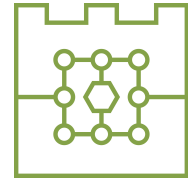




**Politechnika Krakowska
im. Tadeusza Kościuszki**

Wydział Informatyki i Telekomunikacji



Bartłomiej Banaś

Numer albumu: 117971

**Opracowanie systemu informatycznego do kontroli
jakości nadprzewodników wysokotemperaturowych**

**Implementation of high temperature superconductors
software test environment**

**Praca magisterska
na kierunku INFORMATYKA STOSOWANA**

Praca wykonana pod kierunkiem:
dr Radosław Kycia

Recenzent pracy:
dr hab. inż. Maciej Jaworski, prof. PK

Kraków 2022

Spis treści

STRESZCZENIE.....	4
ABSTRACT.....	4
1. WSTĘP.....	5
1.1. CEL	5
1.1. ZAKRES PRACY.....	5
1.2. METODYKA	5
2. CZĘŚĆ TEORETYCZNA:.....	5
2.1. WIELKI ZDERZACZ HADRONÓW	6
2.2. NADPRZEWODNICTWO I PODZIAŁ NADPRZEWODNIKÓW.....	10
2.3. NADPRZEWODNIKI WYSOKIEJ TEMPERATURY NOWEJ GENERACJI	12
2.4. DEGRADACJA WŁAŚCIWOŚCI NADPRZEWODZENIA	13
2.5. ROZWIĄZANIA PROGRAMOWE WYSOKIEJ NIEZAWODNOŚCI.....	16
2.6. SYSTEMY REAL-TIME	16
2.7. PRZYKŁADOWA ARCHITEKTURA OPROGRAMOWANIA SYSTEMU CZASU RZECZYWISTEGO	17
2.8. SYSTEMY FPGA	19
2.9. WPROWADZENIE DO LABVIEW.....	20
3. CZĘŚĆ PRAKTYCZNA	21
3.1. WPROWADZENIE DO PROJEKTU	21
3.2. DEFINICJA WYMAGAŃ.....	22
3.3. DOBÓR PLATFORMY SPRZĘTOWEJ.....	23
3.4. ARCHITEKTURA, KOMUNIKACJA.....	24
3.5. APLIKACJA UŻYTKOWNIKA.....	25
3.5.1. <i>Funkcjonalności</i>	25
3.5.2. <i>Struktura kodu</i>	27
3.5.3. <i>Komunikacja pomiędzy wątkami</i>	28
3.6. MODUŁ CZASU RZECZYWISTEGO	29
3.6.1. <i>Funkcjonalności</i>	29
3.6.2. <i>Struktura kodu</i>	30
3.6.3. <i>Komunikacja pomiędzy wątkami</i>	32
3.7. FPGA	32
3.7.1. FUNKCJONALNOŚCI	32
3.7.2. STRUKTURA PROGRAMU.....	33
3.7.3. KOMUNIKACJA POMIĘDZY PĘTLAMI.....	33
3.7.4. PROBLEM WYZNACZANIA ROZWIĄZANIA NIELINIOWYCH ZALEŻNOŚCI NAPIĘCIE-TEMPERATURA DLA TERMOPARY ...	34
3.7.5. PROBLEM SKALOWALNOŚCI OPROGRAMOWANIA.....	37
4. WALIDACJA, TESTY ORAZ WDRÓŻENIE DO PRODUKCJI	39
5. PODSUMOWANIE	40
6. BIBLIOGRAFIA	41
7. SPIS RYSUNKÓW, TABEL, WYKRESÓW.....	43

Streszczenie

Na potrzeby projektu modernizacji przyłączy wysokiej mocy dla wielkiego zderzacza hadronów LHC, konieczne jest testowanie wysokotemperaturowych nadprzewodników, które dostarczane są do ośrodka CERN (Europejskiego Centrum Badań Jądrowych) od komercyjnych dostawców. Konieczność weryfikacji wiąże się ze zjawiskiem degradacji właściwości nadprzewodnictwa materiału, po ekspozycji na wysoką temperaturę. Proces technologiczny wymaga lutowania przewodów a co za tym idzie – nagrzewania. Z tego względu zaistniała potrzeba implementacji automatycznego środowiska testowego wykonującego testy temperaturowe. Przygotowuje próbki do testów mających na celu weryfikację stopnia degradacji jakości po procesie.

Niniejsza praca ma na celu przybliżenie metodyki implementacji systemów produkcyjnych z wykorzystaniem środowiska LabVIEW, LabVIEW Real-Time oraz LabVIEW FPGA. Część teoretyczna dotyka samych zagadnień implementacyjnych, natomiast część praktyczna opisuje zaimplementowany już system, który jest obecnie wdrażany do produkcji.

Abstract

For the project to upgrade the superconducting links for the LHC large hadron collider, it is necessary to test high-temperature superconductors that are delivered to the CERN facility from commercial suppliers. The need for verification is related to the phenomena of degradation of the material's superconductivity properties after exposure to high temperature. The technological process requires soldering the wires and thus - heating. For this reason, there was a need to implement an automatic test environment that functionally executes temperature tests, which prepares samples for tests aimed at verifying the degree of quality degradation after process of heating.

This paper aims to introduce the implementation methodologies of production systems using the LabVIEW, LabVIEW Real-Time and LabVIEW FPGA environments. The theoretical part concerns the implementation issues, while the practical part describes the already implemented system, which is currently being deployed into production.

1. Wstęp

1.1. Cel

Głównym celem pracy jest zaprojektowanie oraz oprogramowanie systemu informatycznego wykonującego testy temperaturowe próbek materiału, z którego wytwarzane są nadprzewodniki wysokotemperaturowe. Precyzyjne wymagania dotyczące samego systemu zostały opisane w podrozdziale 3.2.

1.1. Zakres pracy

Praca składa się z dwóch części. Część teoretyczna opisuje trudności związane z produkcją i obsługą elektromagnesów wysokiej mocy jak i wprowadza do zagadnień stricte informatycznych, które pozwalają na implementację rozwiązań przemysłowych wysokiej niezawodności. Część praktyczna zawiera opis zarówno architektury i implementacji systemu w środowisku LabVIEW, jak i kwestii istotnych z informatycznego punktu widzenia podczas projektowania podobnych systemów.

1.2. Metodyka

Praca w głównej mierze opisuje rozwiązania programistyczne środowiska LabVIEW, skupiając się uniwersalnych kwestiach dotyczących zarówno architektury jak i konkretnych zagadnieniach implementacyjnych.

2. Część teoretyczna:

W tym rozdziale opisane zostaną aspekty teoretyczne związane z wykorzystaniem nadprzewodników w akceleratorach, wprowadzając potrzebną terminologię wymaganą podczas opracowania systemu do kontroli procesu wytwarzania nadprzewodników. Materiał zostanie przedstawiony w wersji uproszczonej, gdyż głównym tematem pracy jest omówienie samego oprogramowania informatycznego. Następnie przedstawione zostaną zagadnienia techniczne, wprowadzające do technicznych rozwiązań wykorzystanych podczas implementacji.

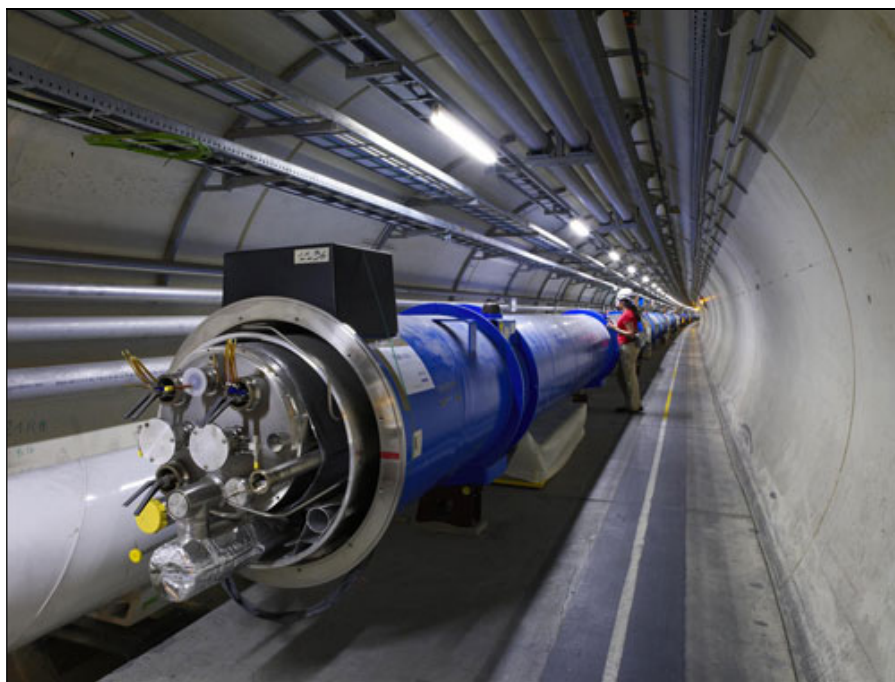
Aspekty fizyczne procesu produkcji

2.1. Wielki Zderzacz Hadronów

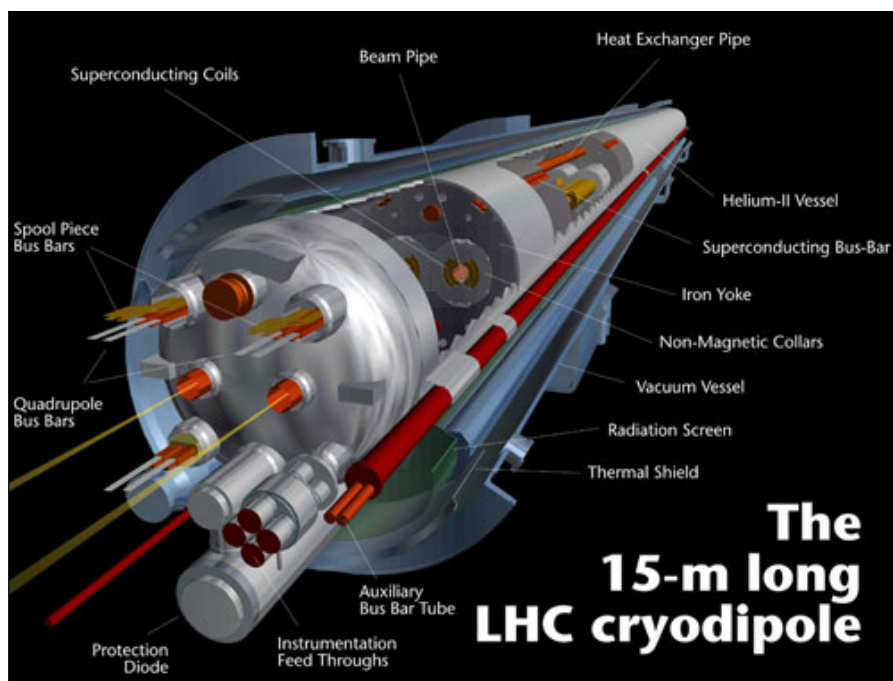
Wielki Zderzacz Hadronów (ang. *Large Hadron Collider - LHC*) rozpędza naładowane cząstki – protony i jądra atomowe w celu późniejszego zderzenia ich z odpowiednio wysoką energią kinetyczną. Cząstki zanim dotrą do LHC są przyspieszane w serii połączonych ze sobą akceleratorów liniowych i kołowych. Gdy osiągną maksymalną prędkość w danym stopniu przyspieszającym, wiązka wprowadzana jest do kolejnego stopnia aż do finalnego pierścienia LHC. Aby prowadzić naładowane cząstki elementarne po skomplikowanych ścieżkach bez utraty prędkości, jest potrzebnych ponad 50 rodzajów magnesów.

Wszystkie magnesy w LHC są elektromagnesami. Elektromagnesy służą do wytwarzania pola magnetycznego w wyniku przepływu prądu elektrycznego przez cewki. Główne magnesy dwubiegunowe – dipole - generują silne pola magnetyczne o natężeniu 8,3 tesli. Magnesy dipolowe, to jedne z najbardziej złożonych elementów LHC - służą do zaginania drogi cząstek. Istnieje 1232 głównych dipoli, każdy o długości 15 metrów i wadze 35 ton [1]. Elektromagnesy wykorzystują do wytworzenia pola prąd o natężeniu 11080 amperów. Wykorzystana nadprzewodząca cewka umożliwia przepływ wysokich prądów bez utraty energii ze względu na rezystancję. Silne pole magnetyczne generowane przez magnesy dipolowe pozwala na większe zakrzywienie toru wiązki, przez co akcelerator może mieć mniejsze rozmiary niż analogiczny akcelerator bez elektromagnesów.

Po rozpędzeniu cząstek elementarnych są one zderzane w miejscach otoczonych przez zestawy detektorów rejestrujących produkty tych zderzeń. Takie detektory są rozmieszczone na obwodzie pierścienia LHC. Kiedy cząstki są zgrupowane, istnieje większe prawdopodobieństwo, że zderzą się w większej liczbie gdy dotrą do detektorów. W tym celu używane są magnesy czterobiegunowe. Kwadrupole pomagają utrzymać cząstki elementarne w ciasnej wiązce. Posiadają cztery bieguny magnetyczne ułożone symetrycznie wokół rury wiązki, aby ścisnąć wiązkę w pionie lub w poziomie. Dipole są również wspierane przez magnesy sześć-, ośmio- i dziesięcio-biegunowe, które korygują drobne niedoskonałości pola magnetycznego na ich krańcach. Rysunki poniżej przedstawiają zarówno sam elektromagnes jak i schematyczny opis jego wnętrza [1].

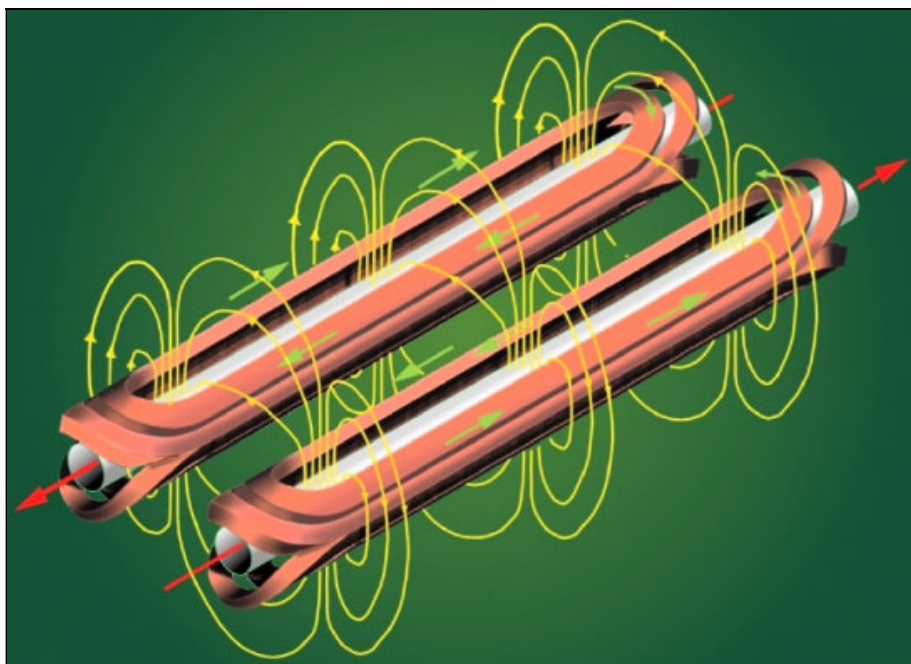


Rysunek 2-1. Elektromagnes dwubiegunowy w tunelu. [2]

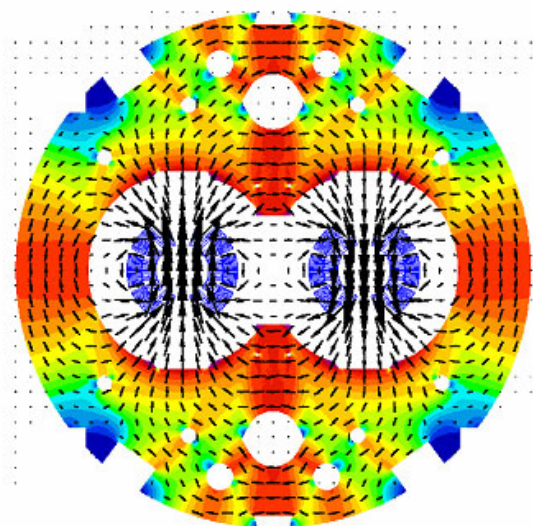


Rysunek 2-2. Schematyczne przedstawienie magnesu dipolowego LHC [2]

Magnesy dipolowe wytwarzają pole magnetyczne pomiędzy biegunami. Na umieszczone w polu magnetycznym naładowane cząstki działa wówczas siła Lorentza, powodująca ich rozpędzanie. Rozkład pola magnetycznego w obrębie elektromagnesu pozwala na wprowadzenie dwóch strumieni naładowanych cząstek, rozpędzanych w przeciwnych kierunkach. Rysunek 3-3 poglądowo przedstawia budowę krytycznej części elektromagnesu.

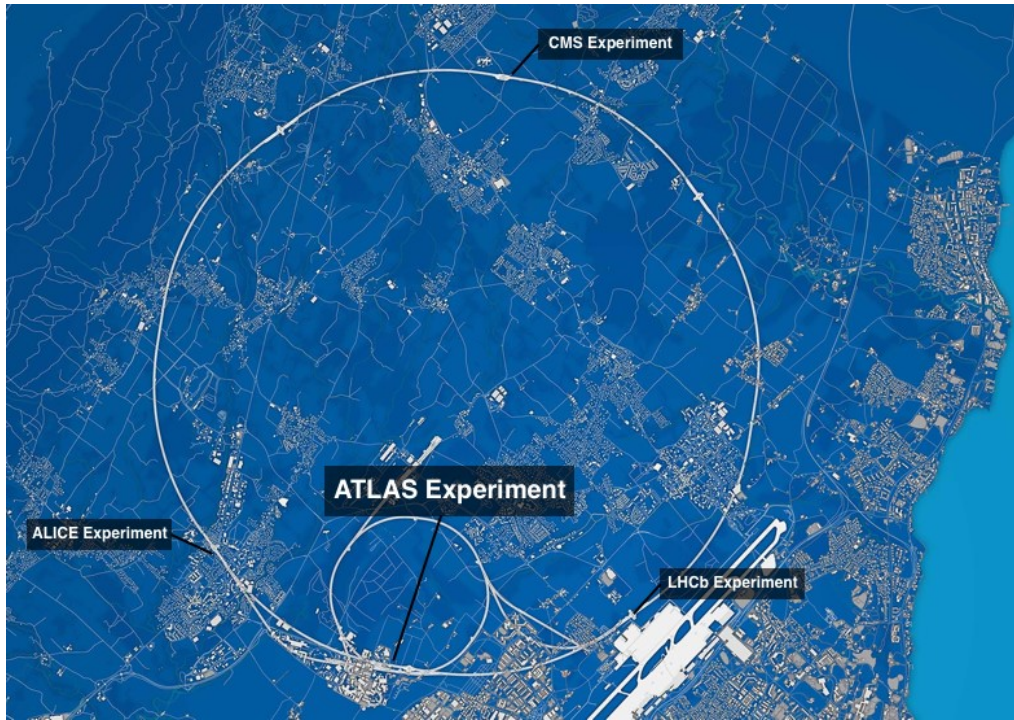


Rysunek 2-3. Linie pola magnetycznego dipola wokół dwóch rur wiązki [2]

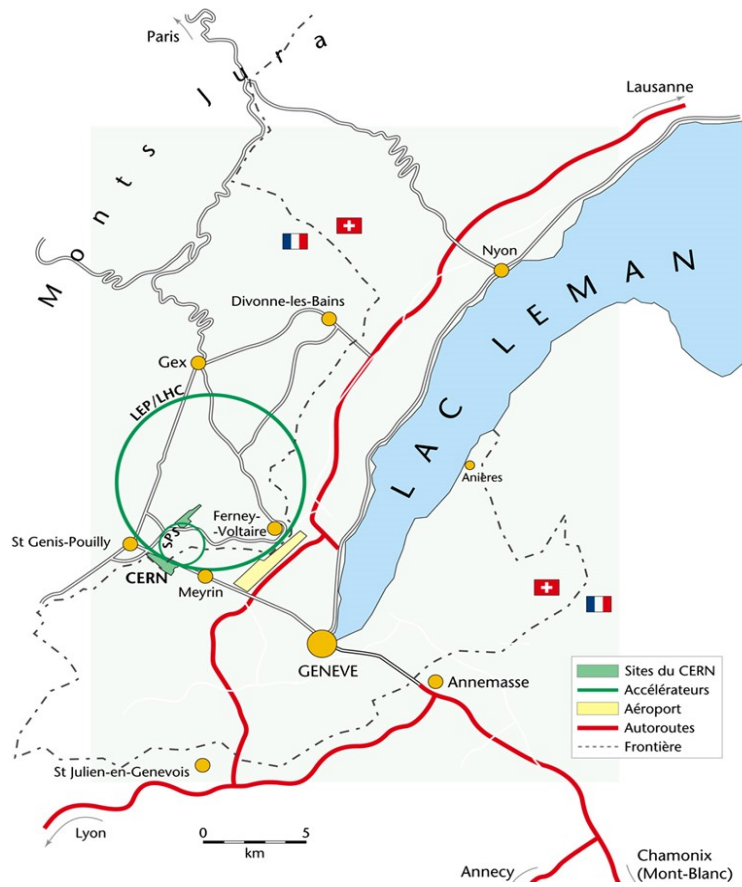


Rysunek 2-4. Rozkład linii pola w przekroju [2]

Kiedy wiązki cząstek dotrą do detektorów, kierowane są dalej przez magnesy wprowadzające (ang. *insertion magnets*). W celu zwiększenia szansy zderzenia cząstek elementarnych pochodzących z dwóch kierunków, promień jest skupiany przez magnesy. Cząstki muszą być ściśnięte bliżej siebie, zanim trafią do detektora, aby zwiększyć szansę zderzenia z cząstkami nadchodzącymi z przeciwnego kierunku. Trzy kwadrupole są używane do stworzenia systemu zwanego wewnętrzną trójką (ang. *inner tripler*). Istnieje osiem wewnętrznych trójek, z których dwie znajdują się w każdym z czterech dużych detektorów zderzacza- ALICE, ATLAS, CMS i LHCb (Rysunek 2-5).



Rysunek 2-5. Rozmieszczenie detektorów na obwodzie zderzacza.[3]



Rysunek 2-6. Położenie zderzacza [3].

Wspomniane magnesy skupiają wiązkę, dzięki czemu jest 12,5 razy węższa od pierwotnej – jej średnica to od 0,2 milimetra do 16 mikrometrów.

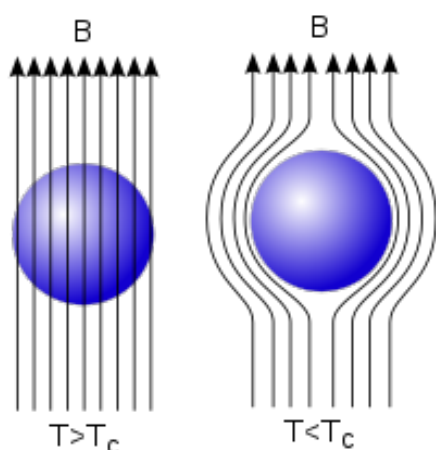
Po zderzeniu wiązki w detektorze ogromne magnesy wspomagają pomiar cząstek. Przykładowo, naładowane cząstki mogą zostać przeprowadzane przez pole magnetyczne które spowoduje odchyleniem trajektorii takiej cząstki od linii prostej. Wówczas, na bazie odchylenia trajektorii można obliczyć masę.

Po zderzeniu wiązki, cząstki są ponownie rozdzielane przez magnesy dipolowe. Kiedy nadchodzi czas na usunięcie cząstek z akceleratora, są one wyprowadzane wzdłuż linii prostej w kierunku obszaru służącego do zrzutu wiązki. W takim obszarze magnes „rozcieńczający” (ang. *dilution magnet*) zmniejsza intensywność wiązki o współczynnik 100000, zanim nastąpi zderzenie z blokiem kompozytu betonu i grafitu w celu ostatecznego zatrzymania.[1].

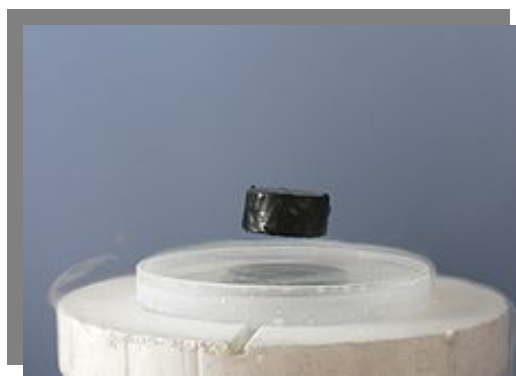
2.2. Nadprzewodnictwo i podział nadprzewodników

Poniżej pewnej „krytycznej” (specyficznej dla materiału) temperatury, materiały przechodzą w stan nadprzewodzący, charakteryzujący się dwiema podstawowymi właściwościami.

- Nie stawiają oporu na przepływ prądu elektrycznego. Gdy rezystancja spada do zera, prąd może krążyć wewnątrz materiału bez strat w postaci ciepła.
- Jeśli pole elektromagnetyczne w którym znajduje się nadprzewodnik jest wystarczająco słabe, wówczas nie przeniknie ono do wnętrza materiału, a pozostanie na jego powierzchni. To zjawisko wypychania pola magnetycznego znane jest jako efekt Meissnera, od nazwiska fizyka, który jako pierwszy zaobserwował je w 1933 roku.



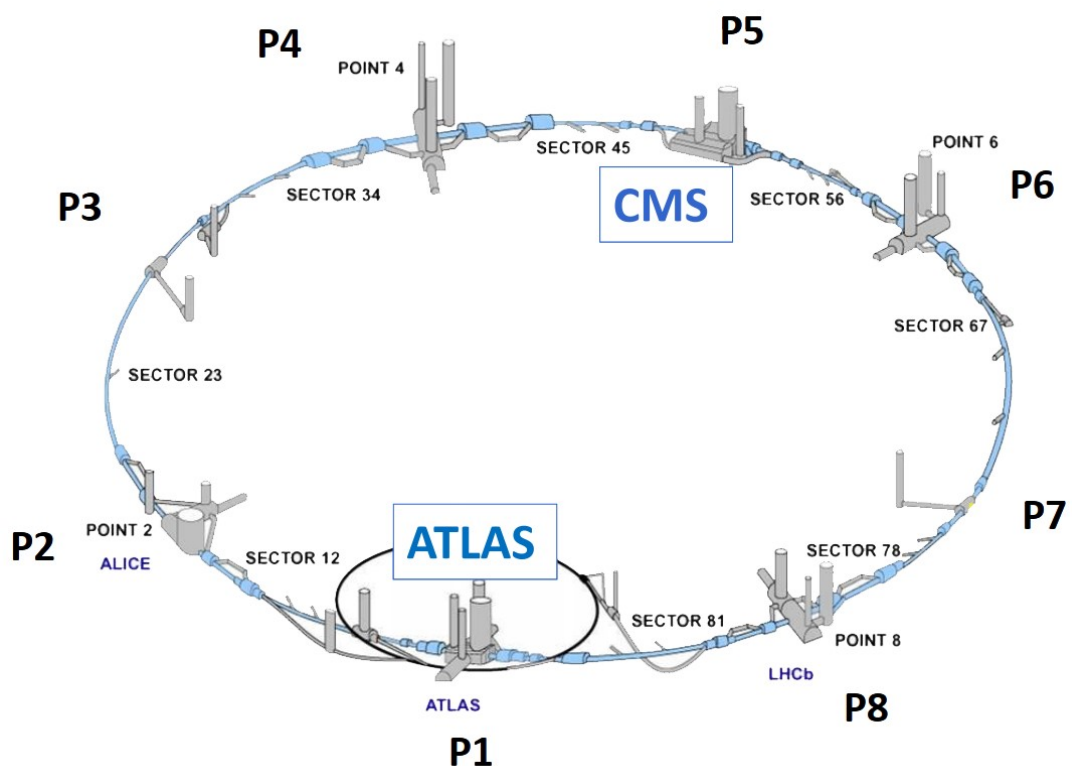
Rysunek 2-7. Rozkład pola magnetycznego w nadprzewodniku [4].



Rysunek 2-8. Zablockowany magnes ze względu na efekt Meissnera [4].

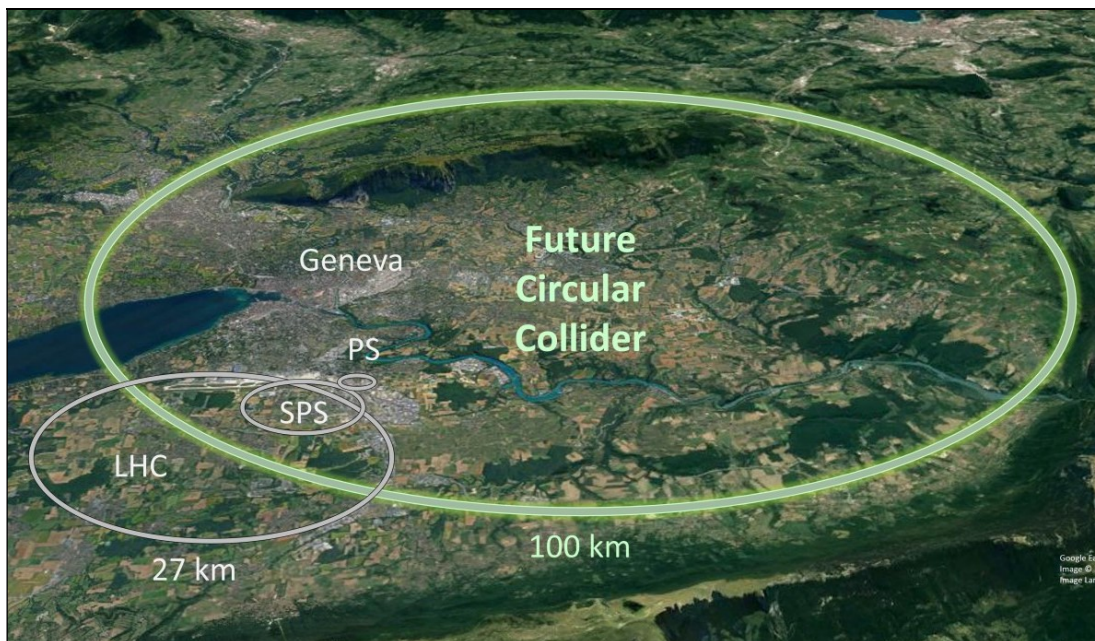
Materiały mogą zostać wyprowadzone ze stanu nadprzewodnictwa przez wzrost temperatury lub przyłożone zewnętrzne pole magnetyczne. Z tej perspektywy rozróżnia się dwa rodzaje nadprzewodników: Materiały typu I pozostają w stanie nadprzewodzącym tylko w przypadku wpływu stosunkowo słabych pól magnetycznych. Powyżej określonego progu pole gwałtownie wnika w materiał, rozbijając stan nadprzewodnictwa. Z drugiej strony nadprzewodniki typu II tolerują lokalną penetrację pola magnetycznego, co umożliwia im zachowanie właściwości nadprzewodzących w obszarze wpływu silnego pola magnetycznego. Nadprzewodniki typu II umożliwiły wykorzystanie nadprzewodnictwa w silnych polach magnetycznych, prowadząc między innymi do opracowania magnesów do akceleratorów cząstek [5].

Tak duża ilość elektromagnesów wymaga skutecznych metod ich zasilania. Technicznie energia doprowadzana jest do magnesów za pośrednictwem przyłączy wysokiej mocy (S.C. – ang. *Superconducting Links*), umieszczonych w ośmiu punktach na obwodzie zderzacza (rysunek 2-9). Przyłącza umieszczone są pod ziemią, w bezpośrednim sąsiedztwie akceleratora, ze względu na konieczność utrzymywania niskiej temperatury. Do przyłączy energia doprowadzana jest konwencjonalnymi przewodami, natomiast dalej stosowane są już nadprzewodniki [6].



Rysunek 2-9. Rozkład przyłączy wysokiej mocy. [6]

Elektromagnesy dipolowe zbudowane są ze związku tytanu z niobem - Nb_3Sn , który posiada optymalne właściwości nadprzewodzące w temperaturze roboczej 1.9 stopni Kelwina, możliwej do osiągnięcia przy wykorzystaniu systemu chłodzenia opartego na ciekłym helu. Schłodzenie do tak niskiej temperatury jest problematyczne z technicznego punktu widzenia [7]. Między innymi z tego względu trwają prace nad opracowaniem i wdrożeniem nadprzewodników nowej generacji, będących w stanie nadprzewodzenia w temperaturze 77 K, osiągananej przy wykorzystaniu ciekłego azotu. Wówczas, przyłącza mogą być instalowane na powierzchni a sam nadprzewodnik chłodzony ciekłym azotem stanowiłby część pośrednią doprowadzającą energię do elektromagnesów. Wykorzystanie ciekłego azotu jest zarówno technicznie prostsze, jak i zdecydowanie mniej kosztowne. Kolejną istotną korzyścią, jest możliwość rozwoju nadprzewodników, aby zwiększyć ich wydajność prądową na potrzeby rozwoju ośrodka i budowy zderzacza nowej generacji FCC (ang. *The Future Circular Collider*). Future Circular Collider, jest projektem będącym aktualnie w fazie planowania. Ma on zapewnić infrastrukturę dla zderzacza nowej generacji, będącego w stanie doprowadzić do zderzenia z energią 100TeV. Dla porównania, LHC jest w stanie wygenerować zderzenia o maksymalnej energii 14TeV [8].

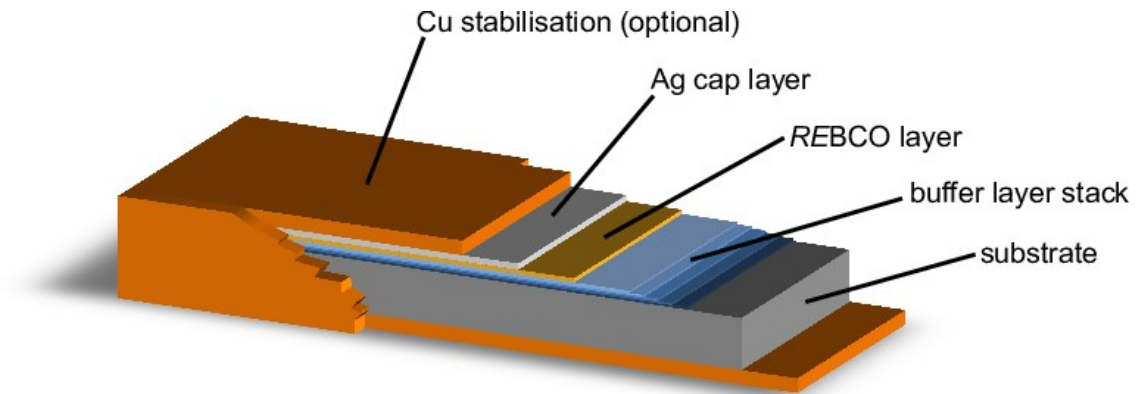


Rysunek 2-10 Mapa planowanego położenia Future Circular Collider [8]

2.3. Nadprzewodniki wysokiej temperatury nowej generacji

Wspomniany nadprzewodnik pracujący w temperaturze ciekłego azotu to HTS (ang. *High Temperature Superconductor*). Ponieważ nie ma właściwości pozwalających na użycie w otoczeniu silnych pól magnetycznych, może zostać z powodzeniem użyty jako przewód doprowadzający prąd do akceleratora.

Materiałem używanym jako nadprzewodnik jest w tym przypadku warstwa ReBCO (ang. *Rare-Earth Barium Cooper Oxide*). ReBCO to rodzina materiałów nadprzewodzących, w dosłownym tłumaczeniu - tlenek miedzi-baru-ziem rzadkich. Jako materiał nadprzewodzący najczęściej (również w przypadku omawianych nadprzewodników) wykorzystywany jest związek YBCO (ang. *yttrium barium copper oxide* – tlenek itrowo-barowo-miedziowy) Same przewody owijane są w taśmę, z której następnie wytwarzana jest wiązka ostatecznego nadprzewodnika [9]. Struktura taśmy HTS przedstawiona jest na rysunku 3-10.



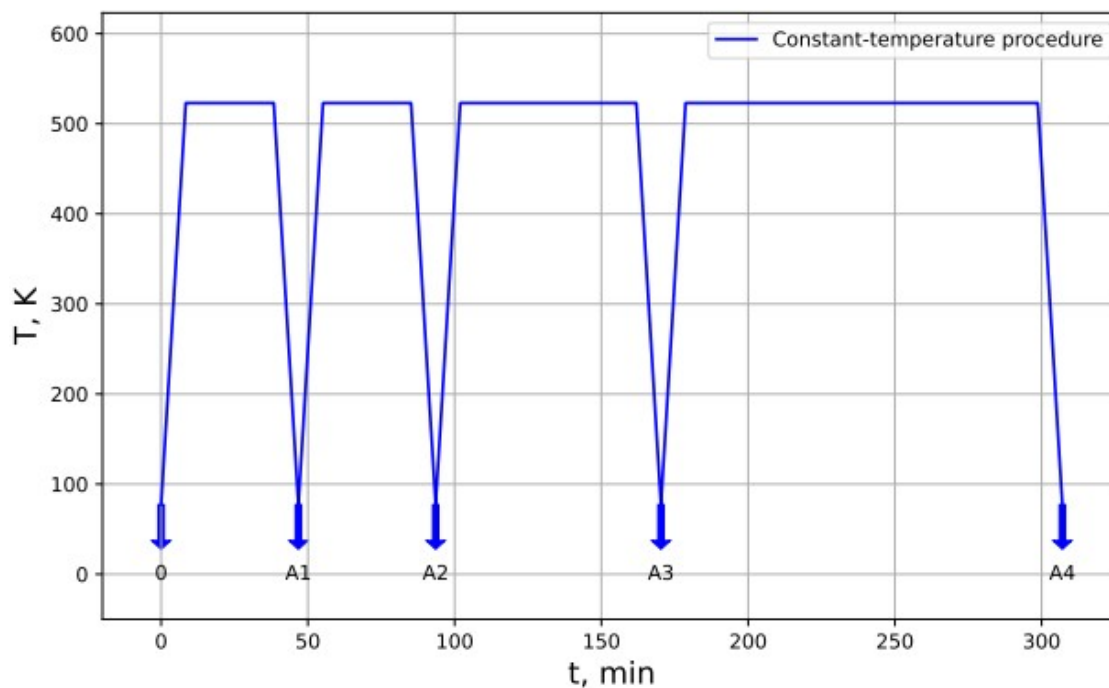
Rysunek 2-11 Struktura taśmy HTS [10]

Składa się ona z kilku warstw:

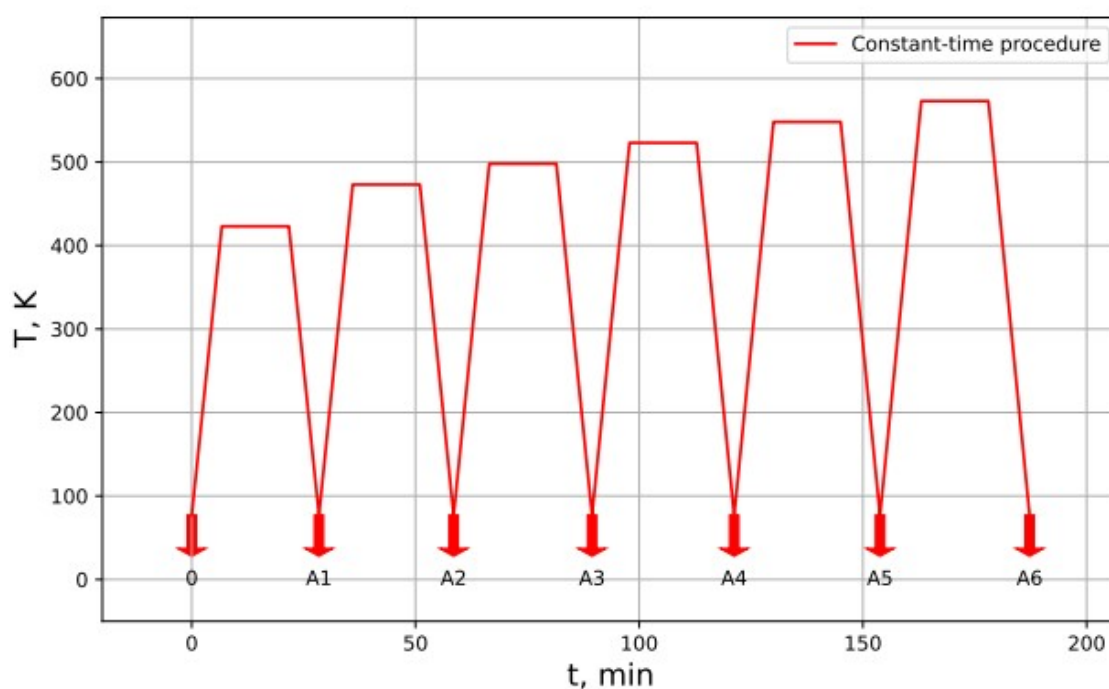
- Miedź (Cu stabilisation) – używana jest przede wszystkim w celu rozproszenia energii w wypadku, kiedy materiał przestaje być nadprzewodzący.
- Warstwa srebra (Ag cap) – warstwa ochronna pomiędzy ReBCO o warstwą miedzi,
- Warstwa ReBCO,
- Warstwa buforowa – częściowo zapobiega dyfuzji tlenu z warstwy ReBCO,
- Podłoże (substrate),

2.4. Degradacja właściwości nadprzewodzenia.

Proces instalacji sieci elektrycznej wymaga łączenia przewodów poprzez lutowanie czy spawanie, co powoduje nagrzewanie taśmy do względnie wysokiej temperatury, rzędu setek °C. Jednym z wyzwań dla procesu wytwarzania taśm HTS jest zjawisko degradacji właściwości nadprzewodnika przy ekspozycji na wysoką temperaturę. Podwyższona temperatura powoduje dyfuzję tlenu ze związku ReBCO. Zawartość tlenu znacząco wpływa na strukturę krystaliczną ReBCO i w efekcie, jej właściwości nadprzewodzenia. [11] Na rysunkach 3-11 oraz 3-12 przedstawione zostały przykładowe profile testów temperaturowych.



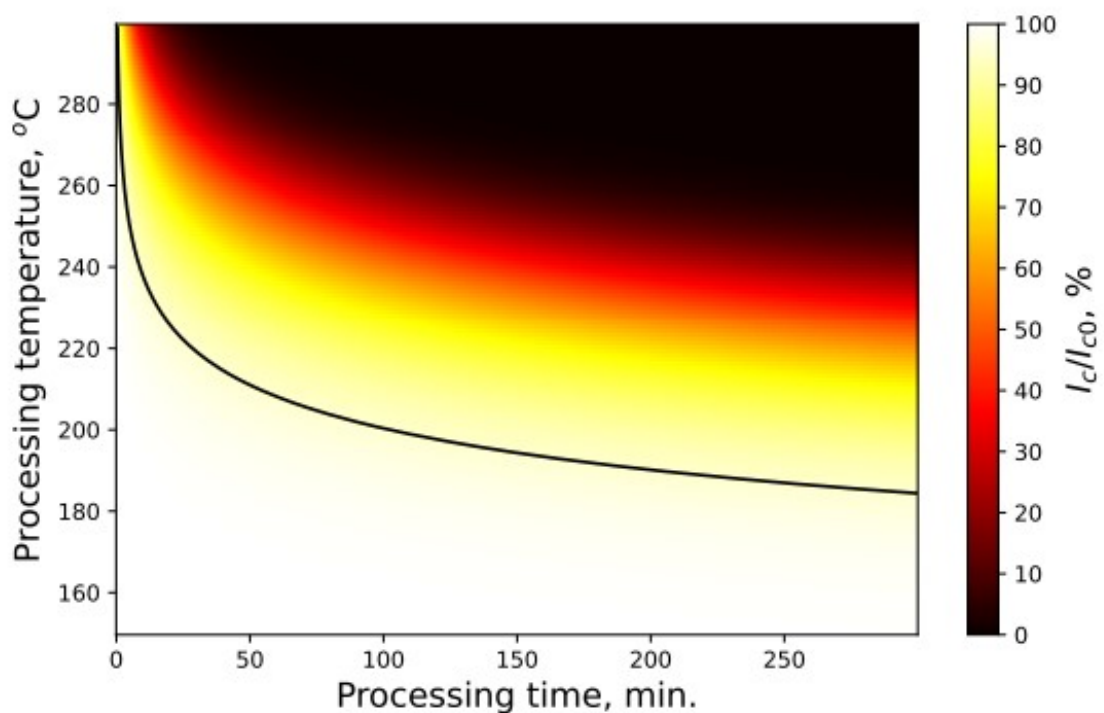
Rysunek 2-12. Przykładowy profil testów temperaturowych [11].



Rysunek 2-13. Przykładowy profil testów temperaturowych [11].

Kluczowym parametrem materiałów nadprzewodzących jest prąd krytyczny. Jest to wartość prądu, przy którym materiał traci właściwości nadprzewodzące i staje się rezystywny. Zjawisko to nazywane jest quench (ang). Zjawisko jest niebezpieczne, szczególnie w warunkach produkcyjnych gdy przez nadprzewodnik przepływa duży prąd.

Wówczas, ze względu na szybką dyssypację energii w postaci ciepła, system staje się niebezpieczny ze względu na możliwość rozszczelnienia zarówno rury akceleratora w której poruszają się cząstki, jak i systemu chłodzenia ciekłym helem. Właśnie z uwagi na ryzyko wystąpienia quenchingu, przewody zawierają znaczne ilości miedzi jako radiatora, który pochłonie nadmiar ciepła zanim system bezpieczeństwa odetnie dopływ prądu. Na rysunku 3-13 została przedstawiona zależność degradacji prądu krytycznego w zależności od czasu ekspozycji oraz wartości temperatury. Czarną linią została zaznaczona granica degradacji rzędu 5%. Oznacza to, że wartość prądu krytycznego w próbce po nagrzewaniu spadła o 5% względem wartości początkowej – przed procesem.



Rysunek 2-14. Degradacja temperaturowa taśm HTS [11].

W celu kontroli jakości taśm HTS stosowanych w przewodach prądowych HL-LHC próbki z dostarczonego materiału muszą zostać poddane precyzyjnemu cyklowi obróbki cieplnej. W zależności od ilości szpul i długości taśm, duża ilość próbek – około 3000 musi zostać poddana testom. Z tego względu zaistniała potrzeba implementacji automatycznego systemu testowego, któremu została poświęcona praktyczna część pracy.

Aspekty informatyczne procesu produkcji

2.5. Rozwiązania programowe wysokiej niezawodności

Systemy przemysłowe muszą wykazywać dużą odporność zarówno na warunki środowiskowe jak i wykazywać dużą niezawodność podczas pracy. Dodatkowo, system operacyjny na którym implementowane jest rozwiązanie sprzętowe musi być niezawodny i stabilny. Z tego względu, warto wspomnieć o kilku rozwiązaniach sprzętowych pozwalających na uruchomienie oprogramowania sterującego. Ze względu na praktyczne wymagania wspomniane później w rozdziale 3.2, pula platform sprzętowych dodatkowo zmniejsza się do rozwiązań takich jak:

- Komputer przemysłowy,
- Sterownik PLC [12],
- Klasyczny system pomiarowy NI cDAQ + PC [13],
- NI CompactRIO [14].

Przedstawione platformy sprzętowe pozwalają na implementację oprogramowania będącego w stanie zarówno spełniać funkcjonalne wymagania dotyczące samego procesu nagrzewania, jak i udostępnić użytkownikowi odpowiednie narzędzia do intuicyjnej obsługi programu z wykorzystaniem graficznego interfejsu. Projekt jest realizowany na zlecenie wewnętrznego klienta, ponieważ posiada on już doświadczenie związane z rozwiązaniami budowanymi na platformie CompactRIO, wspomniany system zostanie wybrany jako platforma sprzętowa.

2.6. Systemy Real-Time

System czasu rzeczywistego RTOS (ang. *Real Time Operating System*) to system operacyjny, w którym możliwe jest definiowanie zadań, których maksymalny czas wykonania jest ściśle określony. System operacyjny może tak przydzielać zasoby obliczeniowe, aby wybrane zadania kończyły się w określonym czasie gwarantując uzyskanie wyniku w sposób deterministyczny. W konwencjonalnym systemie zasoby przydzielane są pomiędzy pracującymi wątkami w zależności od obciążenia procesora i obciążenia systemu, bez precyzyjnej kontroli nad czasem wykonania zadań. Systemy RTOS niekoniecznie działają szybciej od konwencjonalnych, natomiast często prędkość obliczeń nie jest najważniejsza, a właśnie wspomniany determinizm. [15], [16].

Głównymi cechami RTOS są:

- **Ciągłość działania:**
System działa bez przerwy, oczekuje na zmiany wejść z otoczenia,
- **Współbieżność:**
System składa się z wielu podsystemów, które muszą działać współbieżnie w celu wykonania określonej funkcjonalności
- **Wysoka przewidywalność i niezawodność:**
Sygnały zewnętrzne pojawiają się w sposób niezdeterminowany. System musi tak je przetwarzać i reagować na ich wystąpienie, aby na krytyczne funkcje reagować w sposób przewidywalny, zmniejszając nakłady zasobów na funkcje niedeterministyczne lub o mniejszym priorytecie, w ten sposób gwarantując niezawodność oprogramowania [17].
- **Punktualność:**
Odpowiedź systemu na bodźce zewnętrzne muszą być obliczane zgodnie z założonymi restrykcjami czasowymi, w oparciu o implementowane algorytmy [18]

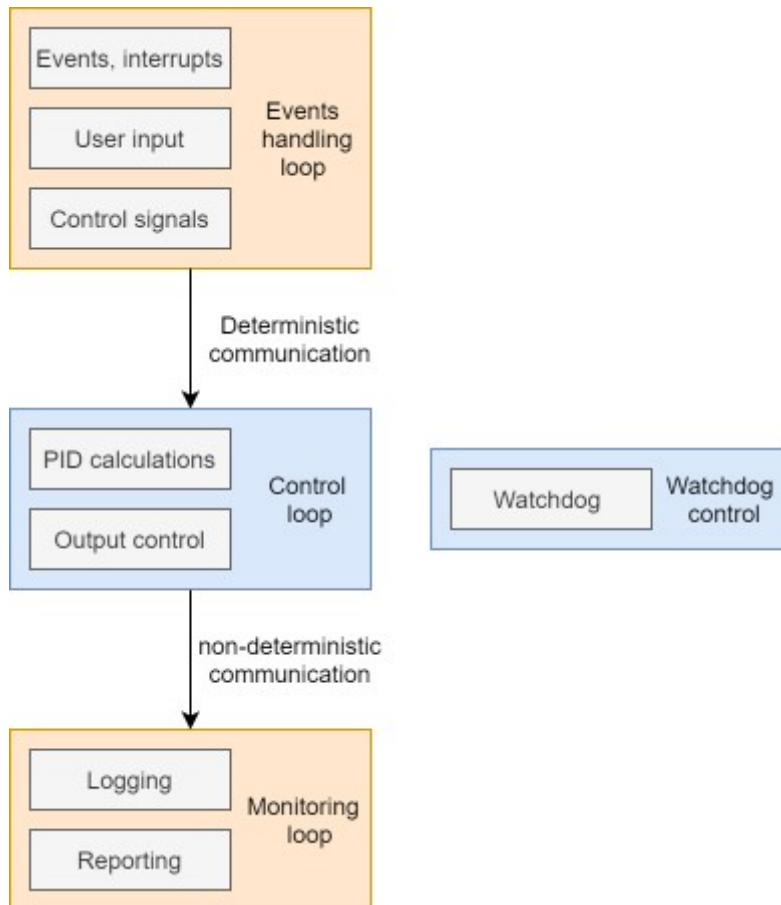
Wyróżniamy dwa typy systemów czasu rzeczywistego w zależności od ograniczeń czasowych:

- **Miękkie systemy czasu rzeczywistego** (ang. *Soft Real-Time Systems*)
W systemach tego typu, oprogramowanie może kontynuować pracę w przypadku przekroczenia ograniczeń czasowych wynikających z założeń. Tego typu rozwiązanie może zostać użyte w przypadku systemów nie będących krytycznymi.
- **Twarde systemy czasu rzeczywistego** (ang. *Hard Real-Time Systems*)
Systemy twarde natomiast posiadają wyższe wymagania oraz precyzyjnie określony maksymalny czas obliczeń funkcjonalności. Systemy tego typu często spotykane są w branży kosmicznej, samochodowej, kontroli produkcji, medycznej czy w robotyce. Tego typu systemy mogą zakończyć funkcjonowanie, jeśli maksymalny czas został przekroczony, ponieważ może to skutkować poważnymi konsekwencjami. Przykładowo, jeśli system czasu rzeczywistego nie będzie w stanie zsynchronizować ruchu ramion robotów w automatycznej linii produkcyjnej, należy taki proces przerwać i poinformować operatora. W tym przypadku niemożliwa jest dalsza praca, ze względu na ryzyko uszkodzeń robotów lub spadek jakości produkowanych wyrobów [18].

2.7. Przykładowa architektura oprogramowania systemu czasu rzeczywistego

Naturalnie, nie wszystkie funkcjonalności realizowane na platformie z systemem czasu rzeczywistego muszą być deterministyczne. Wiele funkcji, takich jak pomiary, przetwarzanie danych czy wyprowadzanie sygnałów są kluczowe dla działania aplikacji. Niemniej często moduł Real-Time obsługuje też funkcjonalności niebędące krytycznymi.

Dla przykładu, może to być raportowanie dla użytkownika, zapis i logowanie danych, reakcja na rekonfigurację systemu czy chociażby inicjalizacja i przygotowanie do obsługi procesu. Przykładowy graf architektury oprogramowania tego typu został przedstawiony poniżej na rysunku 3-14.



Rysunek 2-15 Architektura typowego produkcyjnego systemu Real-Time pełniącego funkcję regulacji. [opracowanie własne].

Determinizm w pętli programowej można uzyskać, gdy czas działania każdego z obliczeń może zostać w pewien sposób przewidziany, zarówno ze względu na zależności w systemie czy sam narzut czasu potrzebny na obliczenia. Przykładami funkcjonalności niepozwalającymi na implementację pętli deterministycznej mogą być:

- Obsługa plików,
- Konieczność synchronizacji przerw,
- Obsługa magistrali systemu operacyjnego,
- Komunikacja z innymi wątkami nie będącymi deterministycznymi,
- Transfer danych o nieokreślonej wielkości,
- Komunikacja z urządzeniami wejścia/wyjścia.

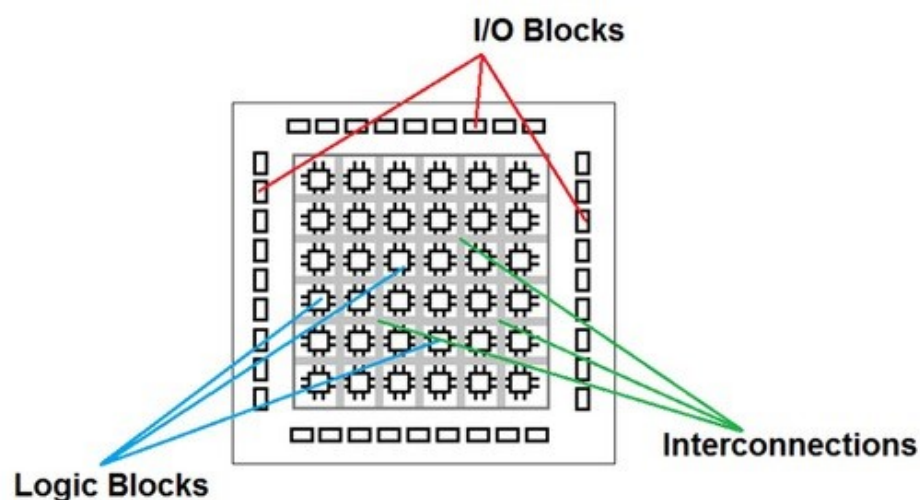
2.8. Systemy FPGA

Kolejnym elementem wykorzystanym do konstrukcji systemu testowego jest bezpośrednio programowalna macierz bramek logicznych (ang. *Field Programmable Gate Array – FPGA*). FPGA to programowalny układ logiczny. Możliwość użycia języka programowania do implementacji układu logicznego znacznie upraszcza i skraca czas budowy specjalizowanego rozwiązania. Bezpośrednio programowalne macierze bramek są zazwyczaj wolniejsze od odpowiadających im funkcjonalnie klasycznych układów logicznych. Łatwość projektowania przekłada się natomiast na znacznie skrócony czas projektowania, zwłaszcza złożonych układów. Z praktycznych względów natomiast często różnica w prędkościach jest w tutaj nieznaczna w kontekście projektowanej aplikacji.

Przede wszystkim, porównując układy FPGA z systemami procesorowymi jest czas wykonywania aplikacji oraz niezawodność. W systemach pracujących w oparciu o procesor, system operacyjny zarządza dostępem do zasobów, w tym czasu dostępu do procesora. System FPGA jest w tym kontekście rozproszony – nie ma tutaj centralnego procesora i systemu operacyjnego a obliczenia wykonywane są niezależnie w wielu miejscach na układzie w zależności od oprogramowania. Problem niezawodności częściowo rozwiązywany jest przez sposób zarządzania zadaniami w systemach czasu rzeczywistego (opartych o procesor), niemniej zwiększenie niezawodności oraz czasu wykonywania logiki w systemie w zasadzie zbudowanego z reprogramowalnego układu logicznego są w tym przypadku niepodważalne.

Każdy układ FPGA jest złożony z ograniczonej ilości programowalnych bloków złożonych z:

- Konfigurowalnych bloków logicznych (ang. *Configurable Logic Blocks - CLBs*),
- Programowalnych obwodów pozwalających na odpowiednie łączenie bloków,
- Bloków wejścia/wyjścia (I/O) – łączących zewnętrzne sygnały podłączone do modułu FPGA.



Rysunek 2-16. Struktura układu FPGA [19].

Naturalnie, przy użyciu układu FPGA należy liczyć się z ograniczeniami sprzętowymi wynikającymi z samej architektury układu. W przypadku procesora, mamy do czynienia ze specjalizowaną jednostką obliczeniową, która komunikuje się również ze specjalizowaną pamięcią różnych typów, dodatkowo obsługując magistralę wejścia/wyjścia. W przypadku użycia układu FPGA, zarówno jednostka obliczeniowa, komórki pamięci jak i interfejsy wejścia/wyjścia muszą zostać oprogramowane z wykorzystaniem uniwersalnych bloków logicznych. Z tego względu, dostępne zasoby sprzętowe w porównaniu z wielkością klasycznych procesorów, czy chociażby ich ceną, są zdecydowanie mniejsze. Układy są natomiast stosowane tam, gdzie klasyczne rozwiązania procesorowe nie spełnią wymagań czy to niezawodności, czy prędkości obliczeń.

W zależności od producenta, typu i oczywiście ceny, ilość bloków zmienia się i jest to głównym czynnikiem determinującym konieczność optymalizacji kodu. Bez specjalnego zagłębiania się w detale, w dalszej części pracy wszystkie bloki, z których zbudowany jest układ będą ogólnie nazywane zasobami sprzętowymi [20].

Aby skutecznie implementować funkcjonalności wykorzystując układ FPGA, należy z rozumą używać zasobów sprzętowych. Sam kod powinien być możliwie niskopoziomowy a funkcje powinny być optymalizowane pod kątem zmniejszenia liczby zmiennych, ich rozmiaru w bitach oraz ilości obliczeń potrzebnych do wykonania zadania.

2.9. Wprowadzenie do LabVIEW

LabVIEW jest graficznym środowiskiem programowania, które jest używane do implementacji oprogramowania inżynierskiego na potrzeby rozwoju, walidacji i testów produkcyjnych systemów przemysłowych. Szeroki wachlarz funkcji, stale rozwijane środowisko, duży zasób sprzętowy współpracujący z LabVIEW oraz integracja urządzeń firm trzecich pozwala na tworzenie profesjonalnych rozwiązań w krótkim czasie [21].

Wybranymi cechami środowiska LabVIEW są między innymi:

- Możliwość szybkiej implementacji wysoce precyzyjnych czasowo pomiarów.
- Duże możliwości przetwarzania sygnałów i post-processingu.
- Integracja z urządzeniami zarówno dostępnymi od producenta, jak firm trzecich.
- Wręcz trywialnie łatwe implementowanie wielowątkowych i rozbudowanych aplikacji.
- Rozbudowane API obsługi danych, oprogramowania firm zewnętrznych oraz protokołów komunikacyjnych.

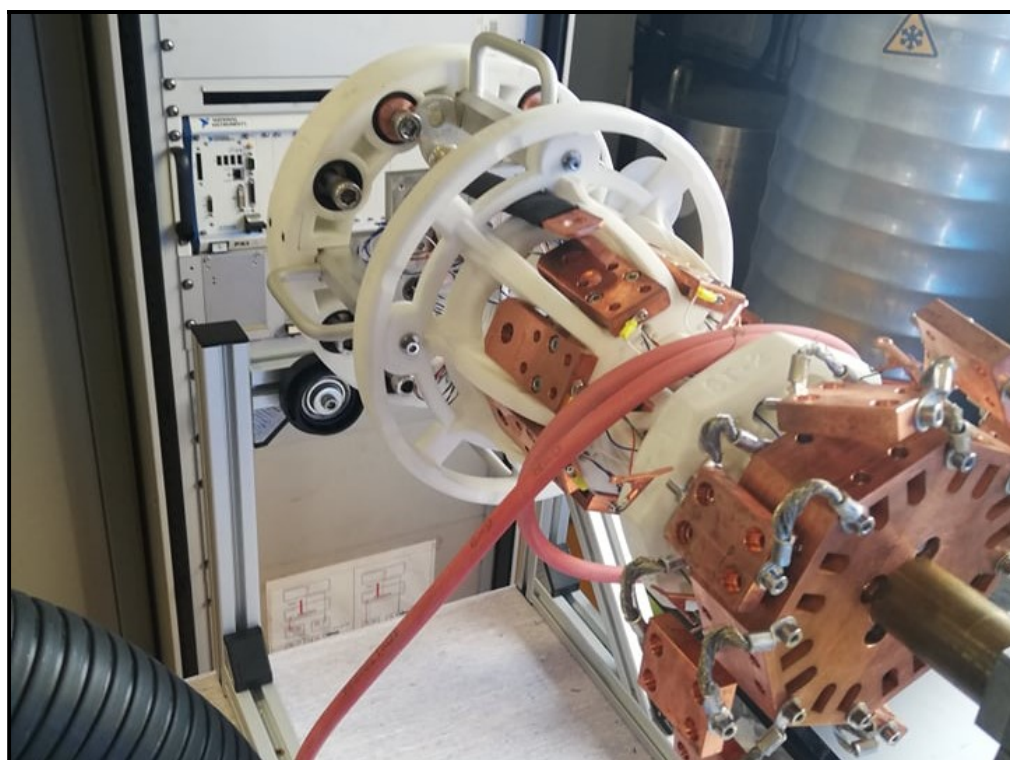
Producent dostarcza szereg dodatkowych funkcjonalności, możliwych do doinstalowania do środowiska. Niektórymi z nich jest właśnie moduł Real-Time oraz moduł FPGA.

3. Cześć praktyczna

3.1. Wprowadzenie do projektu

Ze względu na degradację nadprzewodników wysokiej temperatury pod wpływem ekspozycji na temperaturę, opisaną w rozdziale 3.4, konieczna jest implementacja środowiska testowego. Ma ono na celu nagrzewanie próbki w określonych interwałach czasowych i weryfikację stopnia degradacji materiału pod kątem właściwości nadprzewodnictwa. Wspomniane testy przebiegają następująco:

- Ze szpuli taśmy HTS wycinany jest kawałek do weryfikacji.
- Taśma jest schładzana do temperatury, w której staje się nadprzewodząca. Wówczas, przeprowadzane są testy, mające na celu ustalić prąd krytyczny. Stopniowo, przepływ prądu przez przewodnik jest zwiększany aż do wystąpienia quenchingu. Testy prądu krytycznego są wówczas zapisane jako wartość referencyjna.
- Następnie, wykonywany jest test precyzyjnie symulujący nagrzewanie oraz schładzanie taśmy w określonych interwałach czasowych.
- Po ekspozycji na wysoką temperaturę, taśma znów zostaje schłodzona, aby ponownie przeprowadzić weryfikację prądu krytycznego.
- Mając do dyspozycji te dwa zestawy danych, możliwe jest stwierdzenie czy zjawisko degradacji dla konkretnej partii jest znaczące.



Rysunek 3-1. Prototyp urządzenia służącego do testów prądu krytycznego dla 8 próbek.

3.2. Definicja wymagań

W związku z powyższym, należy zaplanować platformę sprzętową oraz architekturę oprogramowania, pozwalającego na spełnienie wymagań projektowych:

- System powinien niezależnie obsługiwać 8 kanałów testowych, z czego każdy z nich musi być wyposażony w:
 - Kartridż grzewczy, który pozwoli na doprowadzenie czterech czujników temperatury wraz z dwoma grzałkami, w celu równomiernego nagrzewania oraz odpowiedniej kontroli nad temperaturą próbki.
 - 4 czujniki temperatury:
Wykorzystane zostały termopary. Termopara to element, który generuje napięcie, ze względu na zjawisko powstawania siły elektromotorycznej w obwodzie elektrycznym zawierającym różne przewodniki w różnych temperaturach. Wówczas do określenia temperatury można skorzystać z tablic konwersji napięcia na temperaturę a sam termometr potraktować jako element pasywny.
W zależności od typu (materiałów) generowane są różne wartości napięcia (typ K oraz J, które zostały użyte w projekcie).
 - 2 grzałki sterowane sygnałem PWM (ang. *Pulse Width Modulation*) pozwalające na osiągnięcie temperatury do 300 °C.
 - Wiatrak, pozwalający na schłodzenie próbki po zakończeniu testów.
 - Kontrolki wskazujące na status konkretnego kanału.
- Temperatura powinna być mierzona z dokładnością co najmniej 0.1 °C.
- Częstotliwość sterowania wyjściem powinna być nie mniejsza niż 10 Hz.
- W wypadku przekroczenia różnicy temperatur pomiędzy czujnikami termopary w obrębie jednego kartridża, należy przerwać proces oraz zaalarmować operatora.
- Możliwość programowania profili testowych.
- Automatyczne zapisywanie przebiegu testów.
- System powinien działać niezawodnie bez konieczności ingerencji użytkownika.
- Interfejs użytkownika powinien pozwalać na modyfikację parametrów pracy systemu oraz w sposób intuicyjny informować o przebiegu testów dla poszczególnych kanałów.
- Komunikacja interfejsu użytkownika z systemem produkcyjnym powinna odbywać się w obrębie sieci lokalnej.

Dodatkowo, wszystkie z podanych parametrów powinny być możliwe do modyfikacji przez klienta, ze względu na konieczność późniejszego wdrożenia do produkcji.

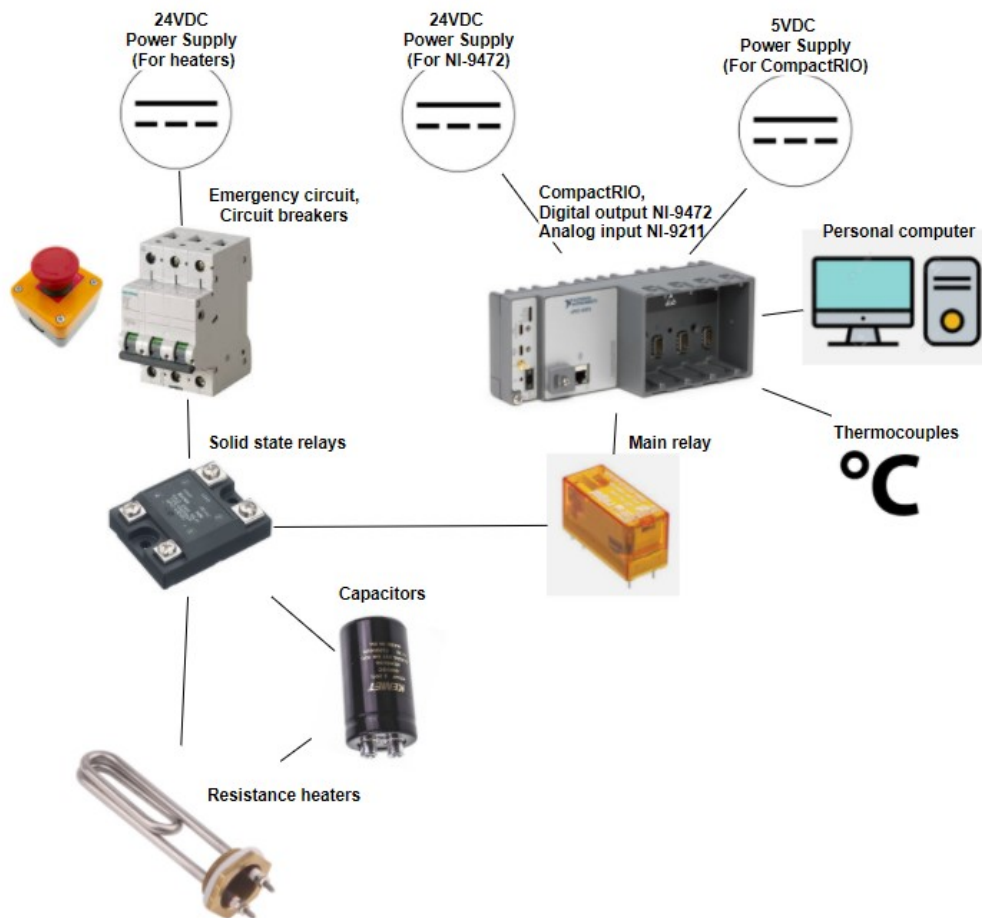
W oparciu o podane parametry zostanie przygotowane stanowisko testowe, które następnie zostanie zastąpione przez rozwiązanie produkcyjne.

3.3. Dobór platformy sprzętowej

Na potrzeby implementacji rozwiązania zostały wykorzystane następujące komponenty:

- NI CompactRIO NI-9053 wraz ze zintegrowanym modułem FPGA.
- Rezystancyjne przewody grzewcze zasilane napięciem 24VDC.
- Przełączniki półprzewodnikowe.
- Termopara typu K oraz J.
- Zasilacze:
 - Zasilacz 5V DC dla CompactRIO.
 - Zasilacz 24 DC dla karty wyjść cyfrowych.
 - Zasilacz 24V DC dla sygnału PWM dla grzałek.
- Kondensatory dla każdej z grzałek – w celu wygładzenia profilu napięcia na wyjściu.
- Obwód bezpieczeństwa.
- Bezpieczniki nadprądowe dla każdego kanału oraz dla całego systemu.

Na rysunku 4-1 został przedstawiony schemat połączenia systemu.



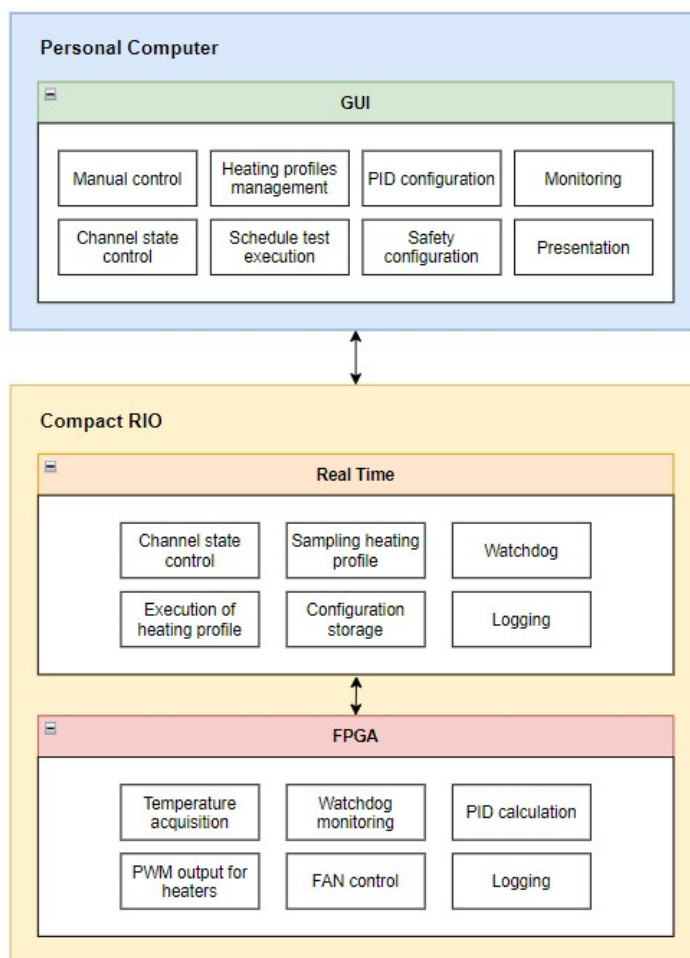
Rysunek 3-2. Schemat systemu [opracowanie własne].

3.4. Architektura, komunikacja

Architektura oprogramowania została zaprojektowana w taki sposób, aby część wykonawcza systemu była zupełnie niezależna od interfejsu użytkownika. Oprogramowanie zostało podzielone na trzy moduły:

- Interfejsu użytkownika (ang. *graphical user interface - GUI*),
- Systemu czasu rzeczywistego,
- Modułu FPGA.

Funkcjonalności poszczególnych modułów zostały przedstawione poniżej na rysunku 5-1.



Rysunek 3-3-3. Architektura oprogramowania [opracowanie własne].

W oprogramowaniu systemu czasu rzeczywistego zaimplementowany jest serwer TCP/IP, do którego może podłączyć się aplikacja kliencka. Cała komunikacja przebiega wówczas przy użyciu gniazda w standardzie TCP. Ze względu na brak twardego reżimu czasowego w protokole TCP/IP, ta część oprogramowania nie jest deterministyczna. Aplikacja użytkownika funkcjonalnie nie jest krytycznym elementem systemu a jej głównym zadaniem jest monitorowanie i planowanie testów. Na rysunku 5-2 przedstawiony jest schemat połączeń pomiędzy modułami. Aplikacja kliencka implementuje dwa sposoby komunikacji:

- Obsługa kolejek FIFO (ang. *First In First Out*). Ten sposób komunikacji jest wykorzystany jako metoda zarządzania systemem czasu rzeczywistego w postaci komend. Komendy wysyłane z poziomu aplikacji klienckiej (start testu, aktualizacja konfiguracji, włączenie wiatraka etc.) mogą być kolejgowane i przetwarzane przez system czasu rzeczywistego zgodnie z kolejnością ich otrzymania, bez utraty nieprzetworzonych jeszcze zleceń.
- Transfer pojedynczych danych, zebranych ogólnie jako strumień TCP. Aplikacja posiada kilka monitorowanych parametrów, takich jak wykorzystanie procesora, zużycie pamięci czy aktualny stan systemu czasu rzeczywistego. Ponieważ wspomniane dane określają chwilowy stan systemu i są wykorzystywane do monitorowania, mogą być obsługiwane przez strumień TCP a z poziomu GUI traktowane są jako cyklicznie aktualizowane, współdzielone zmienne globalne.

Komunikacja pomiędzy systemem czasu rzeczywistego a FPGA zostanie omówiona w dalszej części pracy.

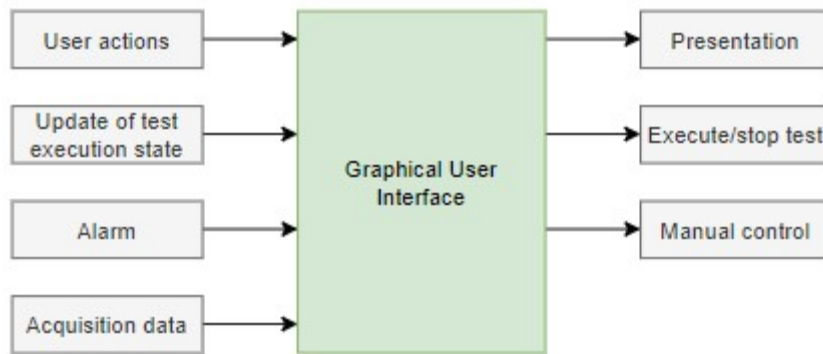


Rysunek 3-4. Komunikacja w systemie [opracowanie własne].

3.5. Aplikacja użytkownika

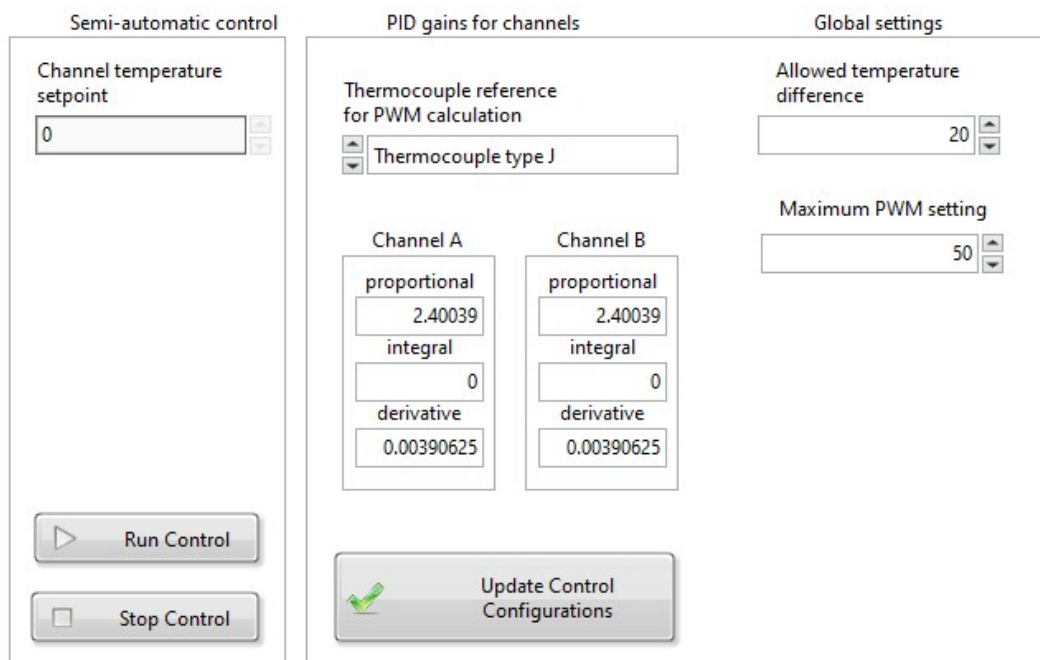
3.5.1. Funkcjonalności

Moduł aplikacji użytkownika należy rozumieć jako klienta, który może łączyć się z systemem czasu rzeczywistego. Głównie jest przeznaczony do monitorowania i planowania testów, natomiast dalsze części systemu (RealTime + FPGA) dbają o pomyślne wykonanie testów. Aplikacja pozwala dodatkowo na rekonfigurację systemu – możliwość zmiany nastaw PID regulatora zaimplementowanego na FPGA, ustawień PWM, temperatury alarmu, manualnej kontroli zadanej temperatury i innych. Na rysunku 5-3 przedstawiono ideowy schemat funkcjonalności modułu GUI.

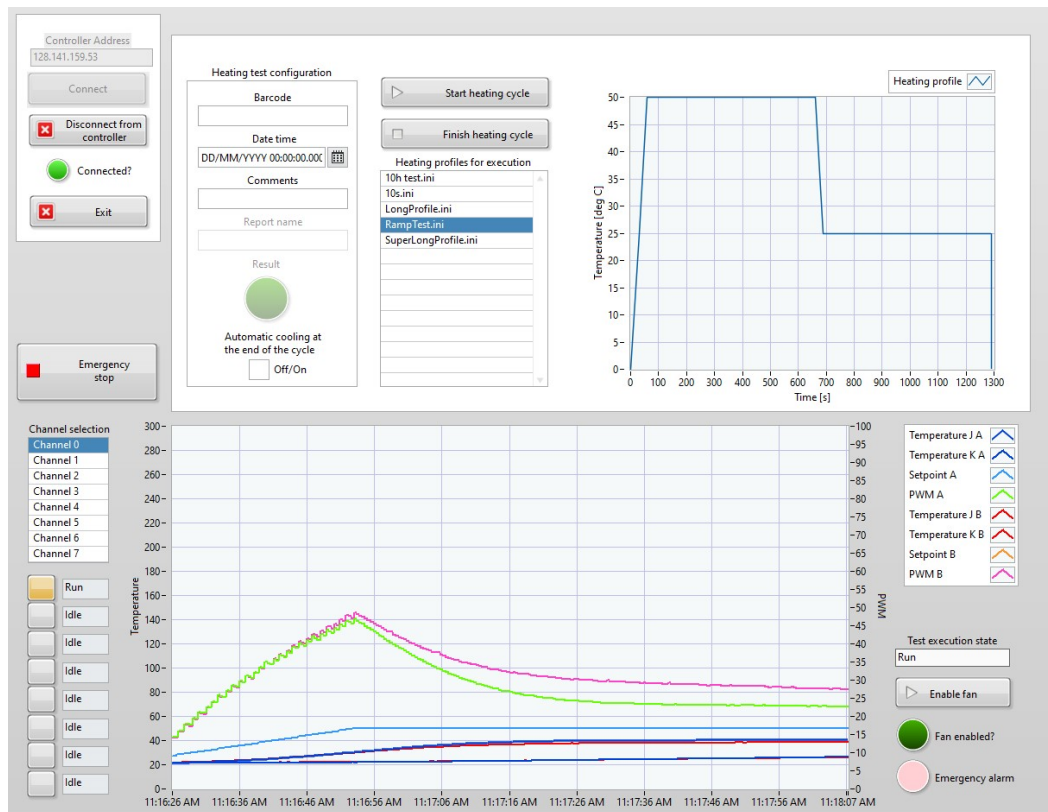


Rysunek 3-5. Funkcjonalności GUI [opracowanie własne].

Aplikacja działa niezależnie od reszty systemu – oznacza to, że obsługuje przypadek rozłączenia od serwera czy restartu. Wszystkie parametry konfiguracyjne przechowywane są lokalnie w module Real-Time, natomiast aplikacja klienta wysyła prośbę do modułu o przesłanie zarówno parametrów konfiguracyjnych, jak i kompletnych informacji o statusie systemu, aby zaktualizować konfigurację lokalną. Rysunki 5-4 oraz 5-5 przedstawiają niektóre z elementów interfejsu użytkownika.



Rysunek 3-6. Panel konfiguracyjny [opracowanie własne].



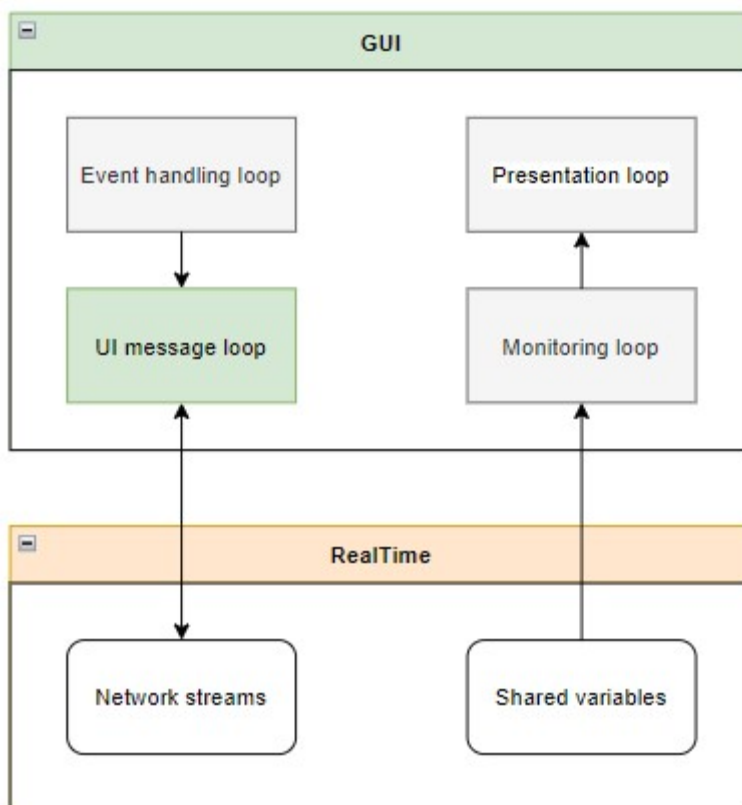
Rysunek 3-7. Interfejs monitorowania kanałów [opracowanie własne].

3.5.2. Struktura kodu

Strukturalnie, kod podzielony jest pomiędzy czterema wątkami, które odpowiadają za:

- Rejestrację zdarzeń a panelu użytkownika. Ten wątek obsługuje wszystkie zdarzenia związane z interfejsem użytkownika, takie jak klikanie przycisków, wybieranie kart czy tabel etc. Na podstawie tych działań tworzona jest właściwa wiadomość i wysyłana do niezależnego wątku w celu dalszego przetwarzania.
- Reakcji na wspomniane zdarzenia oraz przetwarzaniu komend zarówno z innych wątków jak i od modułu czasu rzeczywistego, wraz z implementacją odpowiedniej logiki.
- Monitorowania i buforowania danych przybywających od modułu Real-Time. Dane są przetwarzane i zapisywane w kolejkach stratnych niezależnych dla każdego kanału. Pozwala na przechowywanie danych z pewną historią.
- Prezentacja danych (nastawy temperatur, PWM, aktualne temperatury) na wykresie w zależności od wybranego kanału.

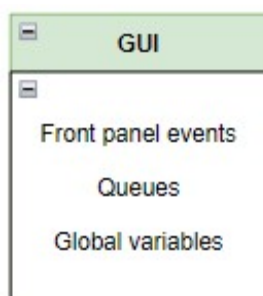
Wspomniane wątki zostały przedstawione poniżej na rysunku 5-6.



Rysunek 3-8 Wątki oraz ich funkcje [opracowanie własne].

3.5.3. Komunikacja pomiędzy wątkami

Ze względu na fakt, iż aplikacja nie jest krytyczną częścią systemu, komunikacja pomiędzy wątkami odbywać się może z wykorzystaniem klasycznych metod dostępnych w środowisku takich jak eventy (zdarzenia), kolejki FIFO (ang. *First In First Out*), czy współdzielone zmienne globalne (rysunek 5-7).



Rysunek 3-9. Komunikacja w obrębie modułu GUI [opracowanie własne].

3.6. Moduł czasu rzeczywistego

3.6.1. Funkcjonalności

Moduł czasu rzeczywistego jest modułem pośrednim, które reaguje na polecenia interfejsu użytkownika w celu przygotowania wątków regulujących pracę modułu FPGA na potrzeby kontroli testów ogrzewania. Zarówno Real-Time jak i FPGA powinny pracować nieprzerwanie i niezawodnie. Z tego względu w module Real-Time implementowany jest licznik watchdog. Mechanizm watchdog to popularne rozwiązanie mające na celu zabezpieczyć aplikację przed nieoczekiwanym wyłączeniem.

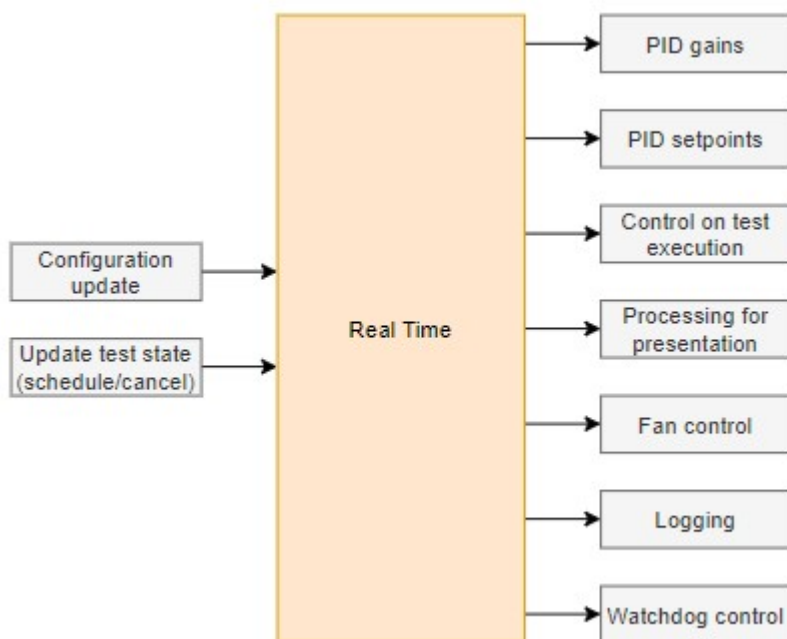
Mechanizm polega na monitorowaniu jednego modułu przez drugi. W omawianym przypadku, moduł FPGA kontroluje stan systemu czasu rzeczywistego. Jeśli ten drugi ulegnie awarii, wówczas należy przerwać testy temperaturowe i poinformować operatora o usterce. Implementacja polega na zdefiniowaniu zmiennej, która będzie dostępna dla dwóch modułów. Wówczas, jeden z modułów (Real-Time) cyklicznie ustawia zmienną w stan wysoki, natomiast drugi (FPGA) sprawdza, czy wspomniana zmienna jest w stanie wysokim a następnie ustawia ją w stan niski. Jeśli FPGA odczyta zmienną w stanie niskim, oznacza to, że nie została ona ustawiona w stan wysoki przez moduł będący monitorowanym, który uległ awarii.

Dodatkowo, moduł czasu rzeczywistego implementuje wątki deterministyczne. Osiągnięcie przewidywalności jest możliwe dzięki zastosowaniu predefiniowanych rozmiarów zmiennych wejścia/wyjścia do pętli, brak zewnętrznych zależności czy synchronizacji oraz zastosowaniu przewidywalnej komunikacji.

Podstawowymi funkcjami, za które odpowiedzialny jest moduł są:

- Inicjalizacja serwera i oczekiwanie na aplikację kliencką.
- Reakcja na komendy wysyłane zarówno z modułu GUI jak i z wewnętrznej komunikacji.
- Pobranie od GUI profilu grzewczego, sekwencjonowanie go i następnie cykliczne dostarczanie nowych nastaw temperaturowych dla FPGA podczas samej egzekucji testu.
- Akwizycja danych, logowanie, przetworzenie i wysyłanie do modułu GUI na potrzeby prezentacji.
- Obsługa systemu watchdog.
- Monitorowanie zużycia zasobów sprzętowych.

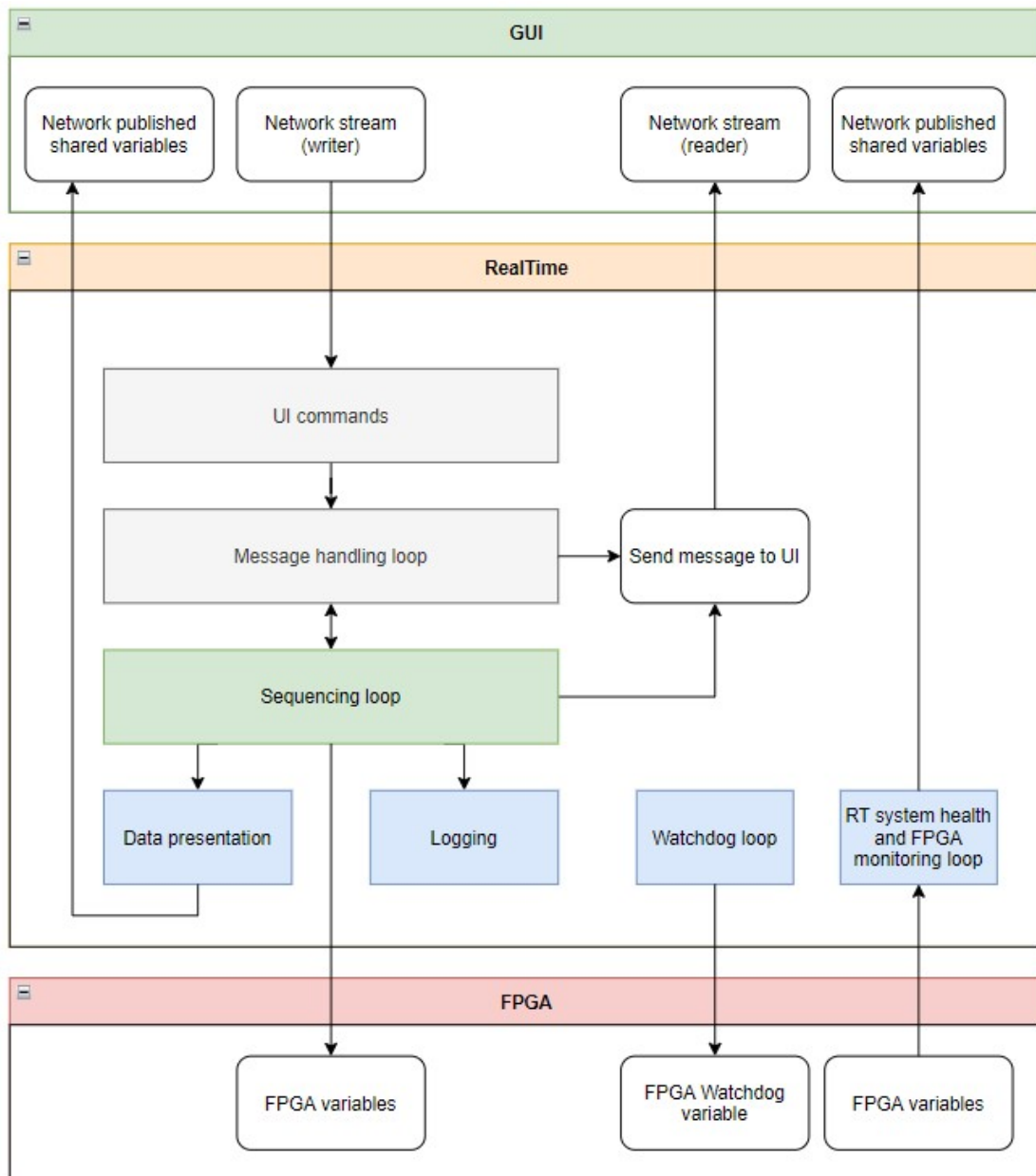
Rysunek 5-8 przedstawia ideowy schemat funkcjonalności.



Rysunek 3-10. Funkcjonalności systemu czasu rzeczywistego [opracowanie własne].

3.6.2. Struktura kodu

System Real-Time jest w tym przypadku najbardziej rozbudowaną częścią kodu. Powyższa lista funkcjonalności oraz rysunek 5-9 przedstawiający odpowiedzialne za nie wątki wystarczająco oddaje strukturę kodu.



Rysunek 3-11. Wątki i ich funkcje [opracowanie własne].

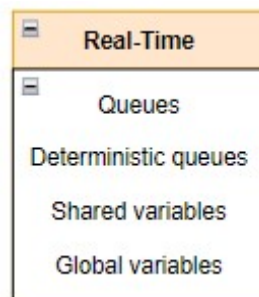
Niemniej, w tym miejscu warto zaznaczyć, że wątek sekwencjonujący jest deterministyczny, o największym priorytecie (oznaczony jako zielony kafelek). Podwyższony priorytet, jednak bez gwarantowanego determinizmu posiadają wątki oznaczone kolorem błękitnym.

3.6.3. Komunikacja pomiędzy wątkami

Sposoby komunikacji pomiędzy wątkami zostały przedstawione na rysunku 5-10.

Wykorzystywane są:

- Kolejki FIFO (zarówno deterministyczne jak i nie) – na potrzeby komunikacji pomiędzy wątkami,
- Zmienne globalne – na potrzeby komunikacji pomiędzy wątkami,
- Zmienne współdzielone poprzez protokół TCP (można je przybliżyć do kolejki FIFO o maksymalnie jednym elemencie – są one implementowane na potrzeby komunikacji z modułem GUI).

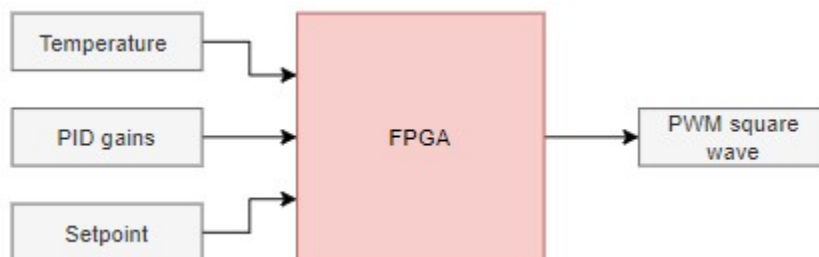


Rysunek 3-12. Komunikacja w obrębie modułu Real-Time [opracowanie własne].

3.7. FPGA

3.7.1. Funkcjonalności

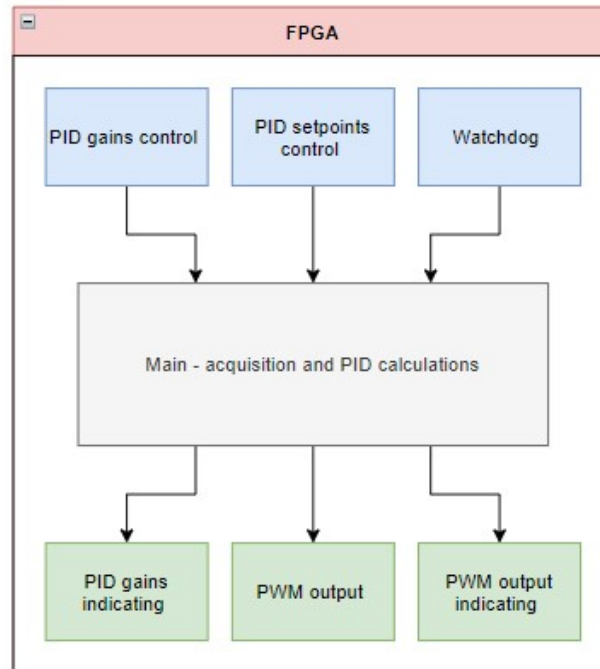
Ponieważ moduł FPGA jest najbardziej niezawodną częścią kodu, przejmuje on kontrolę nad kluczowym procesem - ogrzewaniem. Domyślnie kanały są wyłączone, a polecenie z RealTime może zmienić stan bezczynności kanału na aktywny (grzanie). Moduł FPGA może pracować w dwóch trybach – *safe state* oraz *heating state*. Ponadto jest bezstanowy (nie posiada informacji o stanach konkretnych kanałów czy statusie wykonywanego testu). W zależności od dostarczonego przez Real-Time punktu temperatury do osiągnięcia, oblicza i wyprowadza na wyjściu sygnał PWM, sterowany regulatorem PID (rysunek 5-11).



Rysunek 3-13. Funkcjonalności modułu FPGA [opracowanie własne].

3.7.2. Struktura programu

Biorąc pod uwagę fakt, że w układach FPGA można dowolnie często implementować kod wykonywany równoległe, w oprogramowaniu zaimplementowanych zostało 7 pętli. Schemat struktury kodu przedstawia rysunek 5-12.



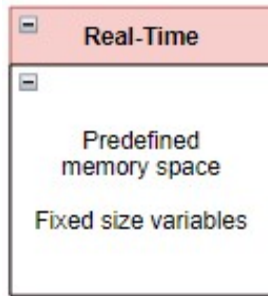
Rysunek 3-14. Podmoduły i ich funkcje [opracowanie własne].

Przedstawione pętle odpowiadają za:

- PID gains control – pobieranie od systemu czasu rzeczywistego komendy zmieniającej nastawy regulatora PID.
- PID gains indicating – publikowanie danych dla system czasu rzeczywistego.
- PID setpoints control – modyfikacja aktualnego zadanego punktu temperatury.
- Watchdog – analiza stanu mechanizmu watchdog.
- PWM output – przetwarzanie procentowej wartości wyjścia regulatora PID na sygnał prostokątny wyprowadzany na wyjście cyfrowe.
- PWM output indicating – publikowanie aktualnych wartości PWM dla Real-Time
- Main – pomiar napięcia z czujników termopary, konwersja do temperatury oraz regulacja PID w zależności od nastaw PID i aktualnie zadanego punktu temperatury.

3.7.3. Komunikacja pomiędzy pętlami

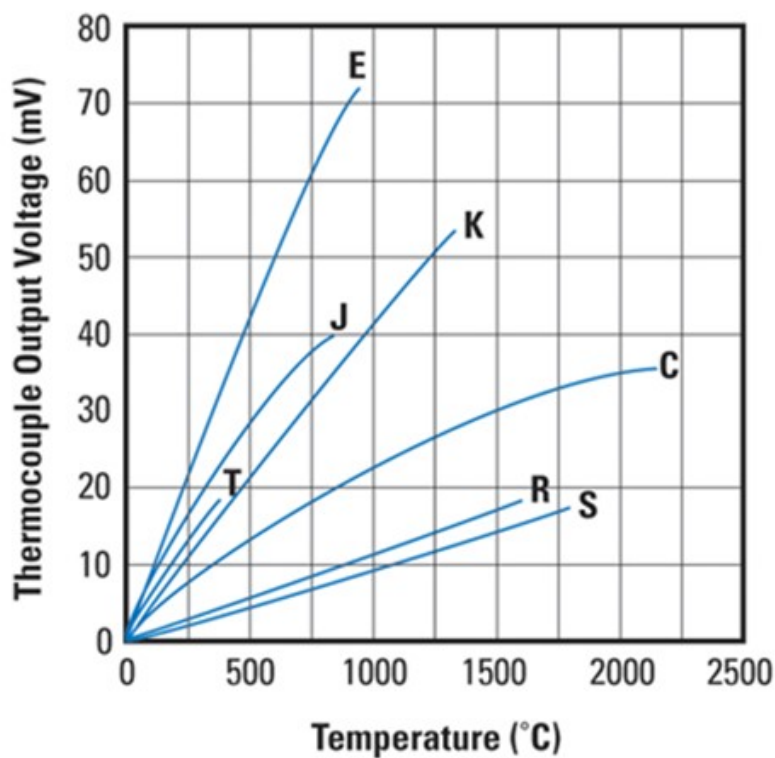
Ze względu na specyfikę FPGA, komunikacja może odbywać się poprzez dostęp do predefiniowanej pamięci lub sygnały zmiennych mogą być fizycznie doprowadzone do poszczególnych pętli (rysunek 5-13).



Rysunek 3-15. Komunikacja w obrębie modułu FPGA [opracowanie własne]

3.7.4. Problem wyznaczenia rozwiązania nieliniowych zależności napięcie-temperatura dla termopary

Czujniki termopary generują napięcie ze względu na występującą różnicę temperatur pomiędzy złączami. Zależność napięcia od temperatury jest nieliniowa i jej przebieg wygląda różnie w zależności od rodzaju termopary. W przypadku omawianego projektu używane są czujniki typu K oraz J. Zależność napięcia od różnicy temperatur została przedstawiona na rysunku 5-14.

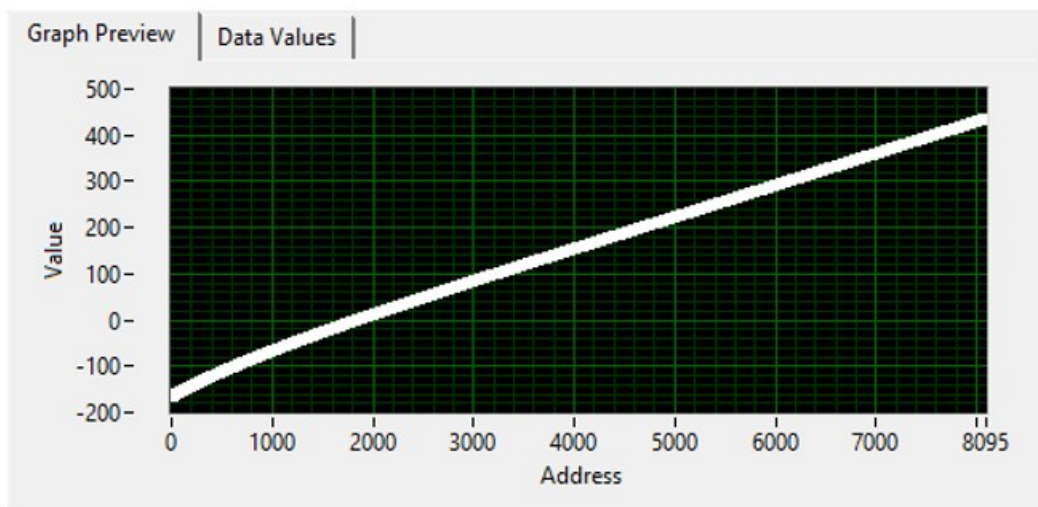
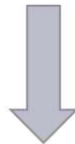
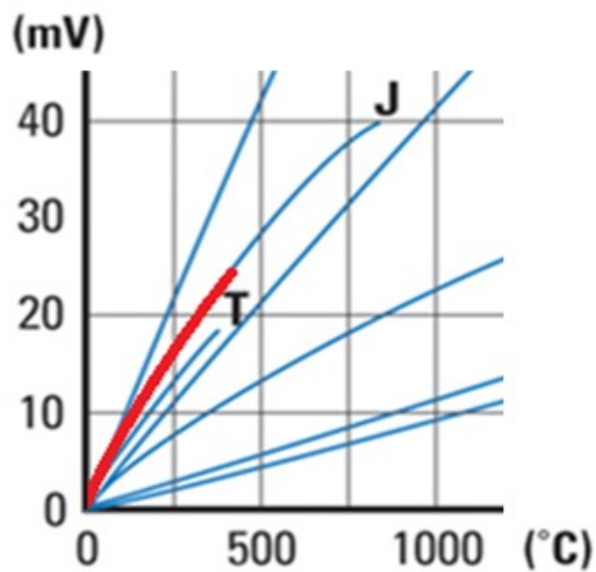


Rysunek 3-16. Zależność generowanego napięcia od różnicy temperatur [22]

Mierzone jest oczywiście napięcie, natomiast do przeliczenia na temperaturę konieczna jest wielomianowa interpolacja. Im większą rozdzielczość konwersji chcemy uzyskać, tym, formuła staje się bardziej rozbudowana. Układy FPGA nie są przystosowane do obliczeń równań wielomianowych. Konsumują one znaczną ilość zasobów sprzętowych i implementacja konwersji w sposób klasyczny staje się niemożliwa.

Propozycja rozwiązania

Aby poradzić sobie ze wspomnianym problemem, rozwiązaniem może być wygenerowanie predefiniowanej tablicy (ang. lookup-table). Tablica działa jak słownik – wypełniona jest danymi w parach – klucz oraz wartość. Wówczas można taką tablicę wypełnić wartościami napięcia o wystarczająco wysokiej rozdzielczości a następnie przeliczyć temperatury odpowiadające napięciu. W ten sposób można zrezygnować z wykonywania obliczeń podczas pracy systemu, a jedynie odczytywać wartość temperatury odpowiadającej najbliższej wartości napięcia zapisanej w tablicy.



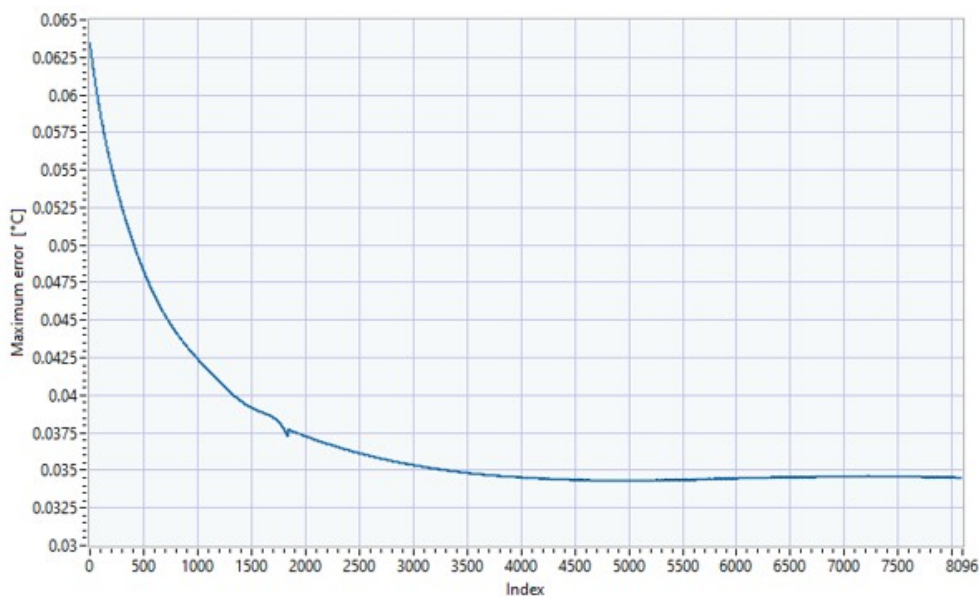
Rysunek 3-17. Obszar pamięci wypełniony przeliczonymi wartościami [opracowanie własne].

Tablica oczywiście wykorzystuje wolne zasoby modułu FPGA, zatem należy z rozwagą określić zarówno jej wielkość jak i rozdzielczość

W omawianym przypadku, zakres temperatur nie przekroczy wartości 0-300 stopni C, zatem tablica została wygenerowana dla wartości temperatury z przedziału z zachowaniem odpowiedniej tolerancji.

Testy oraz analiza wyników

Na rysunku powyżej (5-15) przedstawiony jest zakres napięcia odpowiadający różnicy temperatury na termoparze typu K. Zakres napięć -7:24 [mV] odpowiada temperaturze z przedziału około -160:430 [°C], a sama tablica ma wówczas rozmiar 8096 pozycji. Przy takim doborze parametrów tablicy, błąd konwersji względem formuły przedstawia się następująco:



Rysunek 3-18. Wykres zależności błędu od indeksu w tabeli przeglądowej

Rozdzielczość tablicy: 0.0038147 mV

- Maksymalny błąd: 0.06 °C
- Minimalny błąd: 0.03 °C
- Błąd średni: 0.04 °C

Zalety:

- Oszczędzanie zasobów układu FPGA.

Wady:

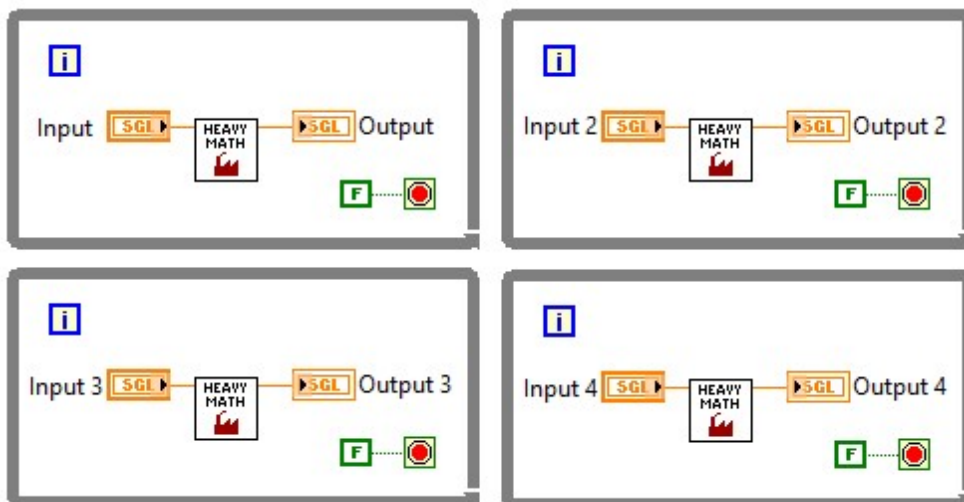
- Błąd konwersji.
- Użycie bloków pamięci.
- Rozwiązanie nie jest elastyczne – tablica musi zostać wygenerowana dla każdego typu termopary, jak również zakresu, dla którego powinna zostać przystosowana.

3.7.5. Problem skalowalności oprogramowania

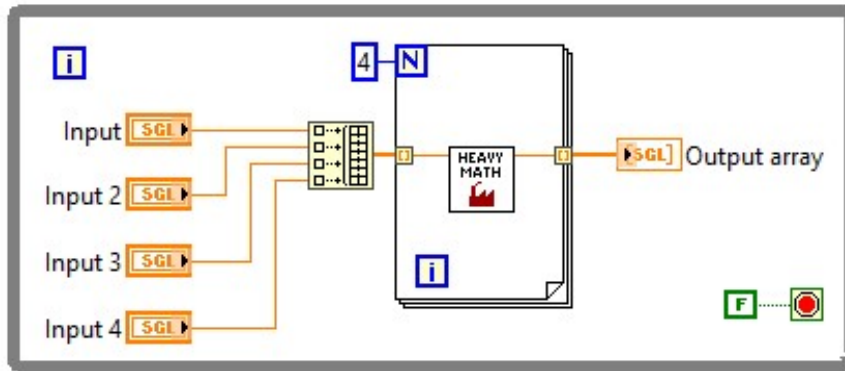
W rozdziale 4.2 zostało sformułowane wymaganie dotyczące skalowalności rozwiązania do 8 niezależnie pracujących kanałów. Ze względu na konieczność implementacji oprogramowania dla jednego kanału na stanowisku testowym, oprogramowanie należało wyskalować na późniejszym etapie implementacji. Najprostszym rozwiązaniem byłoby po prostu powielenie pętli sterującej ośmiokrotnie. Sama pętla sterująca natomiast wykorzystuje funkcję regulatora PID (regulator proporcjonalno-całkująco-różniczkujący), obliczając wyjściowy sygnał w zakresie 0:100. Sama funkcja PID konsumuje natomiast dosyć dużo zasobów sprzętowych. Sygnał następnie jest używany do generowania sygnału PWM. Po powieleniu pętli ośmiokrotnie funkcja obliczająca PID musi być fizycznie umieszczona na układzie 8 razy. Wówczas, wykorzystywane zasoby układu znacznie przekraczają limity, co uniemożliwia kompilację.

Propozycja rozwiązania

Sposobem podejścia do problemu może być użycie jednej pętli, która będzie iterować dane wejściowe. Wówczas funkcjonalnie realizowany jest ten sam cel, niemniej fizycznie w układzie funkcja odpowiedzialna za obliczanie sygnału wyjściowego regulatora PID jest umieszczona wyłącznie jeden raz. Naturalnie, narzut na komunikację jest większy, ponieważ do funkcji trzeba wówczas doprowadzić więcej wejść i wyjść. Równocześnie, czas wykonania obliczeń wzrasta przynajmniej ośmiokrotnie. Należy wówczas sprawdzić, czy nie są przekraczane limity zdefiniowane w wymaganiach. Poniżej został przedstawiony kod przedstawiający poglądowo wspomniane rozwiązania.



Rysunek 3-19. Pętla sterująca powielona czterokrotnie [opracowanie własne].



Rysunek 3-20. Pętla iterująca przez dane wejściowe [opracowanie własne].

Testy oraz analiza wyników

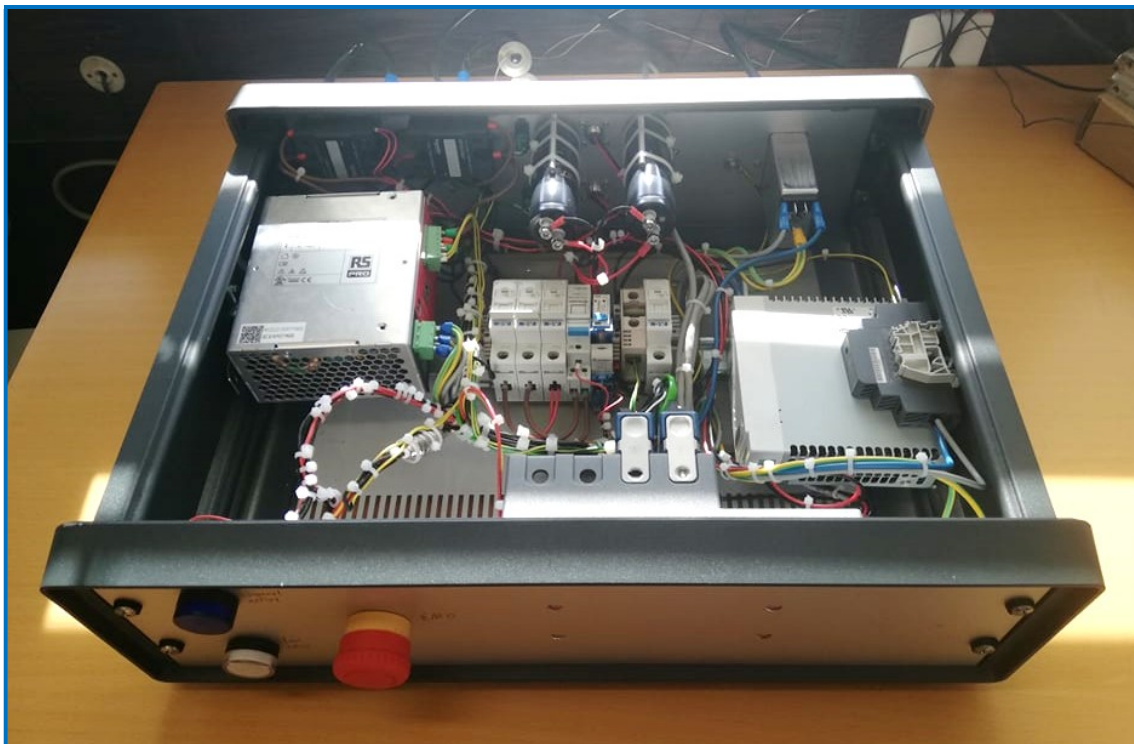
W tabeli umieszczonej poniżej zostały zamieszczone wyniki kompilacji dla dwóch przykładów kodu przedstawionego powyżej. Jednoznacznie widać tutaj skuteczność proponowanego rozwiązania przy zaledwie 4 kanałach.

Tabela 1. Zużycie zasobów sprzętowych dla proponowanych rozwiązań [opracowanie własne].

FPGA resources usage	4x Independent loop	Multiplexing loop	Reduction of resources usage
Total Slices [%]	90.5	43.2	0.48
Slice Registers [%]	25.7	12.9	0.50
Slice LUTs [%]	73.3	29.7	0.41
Block RAMs [%]	18.7	8.0	0.43
DSP48s [%]	16.7	4.2	0.25

4. Walidacja, testy oraz wdrożenie do produkcji

Na potrzeby implementacji została zaprojektowana platforma testowa, funkcjonalnie zbliżona do rozwiązania wykorzystującego jeden kanał (Rysunek 6-1).



Rysunek 4-1. Stanowisko testowe

Planowana integracja w produkcji miała mieć miejsce na początku bieżącego roku, natomiast opóźnienia na rynku związane z dostawami sprzętu znacznie opóźniły proces. Aktualnie ze względu na długi czas oczekiwania na części, stanowisko testowe zostało przystosowane do pracy dla jednego kanału w warunkach produkcyjnych. Na potrzeby wdrożenia z powodzeniem zostały wykonane testy sprawdzające zarówno stabilność jak i funkcjonalność pracy systemu. W momencie redagowania pracy, komora grzewcza oraz część mechaniczna systemu są przygotowywane a system zostanie uruchomiony po zakończeniu bieżących prac.

5. Podsumowanie

Przedstawiona praca prezentuje system wykonany na potrzeby weryfikacji nadprzewodników wysokiej temperatury. System spełnił oczekiwania klienta oraz funkcjonalne wymagania, przy zachowaniu rozsądnego czasu implementacji. Pomimo złożoności systemu pod kątem oprogramowania poszczególnych modułów jak i ich integracji, podczas wdrażania do produkcji nie zostały wykryte poważne błędy. Wyniki testów funkcjonalnych oraz stabilności wskazują na poprawne działanie oprogramowania.

Pomimo wykonanych prac, założenia nie zostały w pełni osiągnięte. Rozwiązanie dla ośmiu kanałów testowych nie jest jeszcze ukończone. Jest to spowodowane znacznymi utrudnieniami na rynku ze względu na brak dostępności komponentów. Producent National Instruments musiał przerwać produkcję urządzeń CompactRIO ze względu na brak specyficznych elementów elektronicznych. Aktualnie, producent jest w trakcie projektowania nowej wersji CompactRIO, aby wykluczyć brakujące komponenty. System został dostosowany do użycia jednego kanału testowego i pomimo wspomnianych trudności, może być z powodzeniem integrowany w środowisku produkcyjnym.

6. Bibliografia

- [1] CERN, „LHC facts,” [Online].
URL: <https://www.lhc-facts.ch/index.php?page=dipol>.
[Data uzyskania dostępu: 10 05 2022].
- [2] „Pulling together: Superconducting electromagnets,” CERN, [Online].
URL: <https://home.cern/science/engineering/pulling-together-superconducting-electromagnets>.
[Data uzyskania dostępu: 10 05 2022].
- [3] „HTS at CERN & LHC Current Leads,” CERN, [Online].
URL: <https://at-mel-cf.web.cern.ch>.
[Data uzyskania dostępu: 10 05 2022].
- [4] CERN, „The ATLAS Experiment,” CERN, [Online].
URL: <https://atlas.cern/de/node/38>.
[Data uzyskania dostępu: 09 06 2022].
- [5] M. Automation, „PLC: Industrial Applications of Programmable Logic Controller,” [Online].
URL: <https://www.mobileautomation.com.au/plc-industrial-application/>.
[Data uzyskania dostępu: 10 06 2022].
- [6] N. Instruments, „CompactDAQ Systems,” [Online].
URL: <https://www.ni.com/pl-pl/shop/compactdaq.html>. [Data uzyskania dostępu: 10 06 2022].
- [7] N. Instruments, „Using CompactRIO with the NI-DAQmx API,” [Online].
URL: <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q000000x2CICAI&l=pl-PL>.
[Data uzyskania dostępu: 10 06 2022].
- [8] jf-parede.pt, „Co to jest system operacyjny czasu rzeczywistego (RTOS) i jak działa?,” [Online].
URL: <https://pl.jf-parede.pt/what-is-real-time-operating-system>.
[Data uzyskania dostępu: 21 04 2022].
- [9] Wikipedia, „Real-time operating system,” [Online].
URL: https://en.wikipedia.org/wiki/Real-time_operating_system.
[Data uzyskania dostępu: 14 02 2022].
- [10] Intel, „Real-Time Systems Overview,” [Online].
URL: <https://www.intel.com/content/www/us/en/robotics/real-time-systems.html>.
[Data uzyskania dostępu: 28 04 2022].
- [11] P. G. dr inż Tomasz Rutkowski, „Systemy Czasu Rzeczywistego (SCR), wykład2: Historia, podstawowe pojęcia i definicje,” [Online].
URL: https://eia.pg.edu.pl/documents/1111711/53561182/W02_SCR_historia_podstawowe_pojecia_definicje.pdf.
[Data uzyskania dostępu: 18 04 2022].

- [12] Stemmer Imaging, „Introduction to FPGA acceleration,” [Online].
URL:<https://www.stemmer-imaging.com/en/technical-tips/introduction-to-fpga-acceleration/>.
[Data uzyskania dostępu: 03 05 2022].
- [13] Wikipedia, „Bezpośrednio programowalna macierz bramek,” [Online].
URL:https://pl.wikipedia.org/wiki/Bezpo%C5%9Brednio_programowalna_macierz_bramek.
[Data uzyskania dostępu: 13 04 2022].
- [14] C. Barth, „Electro-mechanical properties of REBCO coated conductors from various industrial manufacturers at 77 K, self-field and 4.2 K, 19 T,” CERN, [Online].
URL:https://www.researchgate.net/publication/272358146_Electro-mechanical_properties_of_REBCO_coated_conductors_from_various_industrial_manufacturers_at_77_K_self-field_and_42_K_19_T.
[Data uzyskania dostępu: 07 03 2022].
- [15] L. Tomkow, C. Barth, P. Koziol, S. Hopkins, J. Fleiter, A. Bellarino, Practical assessment of commercial ReBCO tapes degradation due to thermal processing, Geneva: CERN, 2022.
- [16] CERN, „Superconductivity,” [Online].
URL: <https://home.cern/science/engineering/superconductivity>.
[Data uzyskania dostępu: 10 05 2022].
- [17] „Cryogenics: Low temperatures, high performance,” CERN, [Online].
URL:<https://home.cern/science/engineering/cryogenics-low-temperatures-high-performance>.
[Data uzyskania dostępu: 10 05 2022].
- [18] A. Ballarino, „Superconducting Links,” Tomy %1 z %24th Joint LHC-LARP Annual Meeting.
- [19] National instruments, „What is LabVIEW? Graphical Programming for test, measurement and control,” [Online].
URL:<https://www.ni.com/pl-pl/shop/labview.html>.
[Data uzyskania dostępu: 19 04 2022].
- [20] electronics.stackexchange.com, „Electrical Engineering,” [Online].
URL:<https://electronics.stackexchange.com/questions/34076/why-do-temperature-voltage-curves-of-all-thermocouple-types-pass-through-the-ori>.
[Data uzyskania dostępu: 07 03 2022].
- [21] „Future Circular Collider,” CERN, [Online].
URL:<https://home.cern/science/accelerators/future-circular-collider>.
[Data uzyskania dostępu: 10 05 2022].
- [22] Wikipedia, „Efekt Meissnera,” [Online].
URL: https://pl.wikipedia.org/wiki/Efekt_Meissnera.
[Data uzyskania dostępu: 10 05 2022].

7. Spis rysunków, tabel, wykresów

Rysunek 2-1. Elektromagnes dwubiegunowy w tunelu. [2]	7
Rysunek 2-2. Schematyczne przedstawienie magnesu dipolowego LHC [2].....	7
Rysunek 2-3. Linie pola magnetycznego dipola wokół dwóch rur wiązki [2].....	8
Rysunek 2-4. Rozkład linii pola w przekroju [2].....	8
Rysunek 2-5. Rozmieszczenie detektorów na obwodzie zderzacza.[3].....	9
Rysunek 2-6. Położenie zderzacza [3].	9
Rysunek 2-7. Rozkład pola magnetycznego w nadprzewodniku [4].	10
Rysunek 2-8. Zablockowany magnes ze względu na efekt Meissnera [4].	10
Rysunek 2-9. Rozkład przyłączy wysokiej mocy. [6].....	11
Rysunek 2-10 Mapa planowanego położenia Future Circular Collider [8]	12
Rysunek 2-11 Struktura taśmy HTS [10].....	13
Rysunek 2-12. Przykładowy profil testów temperaturowych [11].....	14
Rysunek 2-13. Przykładowy profil testów temperaturowych [11].....	14
Rysunek 2-14. Degradacja temperaturowa taśm HTS [11].....	15
Rysunek 2-15 Architektura typowego produkcyjnego systemu Real-Time pełniącego funkcję regulacji. [opracowanie własne].....	18
Rysunek 2-16. Struktura układu FPGA [19].....	19
Rysunek 3-1. Prototyp urządzenia służącego do testów prądu krytycznego dla 8 próbek.....	21
Rysunek 3-2. Schemat systemu [opracowanie własne].	23
Rysunek 3-3-3. Architektura oprogramowania [opracowanie własne].....	24
Rysunek 3-4. Komunikacja w systemie [opracowanie własne].....	25
Rysunek 3-5. Funkcjonalności GUI [opracowanie własne].....	26
Rysunek 3-6. Panel konfiguracyjny [opracowanie własne].	26
Rysunek 3-7. Interfejs monitorowania kanałów [opracowanie własne].	27
Rysunek 3-8 Wątki oraz ich funkcje [opracowanie własne].....	28
Rysunek 3-9. Komunikacja w obrębie modułu GUI [opracowanie własne].....	28
Rysunek 3-10. Funkcjonalności systemu czasu rzeczywistego [opracowanie własne].	30
Rysunek 3-11. Wątki i ich funkcje [opracowanie własne].....	31
Rysunek 3-12. Komunikacja w obrębie modułu Real-Time [opracowanie własne].....	32
Rysunek 3-13. Funkcjonalności modułu FPGA [opracowanie własne].....	32
Rysunek 3-14. Podmoduły i ich funkcje [opracowanie własne].	33
Rysunek 3-15. Komunikacja w obrębie modułu FPGA [opracowanie własne].....	34
Rysunek 3-16. Zależność generowanego napięcia od różnicy temperatur [22].....	34
Rysunek 3-17. Obszar pamięci wypełniony przeliczonymi wartościami [opracowanie własne].	35
Rysunek 3-18. Wykres zależności błędu od indeksu w tabeli przeglądowej.....	36
Rysunek 3-19. Pętla sterująca powielona czterokrotnie [opracowanie własne].	37
Rysunek 3-20. Pętla iterująca przez dane wejściowe [opracowanie własne].....	38
Rysunek 4-1. Stanowisko testowe.....	39