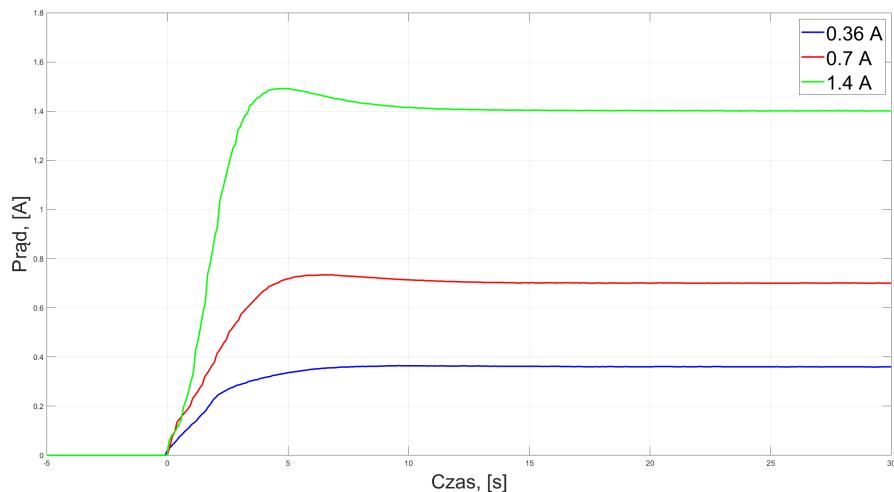
We wszystkich badaniach głównym kryterium zakończenia testu było osiągnięcie minimalnego napięcia dla akumulatora. W przypadku ogniw o mniejszej pojemności wartość wynosiła 11.0 V, a dla dużego 11.2 V. Przykładowe wyniki badań przedstawione zostały w kolejnych podrozdziałach.

6.1 Badanie charakterystyki w trybie CC (stały prąd).

Celem badania w trybie CC było zweryfikowanie możliwości utrzymania zadanej wartości prądu przez układ w trakcie całego procesu rozładowania akumulatora, analiza dy-

namiki układu w stanie przejściowym i ustalonym oraz wpływ obciążenia na wielkość ładunku wyprowadzonego i czas rozładowania akumulatora.

Badanie zostało przeprowadzone na akumulatorze żelowym o napięciu 12 V i pojemności 7.2 Ah. Tryb pracy obciążenia ustawiono na CC z wartością zadaną 360 mA / 700 mA / 1.4 A. Rejestracja danych z interwałem 100 ms została zapisana na karcie SD w pliku CSV. Warunkiem końcowym testu miał być spadek napięcia na akumulatorze do 11.0 V.



Rysunek 6.1: Przebiegi prądowe obciążenia dla prądów: 360 mA / 700 mA / 1.4 A.

6.1.1 Stan Przejściowy.

Stan nieustalony obejmuje początkową fazę testu, w której regulator PID stabilizuje wartość prądu i dąży do wartości zadanej. Taki stan trwa ok. 4 - 7 sekund do osiągnięcia $\pm 1\%$ wartości zadanej (Rys. 6.1). Zaobserwowano, że przebiegi odpowiedzi regulatora różnią się w zależności od nastawionej wartości zadanej. Dla mniejszych prądów stabilizacja jest wolniejsza, natomiast dla wyższych szybsza, jednak możliwe są chwilowe przeregulowania.

6.1.2 Analiza stanu ustalonego.

W celu analizy stanu ustalonego (Rys. 6.2, 6.3, 6.4) obliczone zostały następujące parametry (wartości zostały pobrane z tabel plików CSV wygenerowanych przez urządzenie):

- wartość średnia pomiaru:

$$\bar{x} = \frac{1}{N} \sum_{n=1}^N x_n \quad (6.1)$$

- odchylenie standardowe pomiaru:

$$s_x = \sqrt{\frac{1}{N-1} \sum_{n=1}^N (x_n - \bar{x})^2} \quad (6.2)$$

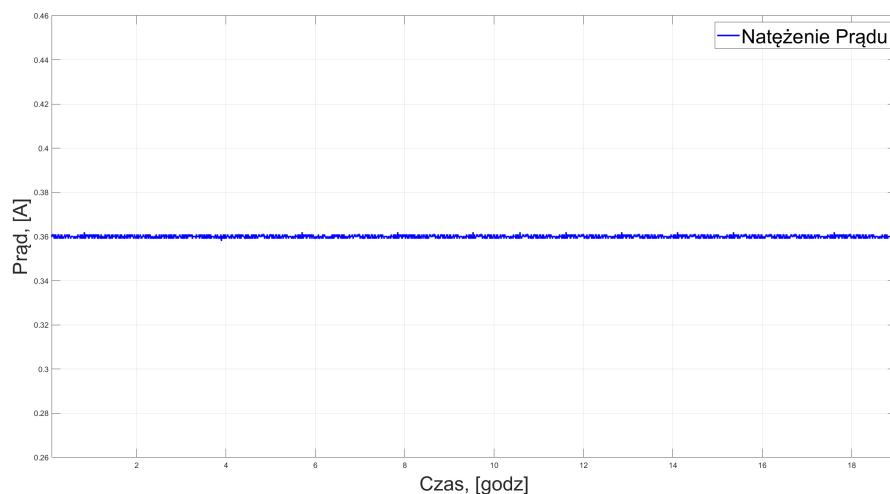
- wartość średnia błędu bezwzględnego:

$$\overline{\Delta_{bez}} = \frac{1}{N} \sum_{n=1}^N |x_n - x_{setpoint}| \quad (6.3)$$

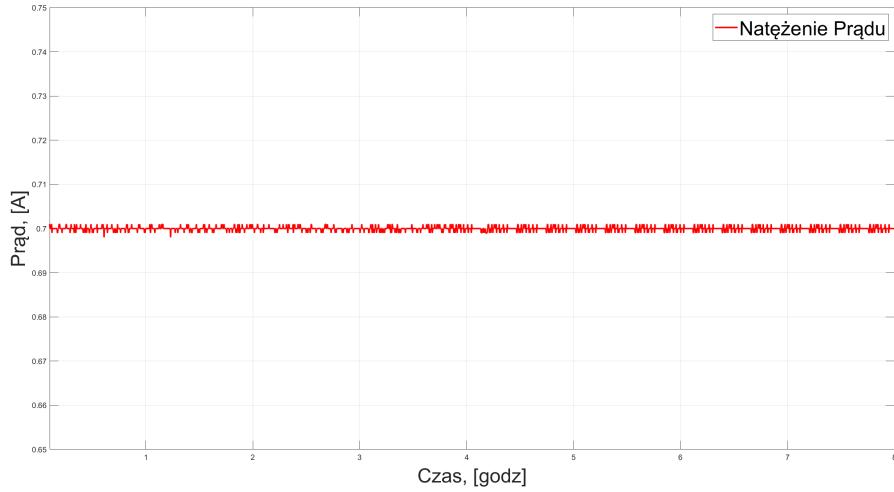
- wartość średnia błędu względnego:

$$\overline{\Delta_{wzg}} = \frac{1}{N} \sum_{n=1}^N \left(\frac{|x_n - x_{setpoint}|}{x_{setpoint}} \cdot 100\% \right) \quad (6.4)$$

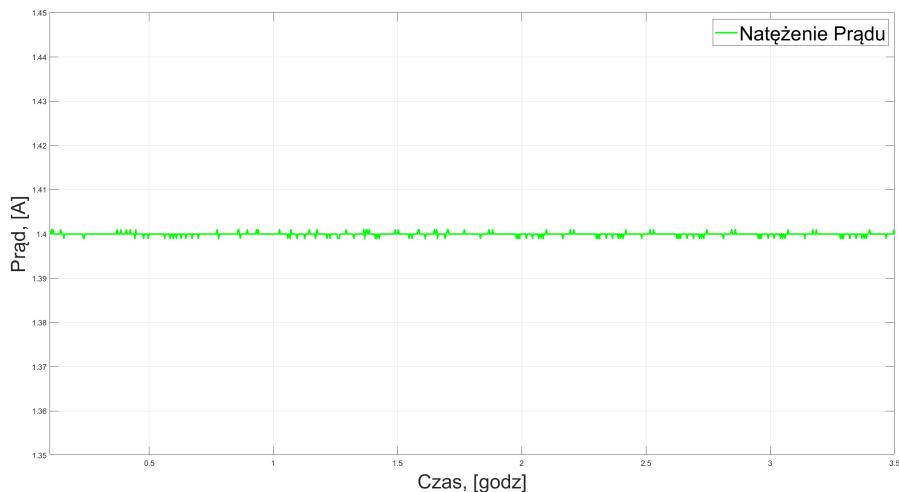
Gdzie: N - liczba pomiarów, n - numer pomiaru, $x_{setpoint}$ - wartość zadana.



Rysunek 6.2: Przebiegi prądowe obciążenia dla prądu: 360 mA.



Rysunek 6.3: Przebiegi prądowe obciążenia dla prądu: 700 mA.

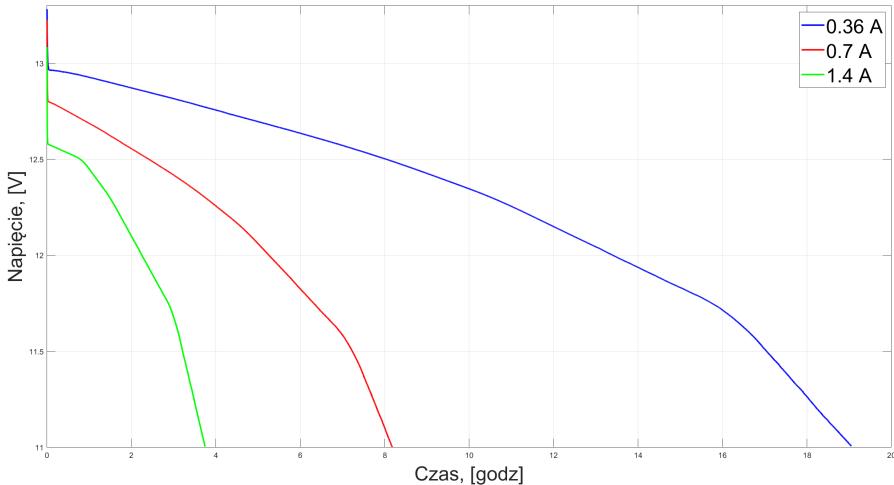


Rysunek 6.4: Przebiegi prądowe obciążenia dla prądu: 1.4 A.

Tabela 6.1: Wyliczone parametry do analizy stanu ustalonego pomiarów.

<i>setpoint</i>	$\bar{x}, [A]$	$s_x, [A]$	$\Delta_{bez}, [A]$	$\Delta_{wzgl}, [\%]$
0.36 A	0.3600	$5.39 \cdot 10^{-4}$	$6.81 \cdot 10^{-6}$	0.0019
0.7 A	0.7004	$3.2 \cdot 10^{-3}$	$5.74 \cdot 10^{-5}$	0.0082
1.4 A	1.4006	$2.4 \cdot 10^{-3}$	$5.90 \cdot 10^{-5}$	0.0042

Analizując dane z tabeli 6.1 można stwierdzić, że układ radzi sobie dobrze z utrzymywaniem stanu ustalonego. Rozrzut chwilowych pomiarów jest na bardzo niskim poziomie dla każdego testu. Wartości błędów względnych i bezwzględnych potwierdzają, że urządzenie działa z dokładnością do 1 mA.



Rysunek 6.5: Charakterystyka rozładowania akumulatora 12 V dla pełnego zakresu czasowego.

6.1.3 Analiza charakterystyk rozładowania.

Na Rys. 6.5 przedstawione zostały charakterystyki rozładowania akumulatora 12 V w trybie CC, dla prądów obciążenia: 360 mA / 700 mA / 1.4 A. Zaobserwować można:

- dla najniższego prądu proces rozładowania przebiegał wolno. Początkowy spadek napięcia był niewielki, a napięcie w stosunku do pozostałych charakterystyk długo utrzymuje się powyżej 12 V. Dalej napięcie spada bardzo liniowo, a na końcu widać znaczny spadek.
- dla prądu 0.7 A spadek napięcia jest wyraźny już od początku rozładowania.
- dla prądu 1.4 A czas rozładowania jest najkrótszy, test trwał mniej niż 4 godziny.

Wykresy dobrze oddają charakterystykę typowych akumulatorów żelowych. Im wyższy prąd obciążenia, tym krótszy czas rozładowania. Ponadto dzięki rejestrowanej wartości ładunku wyprowadzonego można zauważać kolejną zależność. Pomimo tego, że ognisko było za każdym razem ładowane w tych samych warunkach i do tego samego poziomu napięcia, wartość ładunku wyprowadzonego różni się w zależności od prądu obciążenia. Można to zaobserwować w tabeli poniżej.

Tabela 6.2: Wartość ładunku wyprowadzonego z akumulatora 12 V w zależności od obciążenia.

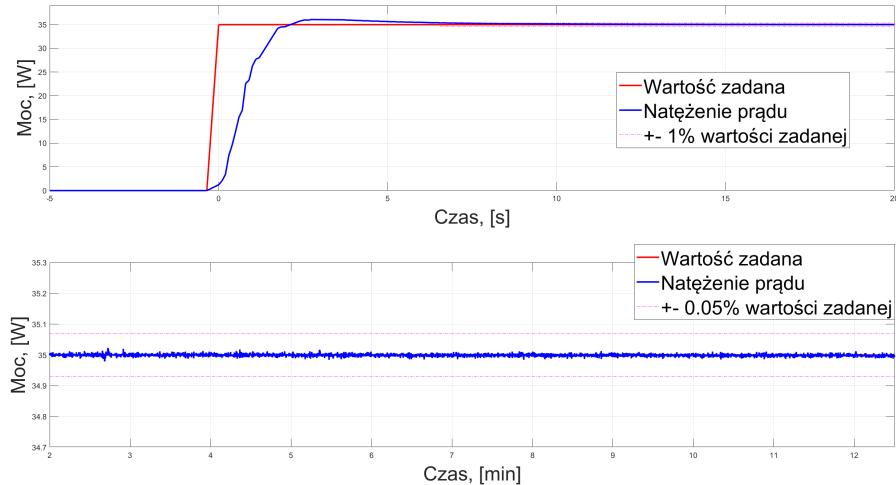
Prąd rozładowania, [A]	0.36	0.7	1.4
Ladunek, [Ah]	6.86	5.75	5.24

6.2 Badanie charakterystyki w trybie CP (stała moc).

Celem badania było sprawdzenie poprawności działania układu w trybie CP, czyli pracy ze stałą mocą obciążenia. Wartość zadaną mocy ustawiono na 35 W. Test został przeprowadzony na akumulatorze 12 V, 7.2 Ah.

6.2.1 Stan przejściowy.

Z wykresów przebiegu mocy (Rys. 6.6) wynika, że regulator PID skutecznie utrzymuje wartość zadanej mocy. Czas przejścia do stanu ustalonego wynosi około 3 sekundy, a moc mieści się w zakresie $\pm 1\%$ wartości zadanej.



Rysunek 6.6: Przebiegi mocy obciążenia dla mocy 35 W.

6.2.2 Analiza stanu ustalonego.

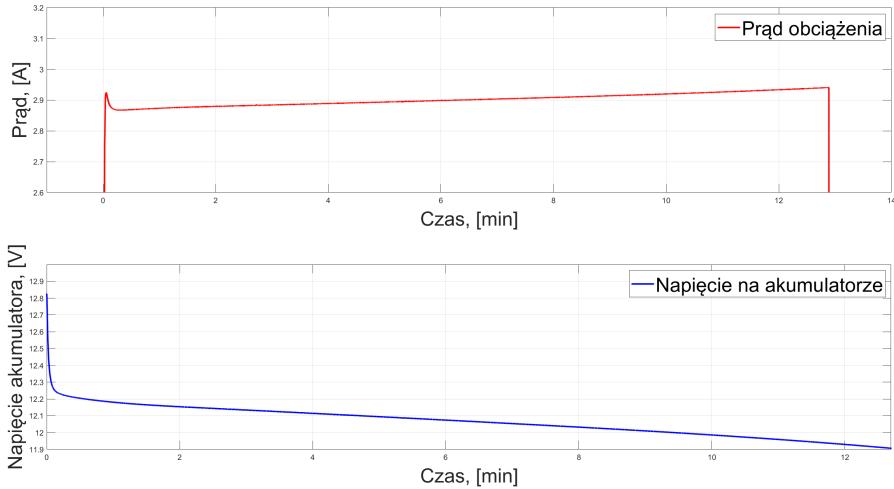
W celu analizy stanu ustalonego obliczone zostały parametry, których wzory znajdują się w podrozdziale 6.1.2.

Tabela 6.3: Wyliczone parametry dla trybu CP do analizy stanu ustalonego pomiarów.

<i>setpoint</i>	$\bar{x}, [W]$	$s_x, [W]$	$\Delta_{bez}, [W]$	$\Delta_{wzgl}, [\%]$
35W	35.0019	$1.84 \cdot 10^{-2}$	$5.7 \cdot 10^{-3}$	0.0163

6.2.3 Analiza charakterystyk rozładowania.

Wykresy prądu i napięcia akumulatora z rysunku 6.7 pokazują charakterystyczną zależność dla trybu CP: wraz ze spadkiem napięcia, aby utrzymać moc na zadanym poziomie, prąd rośnie.



Rysunek 6.7: Charakterystyka rozładowania akumulatora 12 V w trybie CP.

Badanie charakterystyki w trybie CP potwierdziło poprawne działanie zaprojektowanego układu obciążenia w tym trybie. Regulator skutecznie i stabilnie utrzymuje moc na zadanym poziomie, z odchyleniem nieprzekraczającym 0.2%. Ponadto czas nieustalony charakterystyki jest bardzo krótki (około 2 sekund), świadczy to o dobrej dynamice regulatora.

6.3 Badanie charakterystyki w trybie CR (stała rezystancja).

Celem badania było sprawdzenie poprawności działania układu w trybie CR – stałej rezystancji, w którym obciążenie powinno zachowywać się jak programowalny opornik o stałej rezystancji. Wartość zadana wynosiła 4Ω . Test został przeprowadzony na akumulatorze żelowym 12 V o pojemności 7.2 Ah.

6.3.1 Stan przejściowy.

Na podstawie wykresu (Rys. 6.8) można stwierdzić, że układ działa prawidłowo. Po czasie przejścia trwającym około 2.5 sekundy osiągnięty został stan ustalony, w którym wartość rezystancji nie wykracza poza $\pm 1\%$ wartości zadanej.

6.3.2 Analiza stanu ustalonego.

W celu analizy stanu ustalonego obliczone zostały parametry, których wzory znajdują się w podrozdziale 6.1.2.

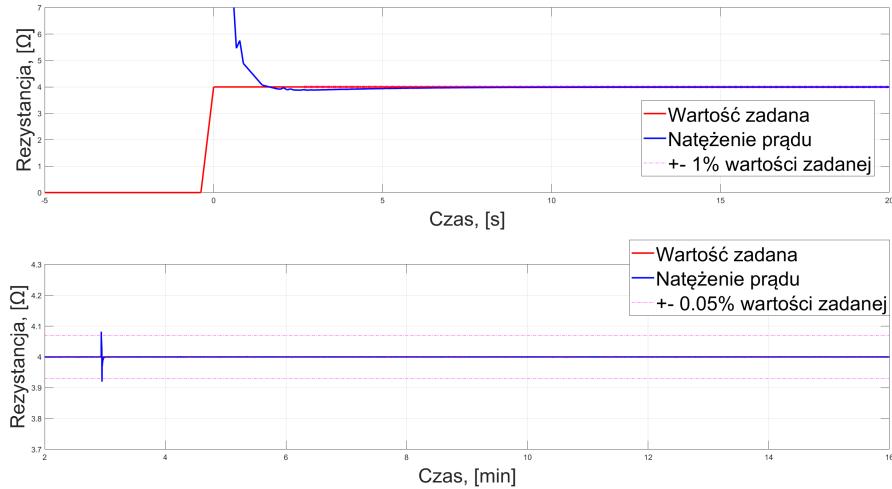
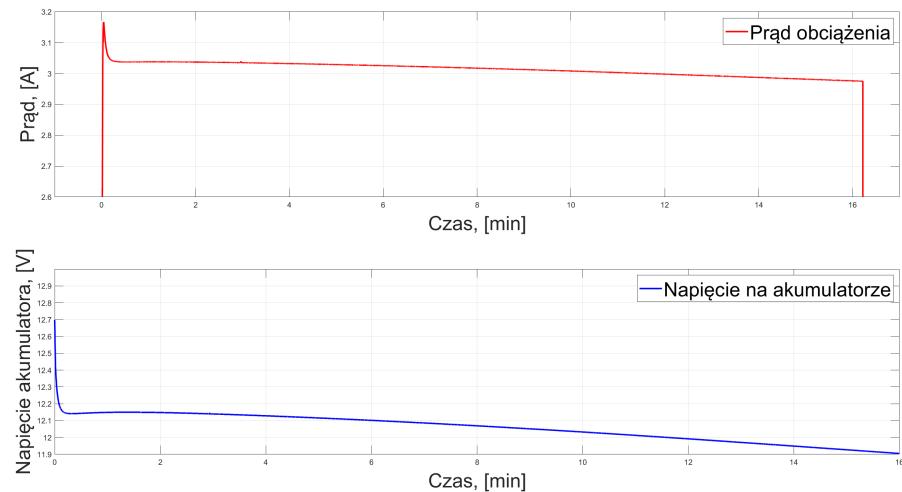
Rysunek 6.8: Przebiegi rezystancji obciążenia dla rezystancji 4Ω .

Tabela 6.4: Wyliczone parametry dla trybu CP do analizy stanu ustalonego pomiarów.

$setpoint$	$\bar{x}, [\Omega]$	$s_x, [\Omega]$	$\Delta_{bez}, [\Omega]$	$\Delta_{wzgl}, [\%]$
4Ω	3.9999	$1.8 \cdot 10^{-3}$	$4.4 \cdot 10^{-4}$	0.0110

6.3.3 Analiza charakterystyk rozładowania.

Analiza wykresu (Rys. 6.9) potwierdza, że wraz z malejącym napięciem na akumulatorze, prąd obciążenia maleje, zgodnie z prawem Ohma.



Rysunek 6.9: Charakterystyka rozładowania akumulatora 12 V w trybie CR.

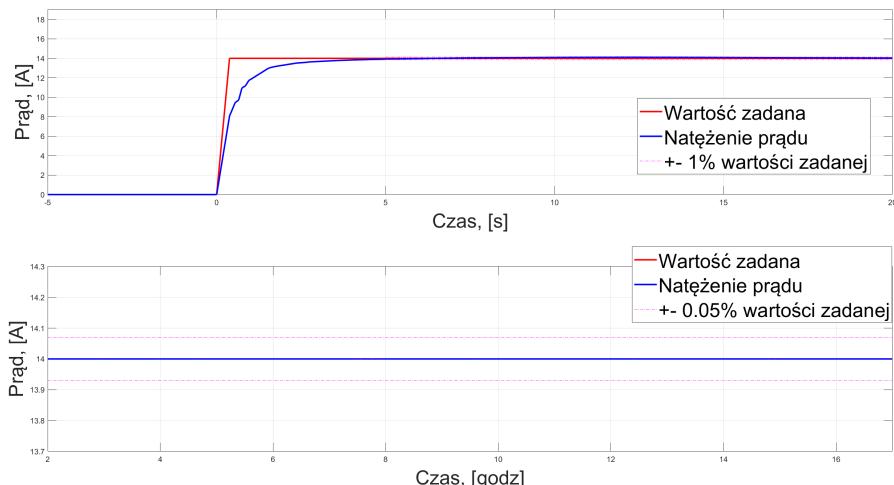
Układ poprawnie stabilizuje się w trybie stałej rezystancji. Regulator skutecznie ustala i utrzymuje wartość obciążenia w zadany trybie pomiarowy z bardzo niewielkimi odchyłkami.

6.4 Badanie charakterystyki w trybie CC (akumulator 12 V – 286 Ah).

Ostatnim badaniem było rozładowanie akumulatora żelowego 12 V pojemności 286 Ah w trybie obciążenia – CC. Prąd ustawiony na obciążeniu wynosił 14 A. Warunkiem końcowym testu była granica minimalnego napięcia ustawionego na 11.2 V. Rejestracja danych odbywała się co 10 sekund, jednak pierwsza minuta pomiaru rejestrowała parametry co 100 ms, w celu dokładnej analizy stanu nieustalonego. Badanie trwało ok. 17 godzin.

6.4.1 Stan przejściowy.

W trybie stałego prądu, regulator PID skutecznie utrzymywał wartość prądu na poziomie 14 A przez cały czas trwania testu (Rys. 6.10). Czas przejściowy trwał około 4 sekundy, następnie osiągnięty został stan ustalony, w którym wartość prądu nie wykraczała poza błąd ± 1 mA. Rejestracja danych odbywała się z interwałem 10 sekund, pozwoliło to na dokładną obserwację dynamiki pracy obciążenia.



Rysunek 6.10: Przebiegi prądu obciążenia dla prądu 14 A.

6.4.2 Analiza stanu ustalonego.

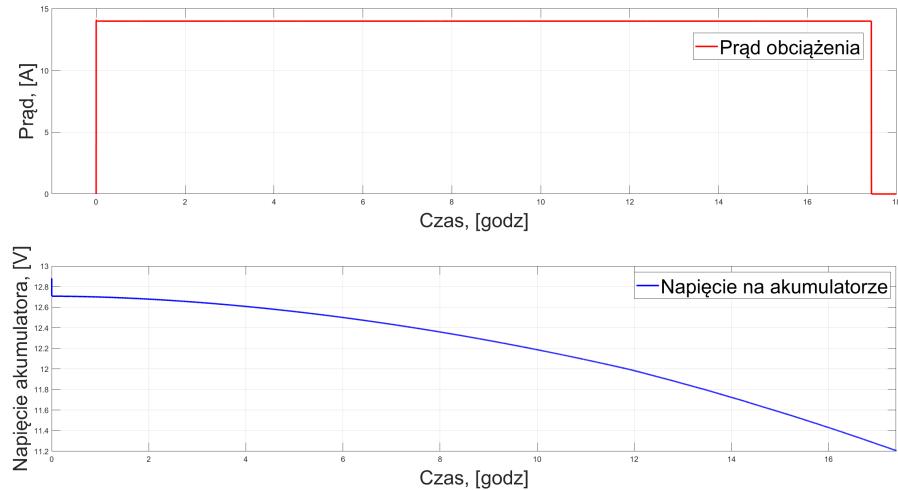
W celu analizy stanu ustalonego obliczone zostały parametry, których wzory znajdują się w podrozdziale 6.1.2.

Tabela 6.5: Wyliczone parametry dla trybu CP do analizy stanu ustalonego pomiarów.

<i>setpoint</i>	$\bar{x}, [A]$	$s_x, [A]$	$\Delta_{bez}, [A]$	$\Delta_{wzgl}, [\%]$
14A	14.0018	$1.24 \cdot 10^{-2}$	$2.2 \cdot 10^{-3}$	0.0155

6.4.3 Analiza charakterystyk rozładowania.

Z wykresu napięcia w funkcji czasu wynika typowe dla akumulatorów żelowych zachowanie: powolny spadek napięcia w czasie, przyspieszający pod koniec rozładowania akumulatora (Rys. 6.11).



Rysunek 6.11: Charakterystyka rozładowania akumulatora 12 V o pojemności 286 Ah.

Ładunek pobrany z akumulatora wskazany przez urządzenie to: 244 Ah.

Przeprowadzone badanie na akumulatorze o dużej pojemności potwierdziło poprawne działanie obciążenia, również w trakcie długotrwałej pracy oraz różnych warunkach prądowych. Regulator PID wykazał bardzo dobrą stabilność, prąd obserwowany jest bez widocznych oscylacji. Układ spełnił założone wymagania dotyczące precyzji. Tak długi test potwierdził także niezawodność i dobrą konstrukcję projektu – systemu pomiarowego, chłodzenia oraz oprogramowania.

Rozdział 7

Podsumowanie

7.1 Podsumowanie celu pracy.

Celem pracy było zaprojektowanie i stworzenie aktywnego obciążenia dedykowanego do testowania akumulatorów 12 V, mającego możliwość pracy w trzech trybach – stałego prądu (CC), stałej rezystancji (CR) i stałej mocy (CP). Dodatkowo urządzenie powinno komunikować się z komputerem PC, zapisywać dane pomiarowe na zewnętrznym nośniku pamięci oraz mieć wydajność prądową rzędu kilkunastu amperów. Testy akumulatorów miały polegać na rozładowywaniu ich zadanimi parametrami do określonych warunków końcowych.

Wszystkie te założenia udało się zrealizować, co potwierdzają testy przeprowadzone w rozdziale „Badania i pomiary”.

7.2 Realizacja i osiągnięcia.

W ramach pracy zaprojektowano system aktywnego obciążenia elektronicznego. Opracowano schematy elektryczne, zaprojektowano i wykonano płytę PCB oraz zrealizowano oprogramowanie mikrokontrolera STM32.

Urządzenie zostało wyposażone w system pomiarowy, oparty na precyzyjnych przetwornikach analogowo-cyfrowych, oraz tor sterowania z wykorzystaniem przetwornika cyfrowo-analogowego i tranzystora MOSFET w układzie regulacji PID. Zrealizowano trzy tryby pomiarowe – stałego prądu, rezystancji oraz mocy.

Zaprojektowano i wykonano panel sterowania złożony z wyświetlacza LCD, impulsatora oraz przycisków. Napisano również aplikację łączącą komputer PC z mikrokontrolerem, umożliwiającą zdальną zmianę parametrów pracy. Dodatkowo urządzenie wyposażone zostało w system zapisu danych na zewnętrznym nośniku pamięci w formacie CSV.

Przeprowadzone testy wykazały, że urządzenie spełnia założone parametry techniczne. Umożliwia stabilne utrzymywanie wartości zadanych w każdym z trybów pracy oraz po-

zwala na długotrwałe działanie, co umożliwia badanie akumulatorów o bardzo dużej pojemności.

Wykonane aktywne obciążenie cechuje się wysoką precyzją oraz stabilnością, co zostało potwierdzone podczas wykonanych badań. Zaletą takiego rozwiązania jest niski koszt wykonania w porównaniu do komercyjnych urządzeń, a także jego architektura, pozwalająca na modyfikacje i dalszy rozwój. Urządzenie może być stosowane w celu sprawdzania stanu technicznego akumulatorów oraz przy badaniach pracy ogniw 12 V.

Projekt może zostać rozszerzony o dodatkowe możliwości, takie jak:

- zwiększenie zakresu napięć wejściowych,
- zwiększenie ilości czujników temperatury,
- integracja układu z innym tranzystorem, dającym większe możliwości w zakresie prądowym,
- zwiększenie kompetencji aplikacji w celu odbioru danych pomiarowych z mikrokontrolera, dla analizy w czasie rzeczywistym.

W projekcie udało się połączyć elementy elektroniki analogowej i cyfrowej, systemów pomiarowych, automatycznej regulacji oraz oprogramowania mikrokontrolerów. Efekt końcowy jest zadowalający, obciążenie nadaje się do stosowania w realnych warunkach pracy oraz stanowi dobrą bazę do dalszego rozwoju.

Bibliografia

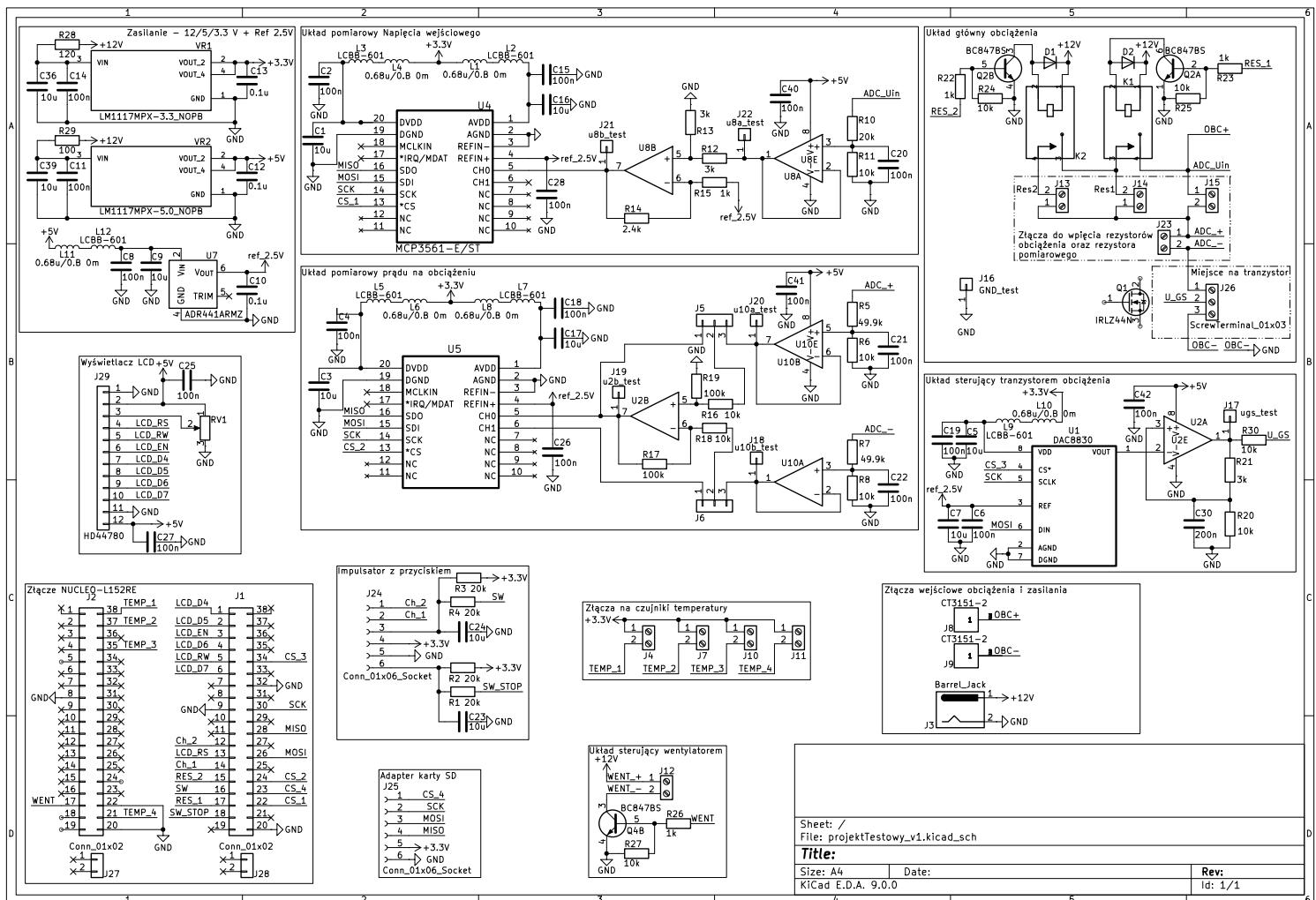
- [1] Paul Horowitz i Winfield Hill, *Sztuka elektroniki*, WKiŁ, 2018. ISBN 978-83-206-1992-8.
- [2] Obciążenie elektroniczne – co to jest i do czego służy? <https://www.amt.pl/p1/nws/obciaszenie-elektroniczne-co-to-jest-i-do-czego-sluzy> (dostęp 24.06.2025).
- [3] Microchip, *MCP3561 – 24-Bit Delta-Sigma ADC with Integrated PGA*, <https://www.alldatasheet.pl/datasheet-pdf/pdf/1318854/MICROCHIP/MCP3561.html>, datasheet, (dostęp: 08.07.2025).
- [4] Texas Instruments, *DAC8830 – 16-Bit Serial Input, Voltage Output DAC*, <https://www.alldatasheet.pl/datasheet-pdf/pdf/461926/TI1/DAC8830-EP.html>, datasheet, (dostęp: 08.07.2025).
- [5] International Rectifier, *IRLZ44N – Logic-Level Power MOSFET*, <https://www.alldatasheet.pl/datasheet-pdf/pdf/68872/IRF/IRLZ44N.html>, datasheet, (dostęp: 08.07.2025).
- [6] KiCad Documentation, *KiCad EDA Software Suite*, wersja 6.0, 2022.
- [7] STMicroelectronics, *STM32CubeIDE User Manual*, wersja 1.10, 2023.
- [8] Analog Devices, *LTspice XVII User Guide*, 2021.
- [9] Obciążenie elektroniczne 200 W DC RS-232. <http://www.biall.com.pl/item,AR-RAY-3722A-obciaszenie-elektroniczne-200W-DC-RS232,114514.html> (dostęp 24.06.2025).
- [10] Dokumentacja techniczna sterownika wyświetlacza LCD. <https://www.alldatasheet.pl/datasheet-pdf/view/63673/HITACHI/HD44780.html>, (dostęp 08.07.2025).
- [11] Åström K.J., Hägglund T., *PID Controllers: Theory, Design, and Tuning*, ISA (2006).
- [12] Practical process control, <https://controlguru.com/>, (dostęp 08.07.2025).

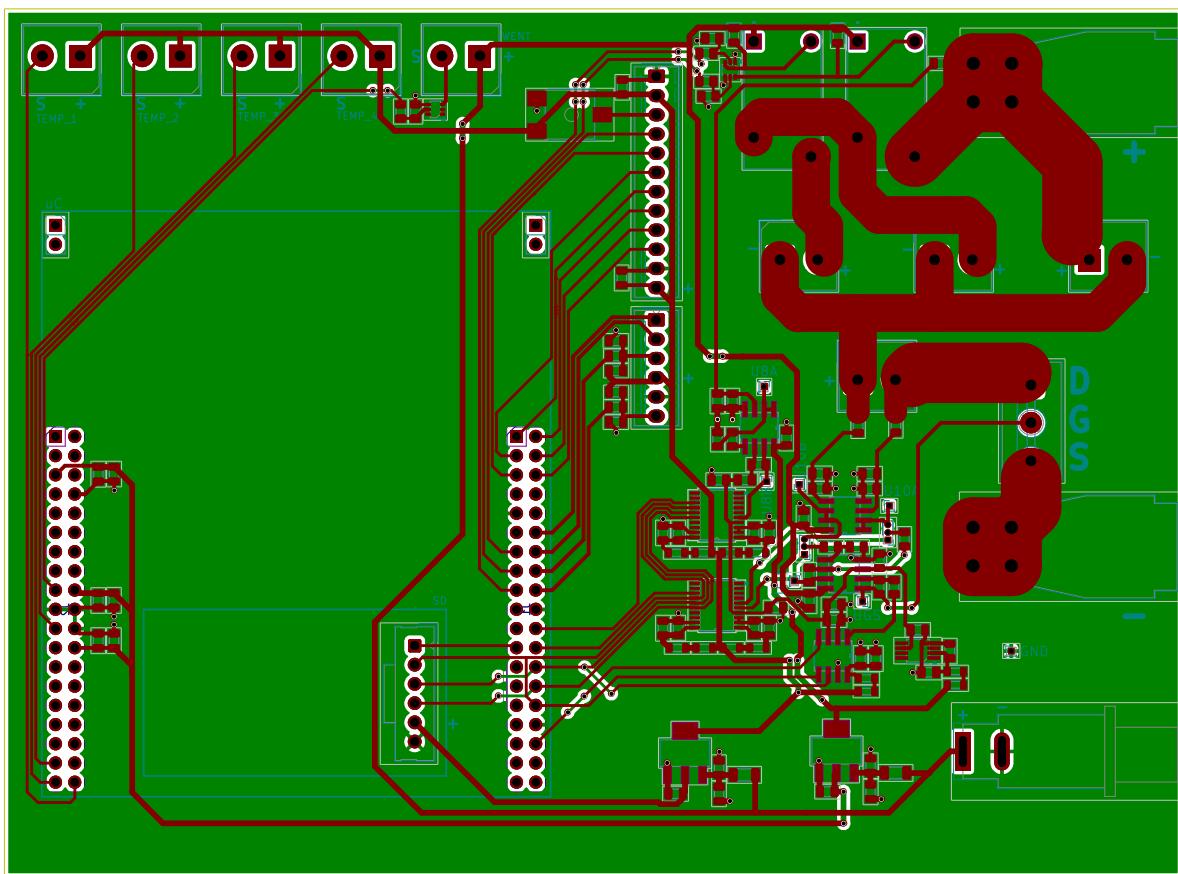
[13] Kurs regulacji PID, <https://iautomatyka.pl/category/wiedza/kurs-automatyki/regulacja-pid/>, (dostęp 08.07.2025).

,IEEE PID Control of Nonlinear Systems, <https://ieeexplore.ieee.org/document/9028342>, (dostęp 08.07.2025).

Dodatki

Dokumentacja techniczna





Spis skrótów i symboli

ADC przetwornik analogowo-cyfrowy (z ang. Analog-to-Digital Converter)

Ah amperogodzina (z ang. Ampere hour)

CC stały prąd (z ang. Constant Current)

CP stała moc (z ang. Constant Power)

CR stała rezystancja (z ang. Constant Resistance)

CSV format plików tekstowych z danymi (z ang. Comma Separated Values)

CV stałe napięcie (z ang. Constant Voltage)

DAC przetwornik cyfrowo-analogowy (z ang. Digital-to-Analog Converter)

DUT urządzenie badane (z ang. Device Under Test)

D dren (elektroda tranzystora)

ϕ przekrój

G bramka (elektroda tranzystora, z ang. Gate)

HAL Biblioteka na STM32 (z ang. Hardware Abstraction Layer)

I prąd

IP65 stopień ochrony obudowy (pyłoszczelność i odporność na strumień wody)

LCD wyświetlacz ciekłokrystaliczny (z ang. Liquid Crystal Display)

LL Biblioteka STM32 (z ang. Low Layer)

N liczba pomiarów

n numer pomiaru

NTC ujemny współczynnik temperaturowy (z ang. Negative Temperature Coefficient)

OSR współczynnik nadpróbkowania (z ang. Oversampling Ratio)

P_{tot} całkowita moc strat

P_{rob} moc wydzielana na rezystorach obciążenia

P_{tranz} moc wydzielana na tranzystorze obciążenia

PC komputer osobisty (z ang. Personal Computer)

PCB płytka drukowana (z ang. Printed Circuit Board)

R rezystancja

RTC zegar czasu rzeczywistego (z ang. Real-Time Clock)

S źródło (elektroda tranzystora, z ang. Source)

s_x odchylenie standardowe

SD karta pamięci (z ang. Secure Digital)

SPI interfejs szeregowy (z ang. Serial Peripheral Interface)

SPICE program do symulacji układów elektronicznych (z ang. Simulation Program with Integrated Circuit Emphasis)

U napięcie

U_{gs} napięcie bramka-źródło tranzystora

UART asynchroniczny interfejs szeregowy (z ang. Universal Asynchronous Receiver-Transmitter)

USB uniwersalna magistrala szeregowa (z ang. Universal Serial Bus)

\bar{x} wartość średnia

x_n n-ty pomiar

$x_{setpoint}$ wartość zadana

$\overline{\Delta_{bez}}$ średni błąd bezwzględny

$\overline{\Delta_{wzg}}$ średni błąd względny

Lista dodatkowych plików

W systemie do pracy dołączono dodatkowe pliki zawierające:

- projekt programu na NUCLEO-L152RE,
- projekt aplikacji napisanej w C#,
- pliki CSV z danymi pomiarowymi pobranymi z urządzenia.

Spis rysunków

2.1 Schemat blokowy typowego obciążenia aktywnego.	6
3.1 Fragment schematu elektrycznego przedstawiający układ główny obciążenia.	13
3.2 Zrzut ekranu przedstawiający symulację pracy tranzystora IRLZ44N.	13
3.3 Wykres przedstawiający napięcie tranzystora IRLZ44N bramka-źródło od prądu na drenie.	14
3.4 Fragment wykresu $U_{GS}(I_D)$ tranzystora w początku pracy obciążenia.	14
3.5 Fragment wykresu $U_{GS}(I_D)$ tranzystora w granicach planowanej pracy obciążenia.	14
3.6 Fragment schematu przedstawiający układ pomiarowy napięcia wejściowego.	18
3.7 Symulacja pracy układu skalującego napięcie (z 10.5/15 V na 0/2.5 V) oraz wykres z programu LTspice.	18
3.8 Fragment schematu przedstawiający układ pomiarowy natężenia prądu w obciążeniu.	19
3.9 Symulacja układu pomiarowego dla maksymalnego prądu i napięcia wejściowego.	19
3.10 Symulacja układu pomiarowego dla minimalnego prądu i maksymalnego napięcia wejściowego.	19
3.11 Symulacja pracy LMV932 w konfiguracji wzmacniacza nieodwracającego oraz wykres zależności napięcia wyjściowego od wejściowego wzmacniacza.	20
3.12 Fragment schematu elektrycznego układu sterującego tranzystorem obciążenia.	21
3.13 Fragment schematu elektrycznego przedstawiającego układ sterowania wentylatora.	22
3.14 Fragmenty schematu elektrycznego przedstawiające układ panelu sterowania użytkownika.	23
3.15 Fragment schematu elektrycznego przedstawiający układ zapisu danych na zewnętrznym nośniku pamięci.	23
3.16 Fragment schematu elektrycznego przedstawiającego układ zasilania.	25
4.1 Projekt płytki drukowanej z podziałem na sekcje.	28

4.2	Urządzenie po otwarciu obudowy.	32
4.3	Złożony projekt aktywnego obciążenia.	32
5.1	Graficzne przedstawienie struktury danych wykorzystywanej w programie obciążenia.	35
5.2	Fragment kodu definiującego strukturę danych z Rys.5.1.	35
5.3	Przykładowe funkcje <i>set()</i> i <i>get()</i>	35
5.4	Przykładowe funkcje <i>define</i>	36
5.5	Fragment kodu pliku „mystruct.c” odpowiadający za inicjalizację przetwornika ADC.	38
5.6	Fragment pliku „mystruct.c” przedstawiający realizacje pomiaru napięcia wejściowego.	38
5.7	Fragment pliku „mystruct.c” przedstawiający odczyt natężenia prądu.	39
5.8	Fragment funkcji <i>setCurrent()</i> przedstawiający realizacje zmiany rezystancji obciążenia oraz aktualizację parametru ładunku wyprowadzonego z akumulatora (linia 128).	39
5.9	Fragment kodu zawierający funkcje obsługujące DS18B20.	40
5.10	Fragment programu zawierający kod przerwania wywołującego się co 50 ms.	41
5.11	Funkcja regulatora PID.	43
5.12	Funkcja <i>compensate()</i>	43
5.13	Kod funkcji pobierającej dane z RTC w celu stworzenia nazwy folderu.	45
5.14	Fragment funkcji realizującej zapis danych pomiarowych na kartę SD.	45
5.15	Funkcje obsługujące komunikacje z wyświetlaczem LCD.	46
5.16	Ekrany wyświetlacza będącego częścią interfejsu użytkownika.	47
5.17	Fragment kodu funkcji realizującej pracę impulsatora.	47
5.18	Graficzne przedstawienie ramki danych służącej do komunikacji z komputerem PC.	48
5.19	Funkcja <i>saveDataToStruct()</i> zapisująca parametry odebrane przez UART do głównej struktury.	48
5.20	Funkcja <i>checkForData()</i> - wywoływana w głównej pętli.	49
5.21	Funkcja EEPROM_Save_Partial służąca do zapisu ważnych parametrów w pamięci EEPROM.	49
5.22	Funkcje <i>set()</i> i <i>get()</i> obsługujące zmienną <i>parameter</i> w strukturze danych. .	50
5.23	Realizacja warunków kończących pracę urządzenia.	51
5.24	Prezentacja aplikacji do konfigurowania parametrów obciążenia.	52
6.1	Przebiegi prądowe obciążenia dla prądów: 360 mA / 700 mA / 1.4 A.	54
6.2	Przebiegi prądowe obciążenia dla prądu: 360 mA.	55
6.3	Przebiegi prądowe obciążenia dla prądu: 700 mA.	56
6.4	Przebiegi prądowe obciążenia dla prądu: 1.4 A.	56

6.5 Charakterystyka rozładowania akumulatora 12 V dla pełnego zakresu czasowego.	57
6.6 Przebiegi mocy obciążenia dla mocy 35 W.	58
6.7 Charakterystyka rozładowania akumulatora 12 V w trybie CP.	59
6.8 Przebiegi rezystancji obciążenia dla rezystancji 4Ω	60
6.9 Charakterystyka rozładowania akumulatora 12 V w trybie CR.	60
6.10 Przebiegi prądu obciążenia dla prądu 14 A.	61
6.11 Charakterystyka rozładowania akumulatora 12 V o pojemności 286 Ah.	62

Spis tabel

2.1	Porównanie modeli obciążeń różnych firm o podobnych parametrach.	5
3.1	Tabela przedstawiająca obliczenia rezystancji obciążenia.	15
6.1	Wyliczone parametry do analizy stanu ustalonego pomiarów.	56
6.2	Wartość ładunku wyprowadzonego z akumulatora 12 V w zależności od obciążenia.	57
6.3	Wyliczone parametry dla trybu CP do analizy stanu ustalonego pomiarów.	58
6.4	Wyliczone parametry dla trybu CP do analizy stanu ustalonego pomiarów.	60
6.5	Wyliczone parametry dla trybu CP do analizy stanu ustalonego pomiarów.	61