

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH



Instytut Automatyki i Informatyki Stosowanej

Praca dyplomowa inżynierska

na kierunku Automatyka i Robotyka

Implementacja oraz wdrożenie platformy ALFRED
w laboratorium detektora ALICE – FIT

Ignacy Mermer
Numer albumu 324873

promotor
dr. inż. Grzegorz Kasprowicz

WARSZAWA 2025

Serdecznie dziękuję mgr inż. Krystianowi Rostonowi za nieocenione wsparcie i pomoc udzielone w trakcie realizacji niniejszej pracy inżynierskiej oraz podczas mojego pobytu w CERN.

Implementacja oraz wdrożenie platformy ALFRED w laboratorium detektora ALICE – FIT

Streszczenie. Przedmiotem niniejszej pracy inżynierskiej była implementacja oraz wdrożenie platformy ALFRED w laboratorium detektora ALICE-FIT, Eksperyment ALICE jest jednym z czterech głównych eksperymentów na Wielkim Zderzaczu Hadronów (LHC) w CERN. Kluczową rolę w tym eksperymencie odgrywa zestaw detektorów FIT, który odpowiada za precyzyjną synchronizację i wyzwalanie akwizycji danych dla innych detektorów.

Prace badawczo-rozwojowe, prowadzone w CERN, skupiały się na modernizacji istniejącej architektury systemu kontroli i sterowania. Autor, jako członek zespołu ALICE-FIT DCS, uczestniczył w rozwoju systemu, którego dotychczasowa struktura, oparta na aplikacjach Control Server i HistogramReader, nie była w pełni zgodna ze standardami zespołu Central ALICE DCS.

Głównym celem było zastąpienie dotychczasowych rozwiązań platformą ALFRED, co wymagało jej adaptacji dla specyfiki detektorów FIT, wykorzystujących protokołów IPbus zamiast standardowego GBT. W ramach pracy stworzono i wdrożono system SCADA w środowisku WinCC OA, który stanowi integralną część nowej architektury. System ten, wykorzystując framework JCOP oraz protokołów komunikacyjny DIM, zapewnia pełną kontrolę nad elektroniką FEE, integrację z bazą danych Oracle oraz systemami centralnymi eksperymentu. Zaprojektowano i zaimplementowano złożony interfejs użytkownika, który nie tylko odwzorowuje, ale i rozszerza funkcjonalności poprzednich aplikacji, umożliwiając m.in. zaawansowane zarządzanie konfiguracjami, wizualizację histogramów oraz automatyzację procedur kalibracyjnych.

Opracowany system został wdrożony i przetestowany zarówno w laboratorium, jak i w docelowej infrastrukturze detektora FV0, co potwierdziło jego stabilność i wydajność. Finalnym rezultatem pracy jest w pełni funkcjonalna, zgodna ze standardami ALICE platforma do sterowania i monitorowania detektorów FIT, która docelowo zostanie zaimplementowana na trzech poddetektorach.

Realizacja niniejszej pracy została zapoczątkowana w ramach programu stażowego SPARC 2024. Projekt był następnie rozwijany w okresie od września 2024 do lutego 2025 roku podczas integracji systemu w laboratorium na Wydziale Elektroniki i Technik Informatycznych Politechniki Warszawskiej. Finalny etap, obejmujący wdrożenie, testy oraz dalszy rozwój w docelowej infrastrukturze, został zrealizowany podczas pobytów w CERN między lutym a sierpniem 2025 roku.

Słowa kluczowe: ALFRED, ALICE, CERN, DCS, DIM, FEE, FIT, JCOP, LHC, SCADA, WinCC OA

Implementation and Deployment of the ALFRED Platform in the ALICE-FIT Detector Laboratory.

Abstract. The subject of this thesis was the implementation and deployment of the ALFRED platform in the ALICE-FIT detector laboratory. The ALICE experiment is one of four major experiments at the Large Hadron Collider (LHC) at CERN. The key element of this experiment are the FIT detectors, which are responsible for the precise synchronization and triggering of data acquisition for other detectors.

The research and development work, conducted at CERN, focused on the modernization of the existing control system architecture. The author, as a member of the ALICE-FIT DCS team, participated in the development of the system, as its previous structure, based on the Control Server and HistogramReader applications, was not fully compliant with the standards of the Central ALICE DCS Team.

The main goal was to replace the existing solutions with the integrated ALFRED platform, which required adaptation to the specific needs of the FIT detectors, which use the IPbus protocol instead of the standard GBT. As part of the thesis, a SCADA system was developed and deployed in the WinCC OA environment, which forms an integral part of the new architecture. Using the JCOP framework and the DIM communication protocol, this system provides full control over the FEE, integration with the Oracle database, and the central experiment systems. A complex user interface was designed and implemented, which not only replicates but also extends the functionalities of the previous applications, enabling, among other things, advanced configuration management, histogram visualization, and the automation of calibration procedures.

The developed system was implemented and tested both in the laboratory environment and on the target infrastructure of the FV0 detector, confirming its stability and performance. The final result of this work is a fully functional, ALICE-compliant platform for controlling and monitoring FIT detectors, which will ultimately be implemented on three detectors.

The development of this thesis was initiated as part of the SPARC 2024 internship program. The project was then developed from September 2024 to February 2025 during system integration in a laboratory at the Faculty of Electronics and Information Technology, Warsaw University of Technology. The final stage, including implementation, testing, and further development in the target infrastructure, was completed during stays at CERN between February and August 2025.

Keywords: ALFRED, ALICE, CERN, DCS, DIM, FEE, FIT, JCOP, LHC, SCADA, WinCC OA

Spis treści

1. Wstęp	6
1.1. CERN (Europejska Europejska Organizacja Badań Jądrowych)	6
1.2. Układ akcelacyjny	6
1.3. Eksperymenty na LHC	8
1.4. Multidetektor ALICE	9
1.5. Zestaw detektorów FIT	9
1.5.1. Struktura detekcyjna	10
1.5.2. Elektroniczny system odczytowy	11
2. Architektura systemu kontroli i sterowania zestawem detektorów ALICE-FIT	14
2.1. Aktualna architektura	14
2.1.1. Aplikacja Control Server	15
2.1.2. Aplikacja HistogramReader	16
2.1.3. System SCADA (Supervisory Control and Data Acquisition)	17
2.2. Domyślna architektura	18
3. Cel, zakres prac i kontrybucja autora	21
4. Przegląd dostępnych rozwiązań	23
4.1. EPICS (Experimental Physics and Industrial Control System)	23
4.2. Tango Controls	23
4.3. ALFRED	24
4.4. SIMATIC WinCC OA (Open Architecture)	26
5. Rozwój i wdrożenie systemu ALFRED	28
5.1. Wprowadzone rozwiązania	28
5.2. Integracja systemu w laboratorium ALICE-FIT	36
5.3. Integracja systemu w detektorze FV0	37
6. Rozwój i wdrożenie systemu SCADA	39
6.1. Wprowadzone Rozwiązania w Systemie SCADA WinCC OA	39
6.2. Integracja systemu w laboratorium ALICE-FIT	59
6.3. Integracja Systemu dla detektora FV0	61
7. Podsumowanie	67
Bibliografia	69
Wykaz symboli i skrótów	72
Spis rysunków	74
Spis tabel	74
Spis załączników	74

1. Wstęp

1.1. CERN (Europejska Europejska Organizacja Badań Jądrowych)

CERN[1] (fr. Organisation Européenne pour la Recherche Nucleaire) jest jednym z największych i najważniejszych ośrodków badawczych na świecie. Znajduje się na terytorium Szwajcarii i Francji. W tym momencie do grupy państw członkowskich zalicza się 23 kraje, w tym Polskę od 1991 roku. Organizacja ta została założona w 1954 roku i od samego początku wspiera współpracę międzynarodową w dziedzinie fizyki cząstek elementarnych[2]

CERN, od samego początku, pełni kluczową rolę w rozwoju fizyki wysokich energii. Uda się to dzięki prowadzeniu eksperymentów mających na celu zgłębianie natury Wszechświata na najbardziej podstawowym poziomie. Ośrodek ten jest znany z budowy i wykorzystania zaawansowanych akceleratorów cząstek. To dzięki nim udało dokonać się licznych odkryć naukowych, w tym potwierdzenie istnienia bozonu Higgsa w 2012 roku [3]. CERN jest także miejscem, gdzie narodziła się sieć WWW (World Wide Web), która została stworzona przez Tima Bernersa-Lee w 1989 roku. Miało to bardzo istotny wpływ na rozwój wymiany danych na świecie [4].

W CERN pracuje 2500 osób oraz ponad 12 200 naukowców i inżynierów z całego świata. Większość z nich jest zaangażowana w badania, jednakże duża grupa jest odpowiedzialna za rozwój technologii potrzebnej do prowadzenia eksperymentów [5]. Badania, które są tam prowadzone, oprócz fizyki cząstek elementarnych, mają zastosowanie również w medycynie (m.in. w technice obrazowania oraz terapii hadronowej), informatyce (w zaawansowanych systemach przetwarzania danych) oraz inżynierii (w rozwoju nadprzewodników i technologii kriogenicznych). Dzięki szerokim możliwościom badawczym, oferowanym przez CERN, jest on kluczowym miejscem dla rozwoju współczesnej nauki i technologii.

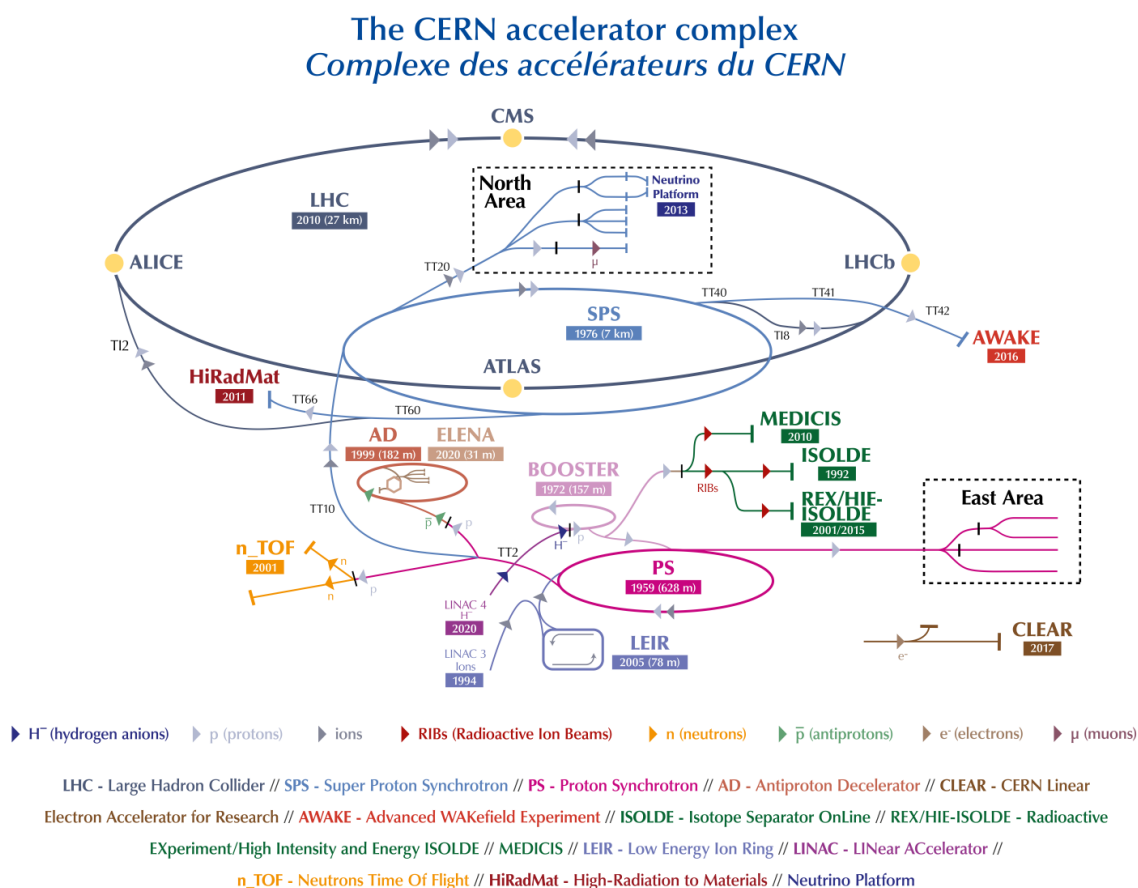
1.2. Układ akceleryjny

Budowa Wielkiego Zderzacza Hadronów LHC (Large Hadron Collider), największego akceleratora cząstek na świecie, jest jednym z najważniejszych osiągnięć CERN. Ma on 27 kilometrów długości i znajduje się w tunelu, około 100 metrów pod powierzchnią ziemi.[6]

Przyspieszanie cząstek w CERN odbywa się w kilku etapach przy wykorzystaniu kolejnych akceleratorów wspomagających. Proces rozpoczyna się w akceleratorze liniowym LINAC4[7] (Linear Accelerator 4). Protony, które uzyskane są podczas oddzielenia elektronów od atomów wodoru, osiągają w nim energię 160 MeV, a następnie trafiają do PSB[8] (Proton Synchrotron Booster), gdzie ich energia wzrasta do 2 GeV. W kolejnym etapie cząstki są przenoszone do PS[9] (Proton Synchrotron), który przyspiesza je do 26 GeV. W przedostatnim etapie trafiają do SPS[9] (Super Proton Synchrotron), gdzie energia cząstek osiąga 450 GeV. W końcowej fazie w LHC, protony uzyskują energię zderzenia w środku

masy o wartości $\sqrt{s} = 13,6 TeV$, co umożliwia przeprowadzanie zderzeń o najwyższych energiach uzyskiwanych w laboratoriach na świecie[10].

W LHC przeprowadzane są również eksperymenty zderzeń ciężkich jonów, między innymi jonów ołowiu, które otrzymywane są przy wykorzystaniu rezonansu cyklotronowego elektronów[11]. Następnie jony przyspieszane są w LINAC3 (Linear Accelerator 3), a w kolejnym kroku przeprowadzana jest selekcja tych jonów[12]. Po ich odfiltrowaniu, system RF (Radio Frequency) przyspiesza ustalone typy jonów do energii 4,2MeV/u, aby następnie skierować je na folię węglową[13]. Dzięki niej można pozbyć się elektronów i uzyskać jony $^{208}\text{Pb}^{53+}$, które wprowadzane są do akceleratorów synchrotronowych, gdzie ich energia stopniowo wzrasta w kolejnych synchrotronach takich jak PS, SPS oraz LHC. Dzięki temu, jony $^{208}\text{Pb}^{82+}$ osiągają energię zderzenia w środku masy $\sqrt{s_{nn}} = 6,8 TeV$ w LHC[14], co umożliwia prowadzenie badań między innymi nad stanem materii, którym prawdopodobnie występował we wczesnym Wszechświecie. Schemat całego kompleksu akceleratoryjnego CERN, w którym odbywają się opisane procesy, przedstawiono na rysunku 1.1.



Rysunek 1.1. Schemat ukazujący cały kompleks akceleratoryjny CERN [15]

Sama struktura LHC wykorzystuje komory zwane *beam pipe*, w których cząstki poru-

szają się w ultrawysokiej próżni o ciśnieniu 10^{-13} atmosfery. Dzięki temu możliwe jest uniknięcie niepożądanych kolizji z innymi cząstkami[16]. *Beam pipe* składa się z dwóch przewodów które są względem siebie równoległe i przecinają się w miejscach czterech eksperymentów przeprowadzanych na LHC.

1.3. Eksperymenty na LHC

Dzięki możliwościom LHC, w jego obrębie prowadzone są cztery eksperymenty. Każdy z nich ma swoje specyficzne cele i stosuje inne metody badawcze. Do tej grupy zaliczają się:

- **ATLAS** [17] (**A Toroidal LHC ApparatuS**) – to największy multidetektor na LHC. Jego celem jest badanie szerokiego spektrum zjawisk fizycznych, w tym testowanie Modelu Standardowego[3] opisującego cząstki elementarne i fundamentalne oddziaływania. ATLAS zajmuje się również poszukiwaniem nowych cząstek i zjawisk wykraczających poza obecne teorie.
Jednym z najważniejszych osiągnięć ATLAS-u było współodkrycie bozonu Higgsa w 2012 roku. Jest to cząstka przewidywana przez Model Standardowy, która odpowiada za nadawanie masy innym cząstkom.
- **CMS** [18] (**Compact Muon Solenoid**) – to drugi z wielkoskalowych multidetektorów na LHC. Wyróżnia go magnes solenoidowy, generujący pole magnetyczne o wartości 4T, co czyni go szczególnie efektywnym podczas wykrywania mionów, czyli kluczowych cząstek w wielu badanych procesach.
Podobnie jak ATLAS, CMS odegrał zasadniczą rolę w odkryciu bozonu Higgsa. Eksperyment ten koncentruje się również na poszukiwaniu dowodów na istnienie ciemnej materii, badaniu rzadkich rozpadów cząstek oraz produkcji ciężkich kwarków.
- **ALICE** [14] (**A Large Ion Collider Experiment**) – to eksperyment na LHC specjalizujący się w fizyce zderzeń ciężkich jonów. Został opisany w rozdziale 1.4.
- **LHCb** [19] (**Large Hadron Collider beauty experiment**) – to specjalistyczny eksperyment na LHC, który wyróżnia się metodą badawczą, w której koncentruje się na cząstkach produkowanych pod małymi kątami wzdłuż osi wiązki. Jego głównym celem jest precyzyjna analiza mezonów B, zawierających kwark b, dzięki czemu możliwe jest badanie subtelnych różnic między materią a antymaterią. LHCb poszukuje odchyleń od przewidywań Modelu Standardowego, aby pomóc wyjaśnić dominację materii nad antymaterią we Wszechświecie.

Eksperymenty na LHC nie działają osobno, ich wyniki są wzajemnie porównywane i uzupełniają się, dzięki czemu wiarygodność odkryć wzrasta. Każdy z nich wnosi swój unikalny wkład w zrozumienie struktury materii, ewolucji Wszechświata oraz próbuje odkryć nowe prawa fizyki.

1.4. Multidetektor ALICE

Eksperyment ALICE jest jednym z głównych projektów badawczych prowadzonych na LHC. Jego głównym zadaniem jest analiza plazmy kwarkowo-gluonowej, która powstaje w wyniku zderzeń ciężkich jonów, na przykład takich jak ołów, przy bardzo wysokich energiach [20]. W takich warunkach kwarki i gluony, normalnie uwięzione wewnątrz protonów i neutronów, stają się bardzo krótko swobodne. ALICE bada proces powstawania plazmy kwarkowo-gluonowej, jej rozwój oraz przejście z powrotem w zwykłą materię hadronową [14].

ALICE składa się z szeregu detektorów, które umożliwiają dokładną analizę właściwości cząstek powstających w wyniku zderzeń [14]. Są to między innymi:

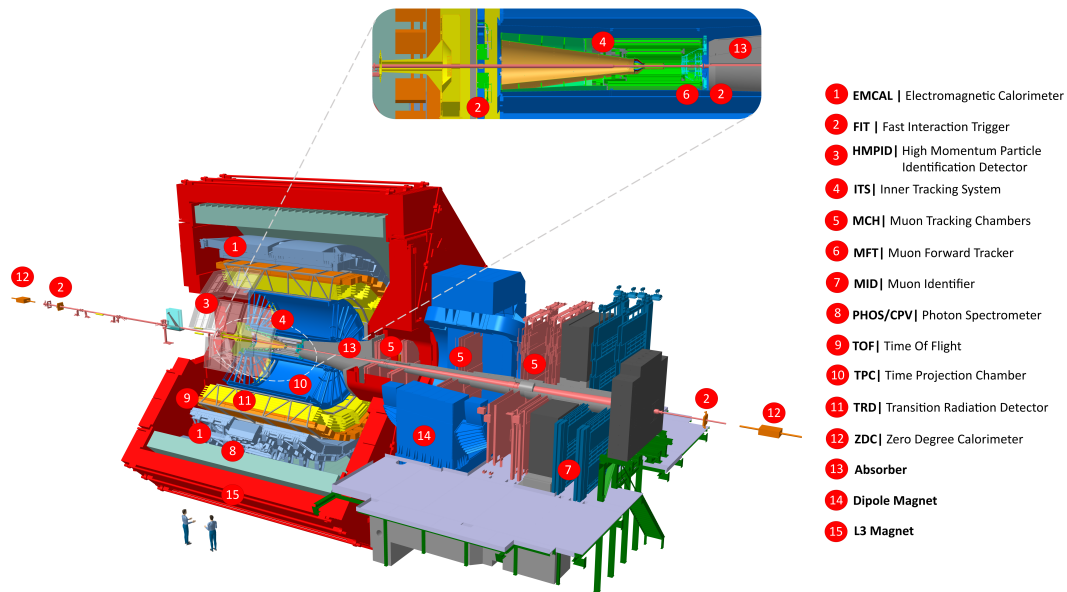
- **ITS** (Inner Tracking System) – zapewnia precyzyjne śledzenie trajektorii cząstek w pobliżu punktu zderzenia.
- **TPC** (Time Projection Chamber) – odpowiada za rekonstrukcję torów cząstek o niskim i średnim pędzie.
- **TRD** (Transition Radiation Detector) – służy do identyfikacji elektronów.
- **TOF** (Time of Flight) – pozwala na pomiar czasu przelotu cząstek, co ułatwia ich identyfikację.
- **EMCal** (Electromagnetic Calorimeter) – mierzy energię elektronów, pozytonów i fotonów.
- **FIT** [21] (Fast Interaction Trigger) – system poprawiający synchronizację detekcji oraz umożliwiający szybkie rejestrowanie zdarzeń.

Zespół detektorów ALICE ukazany jest Rysunku 1.2.

Jednym z największych wyzwań w eksperymentach na LHC, jest bardzo wysoka częstotliwość zderzeń. Pakiety cząstek przechodzą przez detektor z częstotliwością wynoszącą do 40 MHz, co generuje dużą ilość danych. Aby sprostać temu zadaniu, ALICE został wyposażony w zaawansowane systemy akwizycji i przetwarzania danych. W ostatnich latach zostały dokonane znaczące modernizacje, które umożliwiły zwiększenie jego wydajności.

1.5. Zestaw detektorów FIT

System detektorów FIT został zaimplementowany w eksperymencie ALICE w celu ulepszenia koordynacji czasowej jego wielu komponentów. Aktualnie dzięki FIT możliwe jest błyskawiczne inicjowanie akwizycji danych wyłącznie dla tych kolizji cząstek, które odpowiadają zdefiniowanemu kryteriom fizycznym, co pozwala na znaczące odciążenie całości systemu. Na architekturę FIT składają się trzy systemy detekcyjne: FT0, FV0 oraz FDD. Zostały one zintegrowane z multidetektem podczas drugiej długiej przerwy technicznej akceleratora LHC (Long Shutdown 2), która miała miejsce w latach 2018–2022. Każdy z wymienionych detektorów realizuje inne, wyspecjalizowane zadania, które obejmują między innymi wyzwalanie akwizycji danych, pomiar centralności zderzeń, a także ciągłe monitorowanie warunków tła oraz parametrów wiązki.



Rysunek 1.2. Schemat multidetektora ALICE [22]

1.5.1. Struktura detekcyjna

Detektor FT0 ma postać podwójnego układu detektorów Czerenkowa, rozmieszczonych symetrycznie po obu stronach punktu zderzeń (IP – Interaction Point). Konstrukcja ta obejmuje dwa zestawy, z których pierwszy zawiera 96 kanałów detekcyjnych, a drugi 112, co sumarycznie daje 208 niezależnych kanałów odczytu.

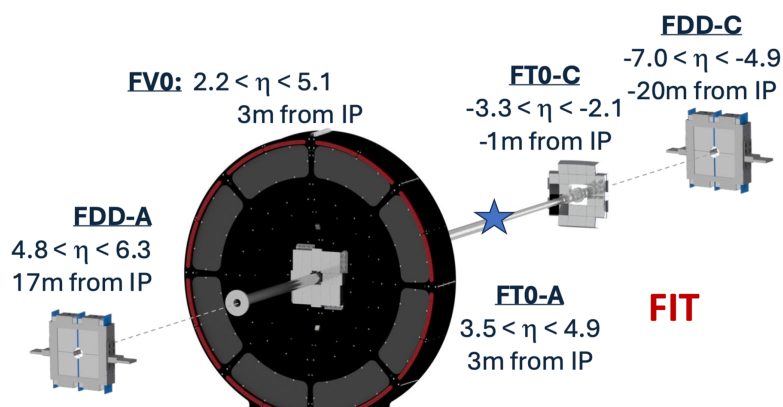
Zasada działania FT0 opiera się na zjawisku Czerenkowa: naładowane cząstki, które poruszają się w ośrodku z prędkością przewyższającą prędkość fazową światła, emitują charakterystyczne promieniowanie [14]. To właśnie ono jest następnie przechwytywane i rejestrowane przez fotopowielacze mikrokanałowe (MCP-PMT – Micro Channel Plate - Photomultiplier Tube), co gwarantuje niezwykle krótki czas reakcji całego systemu. Opóźnienie sygnału nie przekracza 425 ns, natomiast osiągnięta rozdzielczość czasowa jest rzędu 4,4 ps w przypadku zderzeń jąder ołowiu (Pb–Pb) oraz 17 ps dla kolizji proton-proton (p–p). Z tego powodu, jest to bardzo istotny komponent wykorzystywany w celu dostarczania precyzyjnego sygnału czasowego między innymi dla detektora TOF, jest to kluczowe w procesie identyfikacji cząstek. [23]

Kolejny detektor, FV0, jest detektorem scyntylacyjnym o charakterystycznej, dyskowej budowie. Różni się od systemów FT0 i FDD tym, że jest on umiejscowiony tylko po jednej stronie punktu interakcji. Jego struktura została zaprojektowana jako pięć współśrodkowych pierścieni, przy czym każdy z nich podzielono na segmenty obejmujące kąt 45 stopni.

W najbardziej zewnętrznym pierścieniu zastosowano zdwojoną liczbę fotodetektorów dla każdego z optycznie izolowanych sektorów, co przekłada się na 48 niezależnych kanałów odczytowych [21].

Detektor FDD został skonstruowany z myślą o badaniach między innymi zjawisk fizyki dyfrakcyjnej [21]. Jego specyficzna budowa umożliwia detekcję cząstek emitowanych pod bardzo małymi kątami w stosunku do osi wiązki, co ma fundamentalne znaczenie dla analizy procesów dyfrakcyjnych.

Na rysunku 1.5 ukazane są elementy detektora FIT z pokryciem pseudopospieszności [18] oraz z pokazaniem usytuowania względem punktu interakcji.

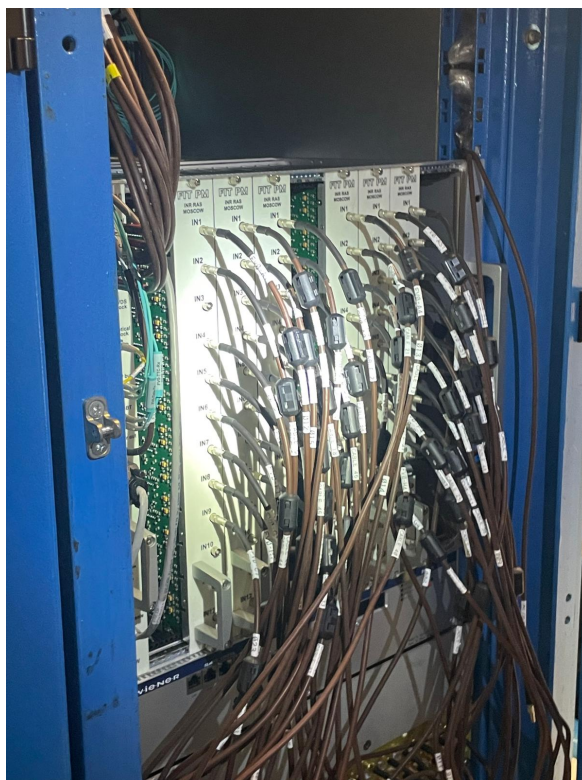


Rysunek 1.3. Ilustracja elementów detektora FIT z pokryciem pseudopospieszności oraz odległością od punktu interakcji [24]

1.5.2. Elektroniczny system odczytowy

System akwizycji danych dla poddetektorów FIT skonstruowany jest wokół dwóch fundamentalnych elementów. Pierwszym z nich jest system zbudowany na bazie kraty VME firmy Wiener, stanowiący miejsce montażu dla modułu TCM (Trigger&Clock Module) oraz dla płyt PM (Process Module). Płyty PM są odpowiedzialne za akwizycję i wstępne przetwarzanie sygnałów analogowych pochodzących z poszczególnych sensorów światłoczułych. Każda płyta PM umożliwia obsługę do 12 kanałów detekcyjnych. Każdy z tych kanałów jest wyposażony w dwa 12-bitowe przetworniki analogowo-cyfrowe ADC (Analog - Digital Converter), które pracują w trybie naprzemiennym, aby zapewnić ciągłość procesu zbierania danych. Podczas gdy jeden przetwornik dokonuje konwersji, dane z drugiego są odczytywane. W przypadku detektora FT0, sygnały cyfrowe dostarczane są bezpośrednio z fotopowielaczy MCP-PMT, natomiast dla FV0 i FDD źródłem sygnałów są fotopowielacze PMT. Liczba użytych płyt PM jest specyficzna dla każdego z detektorów: FT0 wykorzystuje 18 płyt PM (co jest podyktowane dużą liczbą kanałów detektora, wynoszącą 208), FV0 korzysta z 6 płyt PM, co jest wymagane dla obsługi 48 kanałów odczytowych, natomiast FDD operuje na 2 płytach PM, co jest wystarczające dla mniejszej liczby kanałów. Płyta TCM z kolei pełni rolę centralnej jednostki koordynującej. Wszystkie moduły PM przypi-

sane do danego detektora są podłączone do pojedynczego modułu TCM. Zapewnia on między innymi synchronizację zbieranych danych z sygnałem zegarowym otrzymywanym z systemów LHC oraz umożliwia integrację z innymi systemami pomocniczymi, takimi jak system laserowy czy system kontroli i sterowania detektora (DCS – Detector Control System), poprzez protokół IPbus [25]. Transmisja danych pomiędzy modułami TCM a PM realizowana jest z wykorzystaniem przewodów HDMI poprzez protokół SPI.



Rysunek 1.4. System elektroniki odczytowej (FEE) detektora FV0 w eksperymencie ALICE

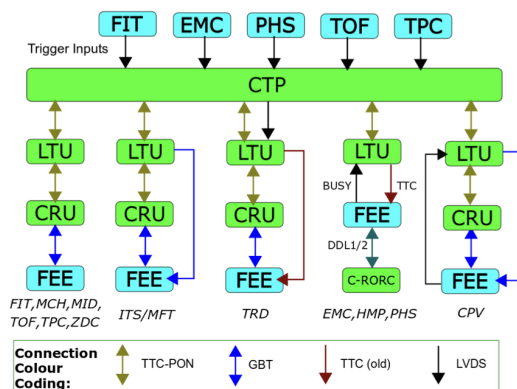
Drugim istotnym elementem infrastruktury detektora jest krata wysokiego napięcia CAEN SY1527, która funkcjonuje niezależnie od kraty VME. Jest ona odpowiedzialna za dostarczanie zasilania dla fotopowielaczy zainstalowanych w detektorach, zapewniając tym samym stabilne warunki pracy dla tych komponentów. System ten może dostarczać napięcie w zakresie od 0 do 3 kV przy mocy do 1,5W na kanał.

W konfiguracji w laboratorium ALICE-FIT dostępne są następujące komponenty: krata Wiener wyposażona w jedną płytę TCM oraz dwie płyty PM (jedna dla detektora FT0, druga dla detektora FDD), podłączone do wejść oznaczonych jako PMA0 oraz PMC0. Laboratorium dysponuje również kratą wysokiego napięcia CAEN, zawierającą dwa moduły kanałowe. Oprócz tych podstawowych elementów, w skład laboratoryjnego zestawu wchodzi także:

- **FLP (First Level Processor – Procesor Pierwszego Poziomu)** – FLP zajmuje się wstępnym przetwarzaniem danych napływających z systemów niższego poziomu hierarchii odczytu. Realizuje ono operacje takie jak filtrowanie, agregacja oraz kompensacja

danych, co pozwala na znaczące zredukowanie ich objętości przed przesłaniem do centralnej infrastruktury obliczeniowej[26].

- **LTU (Local Trigger Unit – Lokalna Jednostka Wyzwalająca)** – stanowi część systemu wyzwalania detektorów w eksperymencie ALICE[27].
- **CRU (Common Readout Unit – Wspólna Jednostka Odczytowa)** – jest kluczowym elementem systemu zbierania danych w eksperymencie ALICE, pełniącym rolę interfejsu pomiędzy elektroniką odczytu detektora a jednostkami obliczeniowymi (FLP)[27].



Rysunek 1.5. Schemat centralnego systemu trigger'ów ukazujący wejścia do CTP (Central Trigger Processor) i typy połączeń między CTP i FEE (FrontEnd Electronics) detektora. [28]

2. Architektura systemu kontroli i sterowania zestawem detektorów ALICE-FIT

2.1. Aktualna architektura

Jak wspomniano w poprzednim rozdziale, system poddetektorów FIT generuje olbrzymie ilości danych, przy częstotliwości zderzeń wynoszącej 40MHz, narzuca to konieczność stosowania zaawansowanej infrastruktury elektronicznej wspartej kompleksowym systemem zarządzania i kontroli DCS. Centralnym elementem tej infrastruktury jest system elektroniki odczytowej detektora FEE, która odpowiada za akwizycję i przetwarzanie sygnałów bezpośrednio z danego detektora. Elektronika FEE dysponuje również zintegrowanymi mechanizmami autokonfiguracji, oferuje możliwość zmiany danych w rejestrach oraz monitorowania liczników, czyli zliczeń ładunków zarejestrowanych przez fotopowielacze, co jest fundamentalne dla śledzenia intensywności oddziaływań. System sterowania i kontroli musi gwarantować bezawaryjną komunikację pomiędzy wszystkimi komponentami, a jednocześnie zapewniać ekspertom łatwy dostęp do zarządzania zarówno globalnymi konfiguracjami systemów elektronicznego i wysokiego napięcia, jak i parametrami ich poszczególnych elementów. Dotyczy to między innymi kontroli nad takimi aspektami jak aktywacja/dezaktywacja lasera, regulacja jego częstotliwości oraz zarządzanie innymi parametrami na poszczególnych kanałach płyt PM.

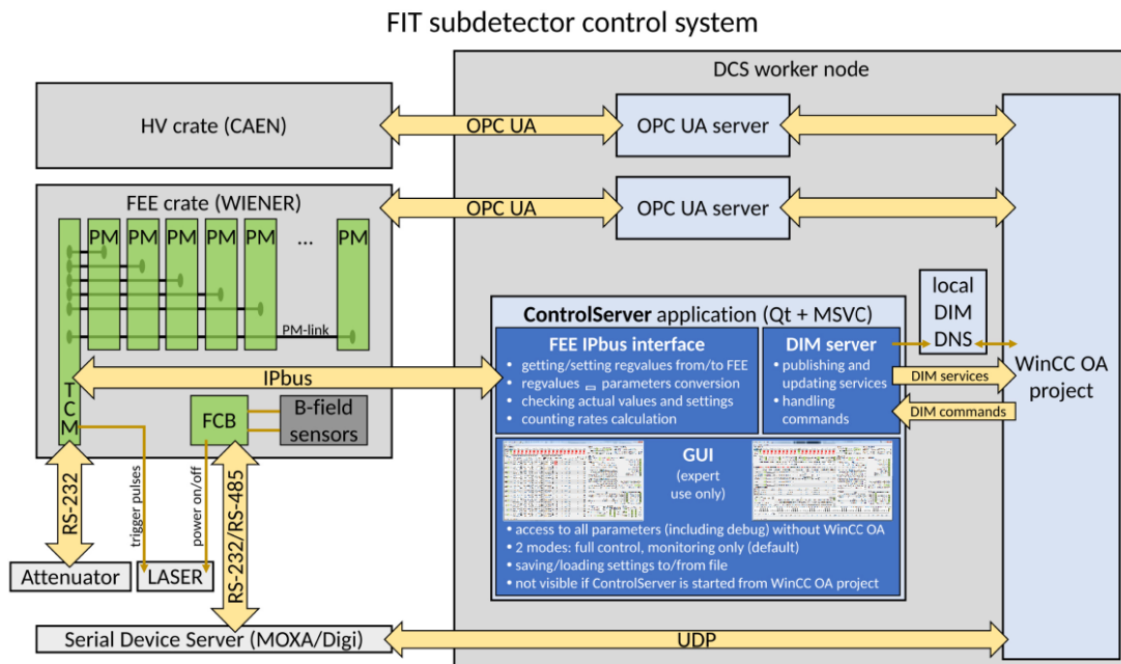
Struktura systemu kontroli i sterowania detektora FIT opiera się na kilku zasadniczych filarach:

- Główny komputer dedykowany zarządzaniu detektorem (DCS Worker Node): Pełni rolę centralnego komputera systemu, na nim uruchomiony jest projekt systemu WinCC OA, oraz koordynuje on komunikacją z komponentami peryferyjnymi (krata VME, krata CAEN).
- Komputer dedykowany zarządzaniu detektorem dla ekspertów (DCS Operator Node): Pozwala na uruchomienie interfejsu użytkownika dla systemu WinCC OA. Posiada jednak ograniczenia na funkcjonalności, które mogłyby wprowadzić problemy podczas działania projektu.
- Krata wysokiego napięcia CAEN (HV Crate): Odpowiada za precyzyjne dostarczanie i monitorowanie wysokiego napięcia zasilającego fotopowielacze.
- Krata VME: Zapewnia zasilanie niskonapięciowe, chłodzenie oraz zarządzanie stanem uruchomienia elektroniki FEE detektora.

Węzeł DCS Worker Node operuje w ramach wewnętrznej sieci AliDCS, która jest fizycznie odizolowana od sieci GPN (Global Private Network), co stanowi gwarancję bezpieczeństwa i stabilności systemu. Komunikacja z kratami HV oraz VME realizowana jest przy użyciu protokołu OPC UA (Open Protocol Communication Unified Architecture) [29]. Protokół ten umożliwia zdalne włączanie i wyłączanie zasilania, a także ciągłe monitorowanie

parametrów takich jak napięcie, natężenie prądu, temperatura oraz status działania wentylatorów.

Dodatkowo, do komunikacji z systemem elektronicznym detektora wykorzystywany jest protokół IPbus. Jest to protokół komunikacyjny bazujący na UDP (User Datagram Protocol), zoptymalizowany pod kątem wymiany danych z elektroniką FEE i zaprojektowany specjalnie na potrzeby eksperymentów fizyki wysokich energii, takich jak ALICE. IPbus charakteryzuje się wysoką przepustowością o wartości 0,5Gbit/s, skalowalnością oraz niezawodnością. Umożliwia on dostęp do rejestrów płyt TCM i PM, co pozwala na zarządzanie ich stanem oraz na przesyłanie danych pomiarowych w czasie rzeczywistym.



Rysunek 2.1. Schemat architektury DCS FIT [30]

2.1.1. Aplikacja Control Server

Jednym z głównych komponentów systemu DCS dla detektorów FIT jest aplikacja Control Server. Aplikacja działająca na platformie DCS Worker Node, dostarcza ekspertom rozbudowany interfejs graficzny, który umożliwia zarządzanie i kontrolę nad elektroniką FEE.

Do kluczowych funkcjonalności należą:

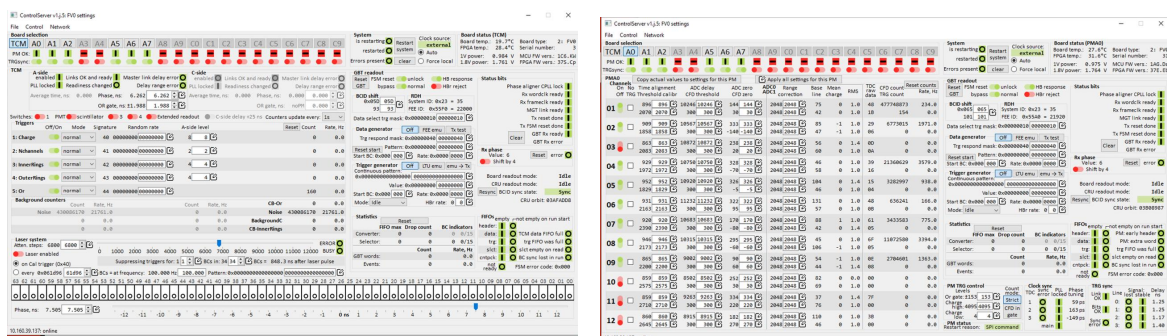
- Inicjowanie pracy FEE: obejmujące automatyczną identyfikację podłączonych oraz aktywnych modułów PM.
- Akwizycja i odświeżanie danych: cykliczne gromadzenie podstawowych informacji, takich jak wartości liczników (Counter'y) oraz wskaźniki częstości zdarzeń (Rate'y)

2. Architektura systemu kontroli i sterowania zestawem detektorów ALICE-FIT

– liczba zarejestrowanych cząstek w jednostce czasu), które następnie są przekazywane do dalszych analiz.

- Detekcja i sygnalizacja błędów: ciągłe monitorowanie systemu pod kątem identyfikacji wszelkich nieprawidłowości w jego funkcjonowaniu.
- Konfiguracja parametrów FEE: ustawienie odpowiednich wartości w rejestrach płyt PM i TCM dla typu prowadzonych badań, na podstawie pliku konfiguracyjnego .cfg.
- Dystrybucja danych: transfer wybranych danych pomiarowych do systemu SCADA za pośrednictwem lokalnego serwera DIM (Distributed Information Management) [31].

Rozbudowana warstwa backendowa aplikacji Control Server zapewnia komunikację z modułem TCM przy użyciu protokołu IPbus. W tym celu wykorzystywane jest połączenie światłowodowe od płyty TCM do switch'a. Dzięki zastosowaniu systemu DIM, aplikacja jest w stanie publikować pozyskane i przetworzone dane z elektroniki do systemu SCADA, który z kolei posiada połączenia z centralnymi systemami eksperymentu ALICE.



Rysunek 2.2. Ekran panelu TCM i PMA0 w aplikacji Control Server

2.1.2. Aplikacja HistogramReader

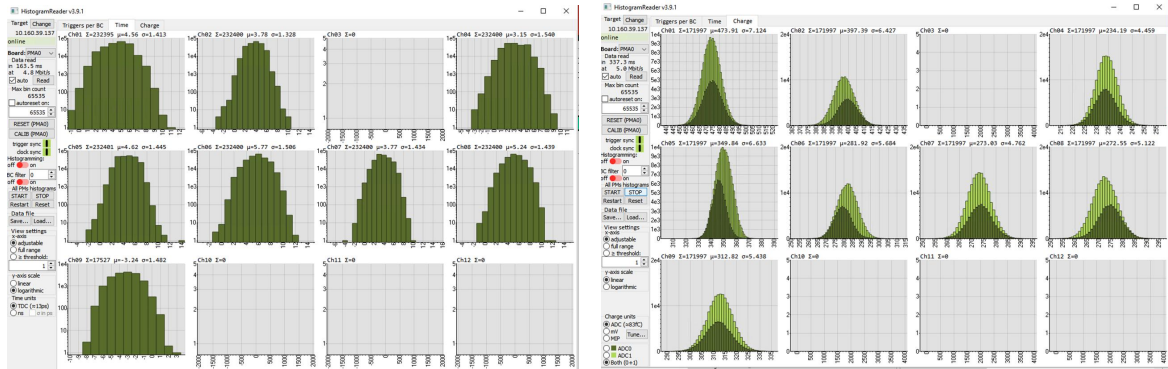
Drugim kluczowym oprogramowaniem zainstalowanym na DCS Worker Node i dedykowanym poddetektorom FIT jest HistogramReader. Narzędzie to pozwala ekspertom na prowadzenie szczegółowych analiz charakterystyk zarejestrowanych cząstek w poszczególnych przedziałach (binach) dla wybranego kanału w konkretnym module PM.

Aplikacja ta również komunikuje się z TCM za pośrednictwem protokołu IPbus, wykorzystując jego zaawansowane możliwości, takie jak:

- Odczyt pełnych bloków rejestrów FEE: co umożliwia efektywne pozyskiwanie obszernych zestawów danych konfiguracyjnych oraz informacji o stanie systemu.
- Odczyt całego bloku danych z rejestru typu FIFO (First In First Out): co pozwala na szybkie pobieranie strumieni danych bezpośrednio z buforów detektora.

Oprócz funkcji analitycznych, oprogramowanie HistogramReader umożliwia również realizację złożonych procedur operacyjnych, takich jak kalibracja parametrów elektroniki

FEE. Proces ten obejmuje sekwencję zadań wykonywanych przez zróżnicowane algorytmy zaimplementowane w kodzie aplikacji, co zapewnia precyzyjne dobranie parametrów systemu oraz analizę zachowania elektroniki i samego detektora z wykorzystaniem prezentowanych danych i wykresów.



Rysunek 2.3. Ekran zakładki Time i Charge dla płyty PMA0 w aplikacji HistogramReader

2.1.3. System SCADA (Supervisory Control and Data Acquisition)

Platforma WinCC OA (WinCC Open Architecture) [32] reprezentuje najwyższą warstwę w hierarchii systemu sterowania i kontroli detektora. Jest to zaawansowany system klasy SCADA, zaprojektowany tak aby sprostać wysokim wymaganiom eksperymentów fizyki wysokich energii oraz aplikacjom w sektorze energetyki jądrowej. W eksperymencie ALICE, WinCC OA jest uruchomione na komputerze o nazwie DCS Worker Node i pełni funkcję centralnego oprogramowania integrującego wszystkie komponenty systemu detektora.

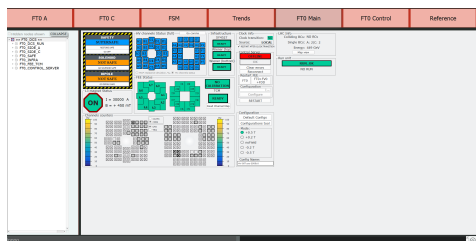
WinCC OA wchodzi w interakcje z poszczególnymi elementami systemu w następujący sposób:

- Z kratami HV CAEN oraz VME firmy Wiener: komunikacja odbywa się poprzez protokół OPC UA.
- Z aplikacją Control Server: wymiana danych następuje z wykorzystaniem mechanizmów DIM, poprzez subskrypcję serwisów publikowanych w sieci lokalnej (localhost).
- Z ogólnymi systemami eksperymentu ALICE, takimi jak FSM (Finite State Machine) [33]: w celu zarządzania stanami operacyjnymi detektora, co zapewnia jego synchronizację i koordynację działań w ramach całego eksperymentu.

System WinCC OA oferuje bogate możliwości wizualizacji zgromadzonych danych, prezentując między innymi:

- Rozkład temperatur w kracie systemu elektroniki odczytowej.
- Wartości napięć oraz natężeń prądu dla kraty VME oraz dla poszczególnych kanałów w systemie zasilania wysokiego napięcia CAEN.
- Wskaźniki częstości zliczeń (Rate'ów) pochodzące z elektroniki FEE.
- Ogólny status operacyjny detektora oraz całego eksperymentu.

Ponadto, system umożliwia transfer zgromadzonych wyników do zewnętrznych systemów w celu dalszej analizy, co jest nieodzowne dla bieżącego monitorowania i oceny pracy detektora.



Rysunek 2.4. Ekran główny systemu SCADA



Rysunek 2.5. Ekran systemu FSM dla systemu SCADA

2.2. Domyślna architektura

Kluczowym dążeniem zespołu odpowiedzialnego za system kontroli i sterowania detektora DCS w ramach projektu FIT jest ustandaryzowanie stosowanych rozwiązań zgodnie z wytycznymi ustanowionymi przez zespół Central ALICE DCS. Dotychczasowy model architektury, mimo poprawnego działania, generował pewne komplikacje w aspekcie zarządzania całością projektu i wykazywał różnice w porównaniu do wzorcowej architektury systemu. Głównym czynnikiem przemawiającym za stosowaniem aplikacji Control Server oraz HistogramsReader w obrębie detektorów FIT była konieczność efektywnego przetwarzania znacznych ilości danych przesyłanych w krótkich, regularnych interwałach czasowych, a także potrzeba zapewnienia wydajnej komunikacji z elektroniką FEE za pośrednictwem protokołu IPbus.

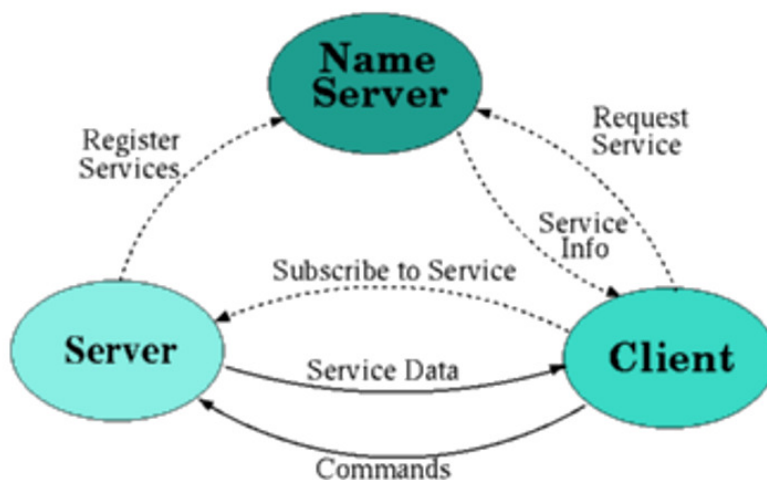
W ramach standardowej architektury systemowej, promowanej przez Central ALICE DCS, główną rolę w bezpośredniej interakcji z elektroniką detektora odgrywa system ALF (ALICE Low Level Frontend) [34]. ALF jest projektem o otwartym kodzie źródłowym, stale rozwijanym w CERN, który umożliwia komunikację z elektroniką FEE poprzez protokół GBT (Gigabit Transceiver) [35], z wykorzystaniem jednostki CRU.

Sekwencja operacyjna systemu ALF przedstawia się następująco:

- Akwizycja komend DIM: ALF przyjmuje polecenia od warstwy nadrzędnej, którą jest system FRED [34] (Front-End Device). Transfer tych poleceń może odbywać się przy użyciu jednego z dostępnych protokołów, na przykład SWT (Single Word Transaction) [34].
- Generowanie zapytań: W oparciu o otrzymane polecenia, ALF formułuje odpowiednie zapytania kierowane do jednostki CRU.
- Komunikacja z FEE: Sformułowane zapytania są umieszczane w buforze, a następnie przesyłane do elektroniki za pośrednictwem magistrali GBT.

- Odbiór odpowiedzi: Po uzyskaniu odpowiedzi od FEE, dane są zwracane tą samą ścieżką do jednostki CRU, a stamtąd trafiają z powrotem do systemu ALF, który następnie przekazuje je do systemu nadrzędnego, wykorzystując do tego mechanizmy komunikacyjne DIM.

Schemat działania systemu DIM w sieci ukazany jest na rysunku 2.6

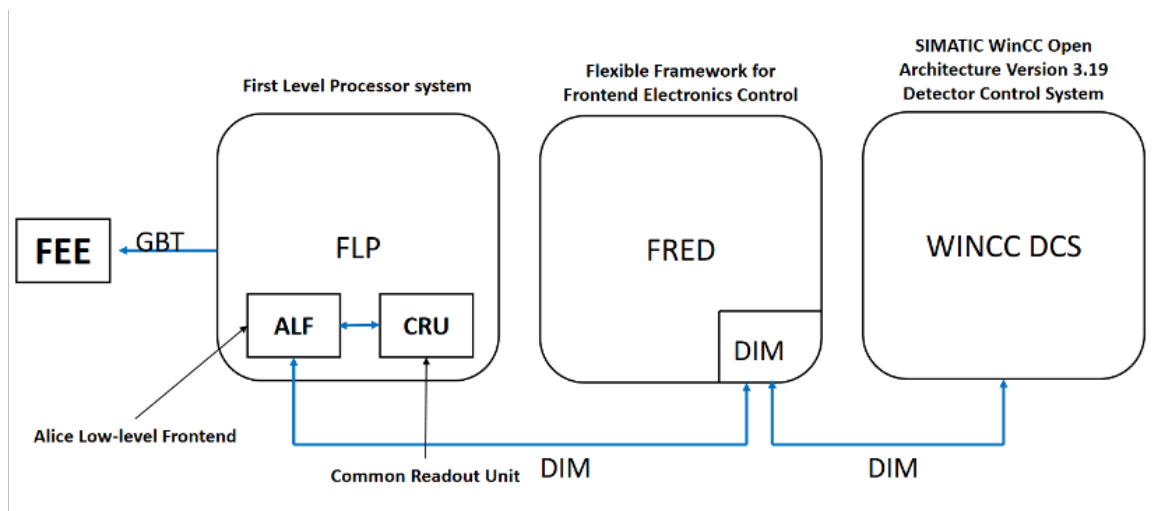


Rysunek 2.6. Schemat działania systemu DIM [31]

W hierarchii tej architektury, bezpośrednio nad systemem ALF, znajduje się oprogramowanie FRED, rozwijane i utrzymywane przez zespół Central ALICE DCS, pełniące rolę kluczowego interfejsu wymiany i przetwarzania danych pomiędzy platformą WinCC OA a systemem ALF. System FRED jest odpowiedzialny za konwersję komend, otrzymanych poprzez system DIM od warstwy SCADA, na odpowiednio sformatowane ramki danych, zgodne z protokołami obsługiwanymi przez ALF. Te ramki są następnie przesyłane do systemu ALF, który realizuje zadania opisane w poprzednim punkcie. Jest to rozwiązanie charakteryzujące się wysokim stopniem konfigurowalności, co pozwala na jego adaptację do specyficznych wymogów danego detektora. Nadrzędnym zadaniem systemu FRED jest zapewnienie płynnego i efektywnego tłumaczenia prostych, wysokopoziomowych poleceń z systemu SCADA na precyzyjne instrukcje zrozumiałe dla systemu ALF, a w konsekwencji dla elektroniki odczytowej detektora.

Na szczycie tej wielowarstwowej struktury umieszczony jest system WinCC OA, który nie tylko dostarcza interfejs użytkownika, ale również funkcjonuje jako platforma komunikacyjna, integrująca system z zewnętrznymi komponentami w eksperymencie ALICE oraz akceleratorze LHC. Umożliwia to transfer informacji pozyskanych na niższych poziomach architektury w celu ich dalszej, szczegółowej analizy. WinCC OA pozwala na zarządzanie pracą i monitorowanie stanu elektroniki poprzez systemy ALF i FRED. Dodatkowo, wykorzystując protokół OPC UA, umożliwia wymianę danych z innymi, peryferyjnymi komponentami. To rozbudowane narzędzie pozwala na realizację licznych operacji wy-

sokiego poziomu, takich jak prezentacja informacji i interakcja z operatorem, a także na komunikację z innymi kluczowymi systemami, w tym z systemem FRED (za pośrednictwem DIM) oraz z kratami zasilającymi VME i CAEN HV (poprzez OPC UA).



Rysunek 2.7. Schemat działania systemu ALFRED w celu komunikacji z FEE [36]

Opisana powyżej na rysunku 2.7, wielowarstwowa architektura, określana mianem ALFRED, stanowi kompleksowe i zintegrowane rozwiązanie systemowe dla kontroli i sterowania, które uzyskało aprobatę zespołu Central ALICE DCS. Głównym celem przyświecającym jej wdrożeniu jest zastąpienie dotychczas wykorzystywanych, dedykowanych aplikacji, przy jednoczesnym zachowaniu i zintegrowaniu pełni ich funkcjonalności w ramach środowisk WinCC OA oraz FRED.

W perspektywie docelowej, system ALFRED ma nie tylko posiadać dotychczasowe możliwości poprzednich rozwiązań, ale również wprowadzić szereg nowych funkcjonalności, których implementacja była utrudniona lub niemożliwa w ramach poprzedniej architektury systemu DCS. Jednym z istotnych usprawnień jest eliminacja problemu związanego z przerwaniem ciągłości przesyłania danych do systemów centralnych ALICE, co mogło mieć miejsce w przypadku nieoczekiwanego zamknięcia aplikacji Control Server i braku jej natychmiastowego ponownego uruchomienia. Dodatkowym ryzykiem, które występowało podczas działania aplikacji Control Server był brak wykorzystania komputera Operator Node. Tym samym każda z osób logujących się na komputer Worker Node miała możliwość edycji lub usunięcia projektu w WinCC OA. Przyjęcie nowej architektury stwarza możliwość opracowania ujednoliconej dokumentacji oraz zapewnia znacznie większą elastyczność w kontekście przyszłych modyfikacji i rozbudowy systemu.

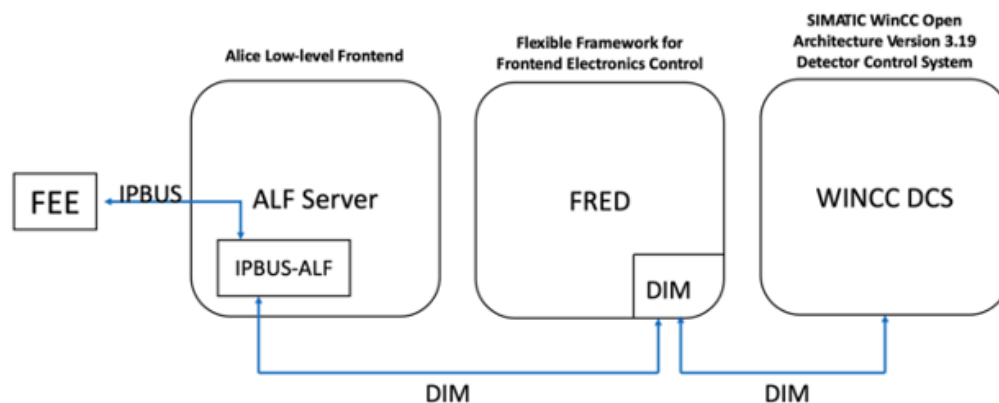
3. Cel, zakres prac i kontrybucja autora

Kluczowym zadaniem zrealizowanym w ramach niniejszej pracy jest integracja oraz wdrożenie systemu ALFRED w laboratorium detektora FIT zlokalizowanego w CERN. Podczas realizacji tego zadania napotkano na szereg wyzwań, których pochodzenie tkwiło przede wszystkim w odmienności architektury komunikacyjnej detektorów FIT w porównaniu do domyślnych założeń i wytycznych zespołu Central ALICE DCS.

Fundamentalnym wyzwaniem okazała się rozbieżność w stosowanych protokołach komunikacyjnych. Architektura systemu elektronicznego FEE poddetektorów FIT bazowała bowiem na protokole IPbus, stanowiącym podstawę dla istniejących aplikacji takich jak Control Server i HistogramReader. Standardowa koncepcja systemu ALFRED zakładała jednak interakcję z jednostką CRU poprzez protokół GBT.

Z powodu tej niekompatybilności, w zespole ALICE-FIT DCS, rozpoczęto rozwój alternatywnego rozwiązania systemowego, nazwanego ALFIPbus[37]. Koncepcja ta miała na celu odtworzenie funkcjonalności standardowego systemu ALF, jednak z tą modyfikacją, że interakcja z FEE miała być realizowana bezpośrednio z użyciem protokołu IPbus, co eliminowało potrzebę integracji jednostki CRU oraz protokołu GBT.

Kolejnym istotnym aspektem, który generował uzasadnione obawy, była potencjalna wydajność mechanizmów wymiany danych oraz ogólna stabilność całej architektury ALFRED, szczególnie w kontekście wysokiej częstotliwości akwizycji danych charakterystycznej dla detektorów FIT. Do momentu pełnej implementacji wszystkich funkcjonalności dotychczasowej aplikacji Control Server w ramach nowego systemu FRED, nie można było definitywnie potwierdzić płynności i niezawodności komunikacji w docelowej architekturze, która ukazana jest na rys. 3.1.



Rysunek 3.1. Schemat architektury systemu ALFRED dostosowanego pod FEE w detektorach ALICE-FIT [36]

Bezpośredni wkład autora pracy w integrację i rozwój systemu kontroli i sterowania detektorów ALICE-FIT w ramach niniejszej pracy obejmował następujące fazy:

1. Przygotowanie i konfiguracja serwerów: Skonfigurowano komputery przeznaczone dla systemu DCS, udostępnione przez Central ALICE DCS, które umożliwiły dalsze prace nad rozwojem projektów.
2. Ustanowienie połączenia komunikacyjnego FRED-WinCC OA: W tej fazie nawiązano komunikację pomiędzy systemem FRED a systemem WinCC OA, z wykorzystaniem mechanizmu DIM.
3. Implementacja systemu FRED oraz integracja systemu ALFIPbus: W tym etapie prac zainstalowany i przetestowany został system ALFIPbus, a także zaimplementowany został system FRED w ramach integracji systemu w oparciu o system elektroniczny detektora w laboratorium. Poprawność działania była również kontrolowana poprzez porównanie wartości z aplikacją Control Server.
4. Rozwój projektu WinCC OA: Po uzyskaniu pozytywnej weryfikacji kompletnej ścieżki komunikacyjnej (od elektroniki aż po system SCADA) oraz po skończeniu implementacji systemu FRED, skoncentrowano się na rozbudowie projektu systemu SCADA w WinCC OA. Celem było dostosowanie go do specyficznych potrzeb ekspertów, tak aby wszystkie kluczowe funkcjonalności aplikacji Control Server zostały odpowiednio przeniesione do nowej architektury ALFRED. Wymagało to zaprojektowania i implementacji licznych, nowych interfejsów użytkownika oraz dedykowanych funkcjonalności w ramach WinCC OA, co umożliwiło migrację operacji z poprzedniego systemu.

Zakres zrealizowanych prac, jak wspomniano powyżej, objął zarówno integrację oraz konfigurację kolejnych komponentów systemu ALFRED, jak i stworzenie podstaw dla jego przyszłej rozbudowy. W dalszej kolejności, główny nacisk został położony na rozwój systemu SCADA w środowisku WinCC OA, dedykowanego obsłudze systemu elektroniki odczytowej w laboratorium oraz dla detektora FV0, co zapewniło pełną funkcjonalność oraz intuicyjny interfejs użytkownika w ramach nowego środowiska. W ramach przeprowadzonych prac zrezygnowano z wykonania dokładnych pomiarów porównujących wydajność docelowej architektury systemu z systemem opartym o aplikację Control Server. Decyzja ta wynikała z faktu, iż nie był to główny cel pracy oraz wielowarstwowa architektura obu systemów wraz z wieloma czynnikami zewnętrznymi uniemożliwiłyby uzyskanie wiarygodnych i powtarzalnych wyników. Rezultatem wykonanych prac było wdrożenie oraz dostarczenie stabilnej oraz wydajnej platformy dla systemu sterowania i kontroli detektorów FIT.

Kontrybucja autora pracy w materiały pokonferencyjne:

- K. Roslon for the ALICE collaboration, [arXiv:2501.04685 [physics.ins-det]] [36].
- K. Roslon for the ALICE collaboration, Nucl. Instrum. Meth. A **1079** (2025), 170609 doi:10.1016/j.nima.2025.170609 [arXiv:2503.05518 [hep-ex]] [23].

4. Przegląd dostępnych rozwiązań

W fizyce wysokich energii oraz w przemyśle wykorzystywane są także inne systemy kontroli i sterowania. Poniżej przedstawione są te, które mogłyby być rozważone jako system DCS dla poddetektorów FIT.

4.1. EPICS (Experimental Physics and Industrial Control System)

EPICS [38] to otwartoźródłowy system sterowania i akwizycji danych, jest on popularny w eksperymentach fizyki wysokich energii oraz w przemyśle. Stanowi on alternatywę dla architektury opartej na systemie ALFRED, oferując inne podejście do zarządzania detektorami wymagającymi rozbudowanego systemu.

System EPICS wykorzystuje rozproszoną architekturę klient-serwer, gdzie główną rolę odgrywają kontrolery IOC (Input/Output Controllers) [39]. Są to autonomiczne jednostki programowalne, które odpowiadają za bezpośrednią komunikację z urządzeniami, na przykład takimi jak elektronika FEE, za pośrednictwem protokołu CA (Channel Access) [40], który łączy IOC z aplikacjami klienckimi (np. GUI (Graphical User Interface)).

W porównaniu do architektury ALFRED, EPICS jest bardziej elastyczny podczas integracji różnorodnych protokołów sprzętowych. W kontekście poddetektorów FIT, które wymagają komunikacji poprzez protokół IPbus, EPICS mógłby być bezpośrednio dostosowany do jego obsługi poprzez odpowiednią konfigurację IOC. Logika komunikacji IPbus mogłaby być scentralizowana w ramach IOC, co mogłoby uprościć implementację i zminimalizować potrzebę tworzenia rozwiązań pośredniczących, takich jak ALFIPbus, gdyż cała logika sterowania mieściłaby się w jednym komponencie.

Niemniej jednak, EPICS posiada również pewne ograniczenia, szczególnie w porównaniu z systemem WinCC OA, będącym integralną częścią architektury opartej na ALFRED. Elastyczność EPICS wymaga posiadania przez zespół projektowy bardziej zaawansowanej wiedzy technicznej na etapie konfiguracji IOC oraz dostosowywania protokołów, co może spowodować, że w początkowej fazie projektu nakład pracy będzie dużo większy. Dodatkowo, kluczowym czynnikiem przemawiającym za wyborem systemu ALFRED w eksperymencie ALICE jest fakt, że EPICS nie jest oficjalnie wspierany przez zespół Central ALICE DCS. Ma to bardzo istotne znaczenie patrząc z perspektywy standaryzacji i utrzymania systemów kontroli i sterowania w całym eksperymencie.

4.2. Tango Controls

Tango Controls [41] - jest to otwartoźródłowy system sterowania i monitorowania, który wywodzi się z Europejskiego Centrum Promieniowania Synchrotronowego w Grenoble. Jest szeroko stosowany w eksperymentach naukowych, między innymi w fizyce wysokich energii i przy akceleratorach. Stanowi on alternatywę dla architektury ALFRED, oferując inne podejście do zarządzania złożonymi systemami kontroli detektora.

Jego architektura reprezentuje każdy element fizyczny (np. elektronikę FEE) jako osobny obiekt systemowy. Komunikacja, oparta na protokole CORBA (Common Object Request Broker Architecture) [42], zapewnia wysoką wydajność wymiany danych.

W odróżnieniu od systemu ALFRED, Tango umożliwia łatwą adaptację do specyficznych dla danego urządzenia protokołów komunikacyjnych poprzez implementację tzw. Device Servers. W kontekście poddetektorów FIT, które wykorzystują protokół IPbus, odpowiednia konfiguracja Device Servers mogłaby zapewnić obsługę tej komunikacji, potencjalnie usuwając potrzebę stosowania warstw pośredniczących takich jak ALF czy FRED, które są podstawą dla architektury ALFRED.

Głównymi zaletami Tango są jego modularność oraz wsparcie międzynarodowej społeczności naukowej. Przekłada się to na wysoką dostępność dokumentacji i zasobów pomocniczych przy tworzeniu systemu. Dla projektu ALICE-FIT, wdrożenie Tango z dedykowanym Device Server'em dla protokołu IPbus mogłoby uprościć integrację z elektroniką FEE i potencjalnie skrócić czas rozwoju w porównaniu do konieczności tworzenia oprogramowania ALFIPbus w architekturze ALFRED.

Z drugiej strony, Tango posiada pewne ograniczenia. Jego implementacja wymaga większego zaangażowania w konfigurację i programowanie Device Servers, co może zwiększyć nakład pracy na początku tworzenia projektu, szczególnie w zespole o ograniczonym doświadczeniu z tym systemem. Tango nie ma także kompatybilności z systemami eksperymentu ALICE.

4.3. ALFRED

System ALFRED, jako standardowe rozwiązanie w systemie kontroli DCS, wspierane przez zespół Central ALICE DCS, pełni rolę podstawowego oprogramowania dla sterowania i nadzoru detektorów w eksperymencie ALICE. Jego konstrukcja bazuje na architekturze, składającej się z komponentów ALF oraz FRED, które utrzymują wzajemną komunikację poprzez system DIM.

Komponent ALF, rozwijany bezpośrednio w CERN, został stworzony z myślą o wymianie danych z jednostką CRU za pośrednictwem protokołu GBT. Protokół GBT jest przyjętym standardem implementowanym dla systemu elektroniki odczytowej w dużej większości detektorów ALICE, z wyłączeniem detektorów FIT, które opierają swoją konfigurację na protokole IPbus. Główną funkcjonalnością ALF jest konwersja ramki danych (np. typu SWT), otrzymanej z warstwy FRED poprzez DIM, na format kompatybilny z protokołem GBT, a następnie przesłanie polecenia do CRU. W kolejnym etapie zadaniem ALF jest przetworzenie odpowiedzi zwrotnej tą samą drogą do systemu FRED.

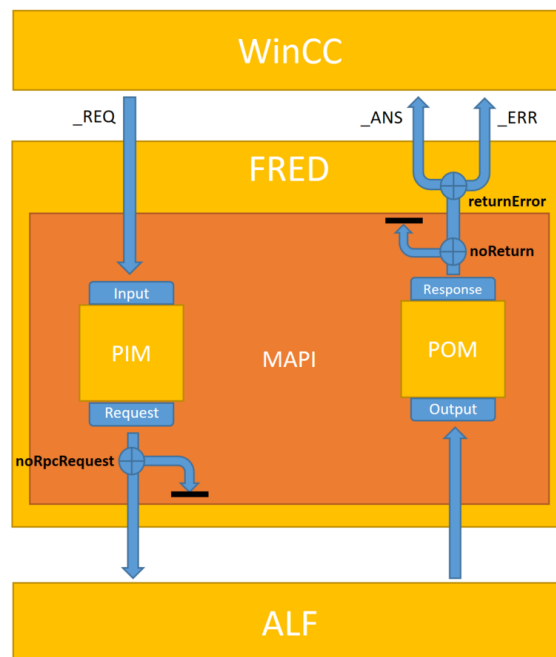
FRED natomiast stanowi znacznie bardziej zaawansowany komponent w systemie kontroli, jest on stale rozwijany w ramach eksperymentu ALICE. Pozwala między innymi na interakcję z systemem DIM, integrację z bazą danych Oracle oraz konwersję poleceń

przychodzących z wyższych warstw systemu na ramki danych w formatach takich jak SWT, SCA (Slow Control Adapter) [43] czy IC (Internal Communication) [34].

Fundamentalnym atutem systemu FRED jest biblioteka MAPI (Message Application Interface) [34], umożliwiająca elastyczne tworzenie kodu oprogramowania dla unikalnych potrzeb poszczególnych detektorów. Biblioteka ta dostarcza zestaw wielu różnych i funkcjonalnych klas, w tym Standard MAPI, Iterative MAPI, Indefinite MAPI oraz MAPI Group. Pozwala to na indywidualną implementację serwisów DIM przeznaczonych do realizacji bardziej skomplikowanych zadań operacyjnych.

Dodatkowo, dla każdego zdefiniowanego serwisu DIM, FRED automatycznie generuje trzy podserwisy, których nazwa bazuje na tekście podanym w definicji parametru, są one rozróżniane sufiksami:

- `_ANS`: dedykowany prezentacji odpowiedzi systemu na wykonane polecenia.
- `_ERR`: przeznaczony do sygnalizowania ewentualnych błędów zaistniałych podczas wykonywania operacji.
- `_REQ`: odpowiedzialny za odbieranie i nasłuchiwanie komend inicjowanych z systemu WinCC OA.



Rysunek 4.1. Schemat wymiany danych z wykorzystaniem interfejsu MAPI [34]

Dzięki wymienionym mechanizmom, FRED zapewnia kompleksową obsługę komunikacji z systemem elektroniki odczytowej detektora, obejmującą zarówno wysyłanie poleceń, jak i ich przetwarzanie w oparciu o dane otrzymywane z warstw nadrzędnych za pośrednictwem systemu DIM.

Warto zaznaczyć, że system ALFRED nie dysponuje własną dedykowaną aplikacją pozwalającą na bezpośrednią interakcję z użytkownikiem. Z tego względu, dla osiągnię-

cia pełnej funkcjonalności operacyjnej, niezbędna jest jego integracja z systemem klasy SCADA, realizowana w środowisku WinCC OA.

4.4. SIMATIC WinCC OA (Open Architecture)

Nawiązując do wcześniejszych informacji, w architekturze systemu kontroli eksperymentu ALICE, nadrzędną rolę zajmuje system klasy SCADA, który pełni funkcję głównego punktu dla zarządzania, wizualizacji i akwizycji danych dla całego detektora. Jego kluczowym zadaniem jest sprawna komunikacja z komponentami na niższych warstwach, takimi jak oprogramowanie FRED, co tutaj realizowane jest poprzez system DIM, zapewniający sprawdzoną wymianę informacji.

W środowisku CERN, jako standardowa platforma SCADA, wykorzystywane jest oprogramowanie SIMATIC WinCC OA [32], produkt dziś należący do firmy Siemens. Jednakże, specyfika wielkich eksperymentów fizycznych, takich jak ALICE, wymusza ciągłe rozszerzanie i dostosowywanie narzędzi. Dlatego też wersja WinCC OA stosowana w CERN jest wzbogacana o dodatkowe biblioteki i rozszerzenia, opracowywane przez wewnętrzne zespoły ekspertów. Biblioteki te mają na celu nie tylko dostosowanie systemu do unikalnych wymagań eksperymentu, ale również implementację zaawansowanych funkcjonalności, niezbędnych do precyzyjnego sterowania, monitorowania i diagnostyki aparatur badawczych w czasie rzeczywistym.

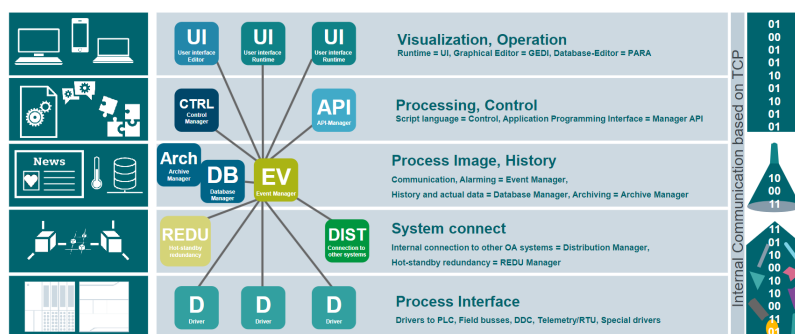
Platforma WinCC OA w CERN jest w szczególności rozszerzona o bibliotekę JCOP (Joint Controls Project) [44]. Jest to narzędzie, które w istotnym stopniu poszerza możliwości standardowej wersji systemu WinCC OA, dostarczając zestaw sprecyzowanych rozwiązań i interfejsów dla potrzeb eksperymentów fizyki wysokich energii. Biblioteka JCOP w tym kontekście zapewnia między innymi:

- Zapewnienie niezawodnej i efektywnej komunikacji z wykorzystaniem protokołu DIM: jest to bardzo istotne dla płynnego przepływu informacji i komend pomiędzy WinCC OA a systemami niższych warstw, takimi jak FRED, odpowiedzialnymi za bezpośrednią interakcję z systemem elektroniki odczytowej detektora.
- Integrację z centralnymi systemami zarządzania eksperymentem ALICE poprzez mechanizm FSM: pozwala to na precyzyjne i skoordynowane zarządzanie stanami operacyjnymi detektora w kontekście pracy całego eksperymentu, zapewniając spójność działania z innymi podsystemami ALICE.
- Wsparcie dla implementacji krytycznych systemów bezpieczeństwa detektora (DSS – Detector Safety System): dostarcza to niezbędne informacje dotyczące stanu i ochrony aparatury detektora oraz pozwala na przeprowadzanie specjalnych operacji w eksperymencie ALICE.

WinCC OA, dzięki swojej elastycznej architekturze, oferuje także rozszerzone możliwości bezpośredniej komunikacji z szeroką gamą urządzeń przemysłowych firmy Siemens, na przykład ze sterownikami programowalnymi PLC serii S7-1200, wykorzystując do tego celu

dedykowane protokoły komunikacyjne. Ponadto, system ten wspiera uniwersalny protokół OPC UA. Jest to otwarty standard komunikacyjny, kluczowy dla zapewnienia komunikacji z urządzeniami różnych producentów, dzięki czemu pozwala na bezproblemową integrację z komponentami detektorów.

Dzięki połączeniu tych wszystkich funkcji, system SCADA zrealizowany na platformie WinCC OA zapewnia pełną i niezawodną integrację z systemem ALFRED. Umożliwia to nie tylko efektywną wymianę danych w czasie rzeczywistym, ale także kompleksowe sterowanie wszystkimi aspektami pracy detektora w ramach eksperymentu ALICE. Tym samym, WinCC OA stanowi centralny pulpit operacyjny, z którego operatorzy mają możliwość nie tylko monitorowania bieżącego stanu i kluczowych parametrów wszystkich podsystemów detektora, ale również aktywnego zarządzania jego konfiguracją, inicjowania procedur kalibracyjnych oraz podejmowania szybkich interwencji w przypadku wystąpienia nieprawidłowości, co jest niezbędne dla zapewnienia ciągłości i wysokiej jakości zbieranych danych fizycznych. Rozdzielenie procesów systemu na podstawowe Manager'y ukazane jest na Rysunku 4.2.



Rysunek 4.2. Struktura systemu WinCC OA podzielonej na kolejne jednostki i managerów [32]

5. Rozwój i wdrożenie systemu ALFRED

5.1. Wprowadzone rozwiązania

Aby umożliwić implementację systemu ALFRED w laboratorium ALICE-FIT należało rozpocząć pracę od uruchomienia podstawowych komponentów wyżej wymienionej architektury systemu, obejmujących serwer DNS (Domain Name Server) dla systemu DIM, oprogramowanie FRED, a także ustanowienie pełnej komunikacji z systemem ALFIPbus oraz WinCC OA. W celu wykonania tych prac należało zastosować podejście, które pozwalało zintegrować konfigurację finalnej infrastruktury sprzętowej, rozwój oprogramowania FRED oraz na końcu wykonanie testów w celu weryfikacji poprawności działania wszystkich komponentów.

Implementacja została rozpoczęta od instalacji systemu operacyjnego CentOS 7 [45] na dwóch serwerach, które zostały przeznaczone i przygotowane w laboratorium ALICE-FIT. Należało zainstalować tę konkretną dystrybucję Linuxa, która była oficjalnie wspierana przez CERN, w celu zapewnienia dostępu do pakietu `devtoolset-7`, niezbędnego do kompilacji oprogramowania FRED, który wymaga specyficznych wersji kompilatorów `g++` i `gcc` oraz wielu innych dodatkowych zależności. Dostępne były dwa serwery, jeden z nich przeznaczony został dla systemu FRED oraz serwera DIM, a drugi z nich zadedykowany dla systemu ALFIPbus i połączenia z FEE w laboratorium, w tym celu należało skonfigurować odpowiednie ustawienia karty sieciowej. Po pomyślnym ustawieniu obu urządzeń, przystąpiono do kompilacji i uruchomienia biblioteki DIM, a następnie oprogramowania FRED. Gdy rozpoczęto pracę, zaraz po aktywacji systemu FRED, można było zaobserwować brak aktywnej warstwy ALF (co było oczekiwanym błędem). Z tego powodu komunikację w obrębie sieci wewnętrznej, na serwerze DIM, przetestowano przy wykorzystaniu aplikacji WebDID, która jest rozszerzeniem systemu DIM. Udało się, w trakcie tych testów, zidentyfikować problemy z wymianą poleceń pomiędzy warstwami architektury, które po konsultacji z ekspertem, można było rozwiązać poprzez otwarcie portów 5100–5110 oraz 2505, było one wcześniej blokowane przez konfigurację zapory sieciowej. Poniżej na rysunku 5.1 znajduje się wycinek log'ów systemu FRED informujący o jego poprawnym uruchomieniu.

```
[imermer@server FRED_MAPI]$ ./bin/FREDServer
[INFO] [2024:08:19 12:39:44.819448] FRED launched!
[INFO] [2024:08:19 12:39:44.825811] Parsing started.
[INFO] [2024:08:19 12:39:45.004449] Parsing Completed. Starting FRED.
[INFO] [2024:08:19 12:39:45.073350] FRED running.
[INFO] [2024:08:19 12:39:45.079572] Server LAB started!
```

Rysunek 5.1. Zapis log'ów po uruchomieniu systemu FRED w laboratorium

Weryfikując dalej zapisy w logach systemu FRED, potwierdzono, że komendy wysyłane z aplikacji WebDID są poprawnie odbierane, oraz że dane, które publikowane są na serwisach DIM odpowiadają oczekiwaniom. Można było przystąpić wtedy do etapu integracji systemu FRED z systemem SCADA w WinCC OA. W tym celu zaprojektowany został

prosty wygląd interfejsu użytkownika w WinCC OA, odzwierciedlający kalkulator i umożliwiający wysyłanie czterech podstawowych operacji matematycznych wraz z dwiema liczbami. W systemie FRED dodany został w tym celu dedykowany serwis MAPI, który był odpowiedzialny za przetwarzanie tych poleceń, realizację operacji arytmetycznych oraz publikowanie wyników na odpowiednim serwisie DIM. Gdy udało się pomyślnie zaimplementować i przetestować cały mechanizm z wykorzystaniem środowiska WebDID, sprawdzona została komunikacja z systemem WinCC OA przy tej początkowej i podstawowej konfiguracji. Pozwoliło to jednoznacznie potwierdzić jej prawidłowe działanie. W tym samym czasie, równolegle, zespół ALICE-FIT DCS, którego autor jest członkiem, rozwijał system ALFIPbus.

Następnym bardzo ważnym etapem prac było zaadoptowanie systemu FRED, tak aby jego implementacja była kompatybilna z elektroniką wykorzystywaną w laboratorium ALICE-FIT. Należało rozpocząć ten proces od szczegółowego przeanalizowania kodu źródłowego aplikacji Control Server, w celu zidentyfikowania, które elementy logiki powinny zostać przeniesione do warstwy FRED, a które do systemu SCADA. Przyjęty został podział odpowiedzialności, w którym: system FRED miał być odpowiedzialny za realizację złożonych operacji, przetwarzanie danych i ich przesyłanie do WinCC OA, natomiast system SCADA miał być wykorzystywany głównie do wizualizacji informacji odbieranych z serwera DIM, komunikacji z systemami centralnymi w eksperymencie oraz zarządzaniu kratami Wiener i CAEN przy użyciu serwera OPC UA.

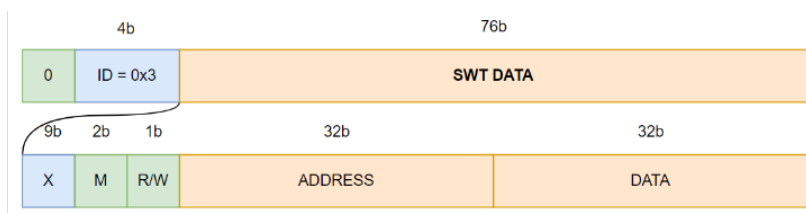
Ostatnim, ale równie bardzo istotnym krokiem w przeprowadzonych pracach integracyjnych systemu było skomunikowanie systemu FRED z systemem ALFIPbus, co umożliwiło weryfikację całej ścieżki komunikacyjnej z elektroniką w laboratorium ALICE-FIT. W tym celu został wykorzystany drugi z serwerów z systemem CentOS 7, na którym zainstalowano pakiety devtoolset-11 oraz cmake3, a następnie skompilowano aplikację ALFIPbus. Równocześnie skonfigurowana została sieć wewnętrzna dla protokołu IPbus. Po uruchomieniu oprogramowania ALFIPbus z przypisanym adresem IP dla płyty TCM, pokazana została odpowiedź w log'ach, która potwierdziła pomyślne nawiązanie połączenia z serwerem DIM oraz prawidłową transmisję testowego pakietu danych. Pozwoliło to na dalszy rozwój systemu FRED, w tym do wdrożenia serwisów DIM, które dały możliwość wysyłania i odbierania danych z rejestrów elektroniki.

Ustanowienie pełnej komunikacji pomiędzy systemami FRED i ALFIPbus wymagało przeprowadzenia kilkunastu spotkań roboczych oraz przeanalizowania zalet i ograniczeń dla dostępnych rozwiązań. Jednym z najbardziej kluczowych zagadnień był wybór formatu ramki danych, która przesyłana była za pośrednictwem serwera DIM. W ramach systemu FRED możliwe do implementacji są trzy opcje protokołów, tak jak wspomniano wcześniej: SWT, SCA oraz IC. Ostatecznie, po kilkunastu dyskusjach, podjęta została decyzja o wykorzystaniu protokołu SWT, biorąc pod uwagę jego elastyczność oraz możliwość dostosowania do wymagań komunikacyjnych. Aby skutecznie przenieść funkcjonalno-

ści aplikacji Control Server, która operuje w oparciu o protokół IPbus, zaproponowano poniższy format ramki:

- 3 bity przeznaczone na określenie typu operacji (np. odczyt, zapis).
- 32 bity dedykowane na adres docelowego rejestru w elektronice.
- 32 bity na dane przesyłane do elektroniki (w przypadku operacji zapisu).

Ta zdefiniowana struktura, zilustrowana na Rysunku 5.2, zapewniła efektywną wymianę informacji pomiędzy systemami FRED i ALFIPbus w środowisku laboratoryjnym ALICE-FIT.



Rysunek 5.2. Struktura ramki SWT w celu wymiany danych pomiędzy FRED i ALFIPbus w laboratorium ALICE-FIT

Gdy ostateczny format wymiany danych został ustalony, można było przystąpić do implementacji obsługi tych danych w ramach serwisu MAPI w systemie FRED. Pierwszym krokiem stało się zaprojektowanie i stworzenie ramki dla operacji odczytu danych w obrębie wcześniej utworzonego serwisu MAPI. W tym przypadku ramka składała się z 19 zer: typ operacji został ustawiony na zero (co oznaczało odczyt z elektroniki), adres rejestru określono jako zero, natomiast pole danych pozostało niewykorzystane, ponieważ system ALFIPbus pomija je podczas realizacji operacji odczytu, dlatego w celu przeprowadzenia powyższego testu, można było to pominąć.

Po otrzymaniu odpowiedzi z elektroniki i po zweryfikowaniu jej poprawności na bazie log'ów systemowych, przystąpiono do etapu sprawdzania przeliczeń wartości z elektroniki na odpowiadające im wielkości fizyczne. Można było przeprowadzić ten proces, wykorzystując równolegle system SCADA w środowisku WinCC OA, który komunikował się z serwerem DIM, oraz aplikację Control Server, co umożliwiała bezpośrednie porównanie wyświetlanych danych w obu programach. Gdy potwierdzono zgodność wartości, które były prezentowane w obu systemach, rozpoczęto testy operacji zapisu danych do rejestru FEE. Operacja ta również zakończyła się sukcesem, jednak podczas analizy aplikacji kodu Control Server pojawiły się dodatkowe funkcjonalności, które należało zaimplementować. Dwa rejestry odpowiadały za regulację fazy zegara dla części A oraz części C elektroniki w detektorze. Wymagało to wprowadzenia krótkiej przerwy w działaniu systemu po każdej zmianie tych wartości, aby zapewnić stabilność pracy systemu elektroniki odczytowej. W kolejnych etapach mechanizmy zabezpieczające tego typu zostały dodane, jednak wymagały one dokładnego sprawdzenia podczas operacji zapisu. Poniżej znajduje się informacja publikowana na jednym z serwisów DIM, rys. 5.3.

```
Service LAB/PM/PMA0/1VPOWER_ANS (C) Contents :
Timestamp: Mon Aug 19 14:15:26.587 2024    Quality: 0    Size: 9 bytes
C    0H: 30 2E 39 36 37 33 39 32 00        '0.967392.'
```

Rysunek 5.3. Podgląd danych w narzędziu WebDID dla jednego z rejestrów

Gdy testy, które jednoznacznie potwierdziły poprawność zarówno operacji odczytu, jak i zapisu poleceń, zostały zakończone pomyślnie, możliwe stało się przejście do dalszych etapów rozwoju systemu FRED. Kolejną rzeczą do dodania stało się zaimplementowanie dodatkowych serwisów DIM dla wszystkich pozostałych rejestrów elektroniki.

W tym celu, dla każdego rejestru na płytach elektroniki utworzono odrębny serwis DIM – łącznie po 133x3 serwisów dla płyty TCM oraz po 275x3 serwisów dla każdej podłączonej płyty PM. Poniżej znajduje się wycinek kodu odpowiedzialnego za dodawanie serwisów MAPI z początku rozwoju systemu, jest on ukazany na rys. 5.4

```
services.push_back(serviceName+"MODE_SETTINGS");
services.push_back(serviceName+"TRIGGER_RESPOND_MASK");
services.push_back(serviceName+"DATA_BUNCH_PATTERN");
services.push_back(serviceName+"TRIGGER_SINGLE_VALUE");
services.push_back(serviceName+"TRIGGER_CONT_PATTERN_MSB");
services.push_back(serviceName+"TRIGGER_CONT_PATTERN_LSB");
services.push_back(serviceName+"TRIGGER_CONT_VALUE");
services.push_back(serviceName+"GENERATORS_BUNCH_FREQ");
services.push_back(serviceName+"GENERATORS_FREQ_OFFSET");
services.push_back(serviceName+"RDH_FIELDS1");
services.push_back(serviceName+"RDH_FIELDS2");
services.push_back(serviceName+"DELAYS");
services.push_back(serviceName+"DATA_SELECT_TRG_MASK");
services.push_back(serviceName+"MODES_STATUS");
services.push_back(serviceName+"CRU_BC");
services.push_back(serviceName+"CRU_ORBIT");
services.push_back(serviceName+"FIFO_COUNT");
services.push_back(serviceName+"SEL_FIRST_HIT_DROPPED_ORBIT");
services.push_back(serviceName+"SEL_LAST_HIT_DROPPED_ORBIT");
services.push_back(serviceName+"SEL_HITS_DROPPED");
services.push_back(serviceName+"READOUT_RATE");
services.push_back(serviceName+"CURRENT_ADDRESS");
services.push_back(serviceName+"HISTOGRAM_DATA_READOUT");
services.push_back(serviceName+"ATX_TIMESTAMP");
services.push_back(serviceName+"FW_UPGRADE_COMM");
services.push_back(serviceName+"FW_UPGRADE_DATA");
services.push_back(serviceName+"FW_UPGRADE_END");
services.push_back(serviceName+"FW_UPGRADE_STATUS");
services.push_back(serviceName+"FPGA_TEMPERATURE");
services.push_back(serviceName+"1VPOWER");
services.push_back(serviceName+"18VPOWER");
services.push_back(serviceName+"FPGA_TIMESTAMP");
```

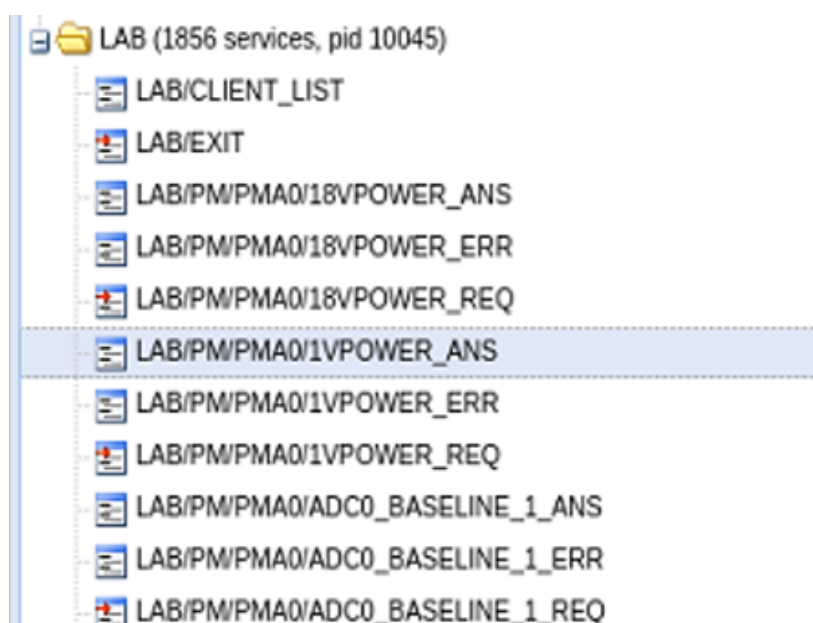
Rysunek 5.4. Ręczna deklaracja serwisów na początku rozwoju systemu

Były to serwisy aktualizowane z częstotliwością jednego Hz, z powodu odczytu bieżących wartości rejestrów, dokonywania ich przeliczeń na wielkości fizyczne i przekazywania ich do systemu SCADA, tak jak zostało to opisane wyżej. Na rys. 5.5 znajduje się część serwisów DIM utworzonych podczas rozwoju systemu FRED.

W tym celu również należało przeprowadzić szczegółową analizę kodu źródłowego aplikacji Control Server w celu opracowania odpowiednich funkcji przeliczeniowych oraz wprowadzenia ograniczeń na pola bitowe i dopuszczalne wartości liczbowe. Wprowadzenie tych ograniczeń miało kluczowe znaczenie dla zabezpieczenia rejestrów przeznaczonego

czonych wyłącznie do odczytu przed przypadkowym nadpisaniem, a także dla ochrony elektroniki przed wprowadzeniem wartości mogących spowodować jej nieprawidłowe działanie lub uszkodzenie.

Podczas implementacji przetestowano również zaawansowane operacje zapisu, takie jak ReadWriteModify, które umożliwiały modyfikację wybranych bitów rejestru za pomocą operacji logicznych OR oraz AND. Dzięki wykorzystaniu tego rozwiązania możliwe stało się zapisywanie pojedynczych parametrów elektroniki z poziomu systemu SCADA w WinCC OA, minimalizując jednocześnie ryzyko błędów wynikających z konieczności manualnego konstruowania wartości w ramce dla całego rejestru.

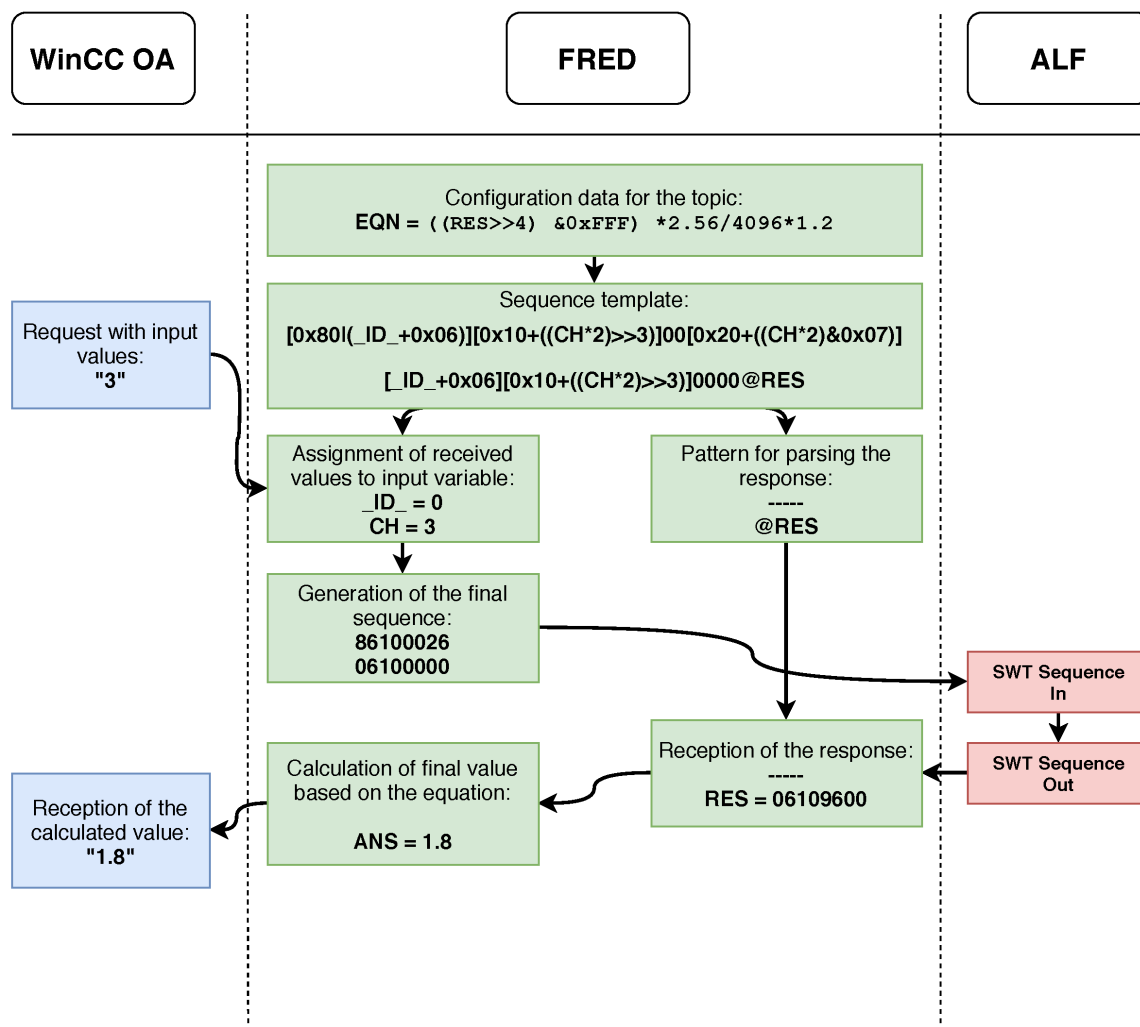


Rysunek 5.5. Część serwisów oraz komend dla płyty PMA0

Schemat blokowy komunikacji, obejmujący interakcje między systemami SCADA, FRED oraz ALF z wykorzystaniem serwera DIM, został szczegółowo przedstawiony na rysunku 5.6.

Gdy wyżej opisany etap implementacji odczytu i zapisu parametrów dla aplikacji Control Server się zakończył, możliwe stało się rozpoczęcie prac nad operacjami wywoływanymi na żądanie użytkownika lub w odpowiedzi na określone zdarzenia systemowe. System FRED został wyposażony w serwisy MAPI takie jak INITIALIZE oraz RESET_SYSTEM.

Pierwsza z tych operacji była odpowiedzialna za wgrywanie domyślnej konfiguracji oraz inicjalizację maski podłączonych płyt PM i ustawianie parametru (PM_MASK_SPI) podczas procedury uruchamiania systemu. Brak poprawnie zainicjalizowanej maski uniemożliwia odczyt danych z płyt PM, ponieważ płyta TCM nie jest w stanie rozpoznać podłączonych modułów, nawet jeśli działają one w pełni prawidłowo. Była to niezbędna funkcja pozwalająca na zapewnienia pełnej funkcjonalności systemu. Druga z powyższych operacji, polegała na wysłaniu odpowiedniej komendy resetu do wybranych bitów



Rysunek 5.6. Schemat komunikacji pomiędzy systemami WinCC OA, FRED oraz ALF z wykorzystaniem serwera DIM [34]

w rejestrach elektroniki, a następnie na oczekiwaniu na zakończenie procesu resetowania poszczególnych podsystemów. Dodatkowo, zaimplementowano również funkcje takie jak RESET_GBT, RESET_ERRORS oraz RESET_STATISTICS. Wszystkie te procedury były szczegółowo opisane w kodzie źródłowym aplikacji Control Server, dzięki czemu mogły zostać przeniesione do systemu FRED, z zachowaniem ich pierwotnej logiki działania.

Po wdrożeniu powyższych funkcjonalności, system FRED posiadał zdolność zarówno do pobierania danych, jak i ustawiania parametrów systemu elektroniki odczytowej, oferując jednocześnie zestaw procedur uruchamianych na żądanie eksperta oraz automatycznie przez system nadrzędny taki jak WinCC OA. Następnie prace skoncentrowały się na jednej z kluczowych funkcji systemu, mechanizmie pobierania danych z liczników, nazywanych Counter'ami. Na podstawie tych surowych danych obliczane były wartości tzw. Rate'ów, wyrażane w jednostce Hz. Wskazywały one częstotliwość pojawiania się zdarzeń w danym kanale w jednostce czasu (ustawianej w rejestrze elektroniki). Dane te, ze względu na ich

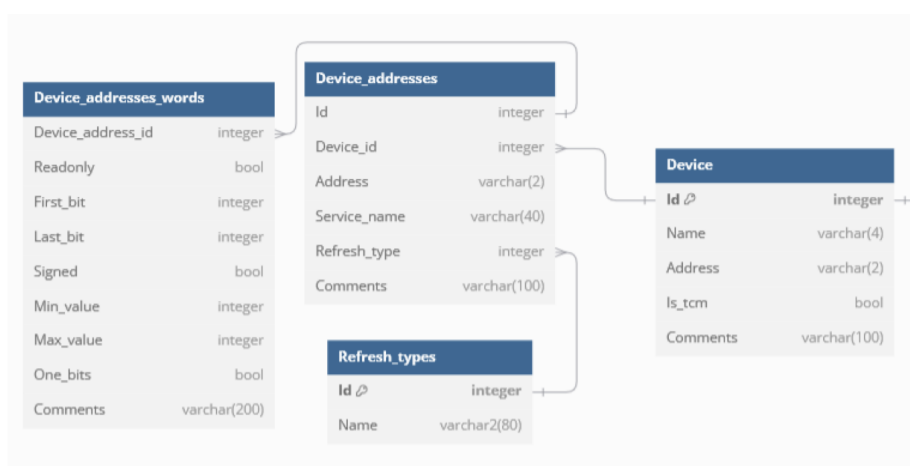
znaczenie dla analiz fizycznych prowadzonych w eksperymencie ALICE, wymagały dokładnego przeliczenia. Analiza kodu aplikacji Control Server oraz konsultacje z ekspertami od detektora wykazały, że dane z Counterów są zapisywane w rejestrach typu FIFO. W związku z tym, proces odczytu tych danych wymagał ciągłego odpytывania odpowiednich rejestrów, dopóki zawierały one dane, oraz analizy tych rekordów, które były aktualne i nieprzetworzone. Liczniki aktualizowały swoją zawartość w stałych, zdefiniowanych interwałach czasowych, które mogły być konfigurowane przez użytkownika za pomocą wcześniej zaimplementowanej funkcji zapisu do odpowiedniego rejestru. Po pobraniu danych, były one publikowane jako wartości Counter'ów w systemie SCADA, a odpowiadające im Rate'y były obliczane poprzez podzielenie wartości liczników przez ustalony interwał czasowy, wyrażony w sekundach.

Następną także ważną funkcjonalnością, która została zaimplementowana jako serwis MAPI w systemie FRED, było odczytywanie danych z histogramów korzystając z odpowiednich rejestrów elektroniki. Dla aplikacji HistogramReader jest to główna funkcja, którą oferuje. Pozwala ona na precyzyjne określenie liczby cząstek zarejestrowanych w poszczególnych binach dla każdego kanału na płytach PM. Ze względu na bardzo dużą ilość gromadzonych danych, odczyt tych informacji jest szczególnie wymagający pod względem wydajnościowym. Dane są zapisywane w rejestrach FIFO i dzielą się na pięć odrębnych kategorii dla każdego kanału: biny od 0 do 4095 dla przetwornika ADC0, biny od 0 do 4095 dla przetwornika ADC1, biny dla informacji czasowych z detektora (zakładka TIME) oraz biny o wartościach ujemnych od -255 do -1, również osobno dla ADC0 i ADC1. Zaimplementowanie mechanizmu odczytu i publikacji tych wszystkich danych w ramach jednego, zbiorczego serwisu DIM, z zachowaniem podziału na poszczególne kategorie, umożliwiło przeniesienie istotnej części funkcjonalności aplikacji HistogramReader bezpośrednio do warstwy systemu FRED, co pozwoliło systemowi SCADA na ich wizualizację oraz prezentację użytkownikowi.

Gdy system FRED był rozwijany w wielu momentach napotymano różnorodne wyzwania, które wymagały przeprowadzenia analizy oraz implementacji odpowiednich mechanizmów optymalizacyjnych. Jednym z odkrytych problemów było odświeżanie dużej liczby serwisów DIM co sekundę. Początkowo do tego celu została wykorzystana funkcjonalność MAPI Group, co jednak w praktyce powodowało znaczne opóźnienia w komunikacji, dla przykładu, dla konfiguracji z dwiema płytami PM w laboratorium generowano ponad 650 pojedynczych serwisów MAPI na sekundę. Po konsultacjach z zespołem Central ALICE DCS, wprowadzono rozwiązanie oparte na serwisie Indefinite MAPI, które umożliwiło pobranie wartości dla wszystkich rejestrów elektroniki w ramach jednej paczki komend. Wymagało to również, aby zespół Central ALICE DCS dokonał odpowiednich modyfikacji w systemie FRED, dostosowując tę funkcjonalność do specyficznych potrzeb detektorów FIT. Jednakże, w dalszym etapie rozwoju, przywrócono wykorzystanie serwisu MAPI Group, lecz w zmodyfikowanej formie. Dane z elektroniki były zapisywane lokalnie w pamięci

operacyjnej systemu FRED, a pojedyncze serwisy MAPI jedynie publikowały te wcześniej zbuforowane odpowiedzi, bez konieczności bezpośredniej komunikacji z elektroniką przy każdym zapytaniu. W tym celu wysyłano wewnętrzne polecenia READ, które zwracały dane pobrane w dużej paczce, a właściwa komunikacja z elektroniką za pośrednictwem systemu ALFIPbus była nawiązywana tylko w razie rzeczywistej potrzeby, gdy takie polecenie było wysyłane z systemu SCADA.

Kolejną istotną zmianą było przeniesienie konfiguracji dla poszczególnych serwisów DIM z kodu źródłowego bezpośrednio do bazy danych Oracle, która została dostarczona i była zarządzana przez zespół Central ALICE DCS. Krok ten umożliwił centralizację kluczowych informacji dotyczących rejestrów elektroniki, ich dopuszczalnych zakresów wartości oraz wzorów przeliczeniowych, służących do konwersji wartości elektronicznych na wielkości fizyczne i odwrotnie. System FRED pozwalał na domyślne nawiązanie połączenia z bazą danych przy użyciu zdefiniowanych parametrów, takich jak *connection string*, nazwa użytkownika i hasło, co pozwalało na dynamiczne odczytywanie parametrów konfiguracyjnych podczas inicjalizacji systemu i przekazywanie ich do odpowiednich obiektów w aplikacji. Dzięki temu osiągnięto ujednolichenie parametrów oraz znaczące zwiększenie przejrzystości procesu walidacji danych. Do realizacji przeliczeń wartości wykorzystano bibliotekę *Equation* [34], która jest dostępna w ramach systemu FRED. W bazie danych zapisano także szczegółowe informacje o podłączonych płytach PM, ich adresach bazowych na płycie TCM oraz pełne konfiguracje elektroniki, które wcześniej przechowywane były w plikach *.cfg* używanych przez aplikację Control Server. Pliki te, zapisane w formacie *.ini* (struktura <klucz>, <wartość>), zawierały klucze odpowiadające poszczególnym rejestrom elektroniki oraz ich wartości zapisane w formacie szesnastkowym. Było to nieczytelne dla użytkownika. Schemat bazy danych, przeznaczony do przechowywania informacji o parametrach elektroniki znajduje się na rys. 5.7.



Rysunek 5.7. Schemat bazy danych dla parametrów elektroniki

Wprowadzono także standaryzację dodatkowych funkcji, aby podatność kodu na błędy przy implementacji przyszłych modyfikacji była jak najmniejsza. W tym celu stworzone

zostały oddzielne, dedykowane procedury do generowania ramek SWT na potrzeby komunikacji z systemem ALF oraz do obsługi zapytań przychodzących od systemu SCADA, działającego w środowisku WinCC OA.

5.2. Integracja systemu w laboratorium ALICE-FIT

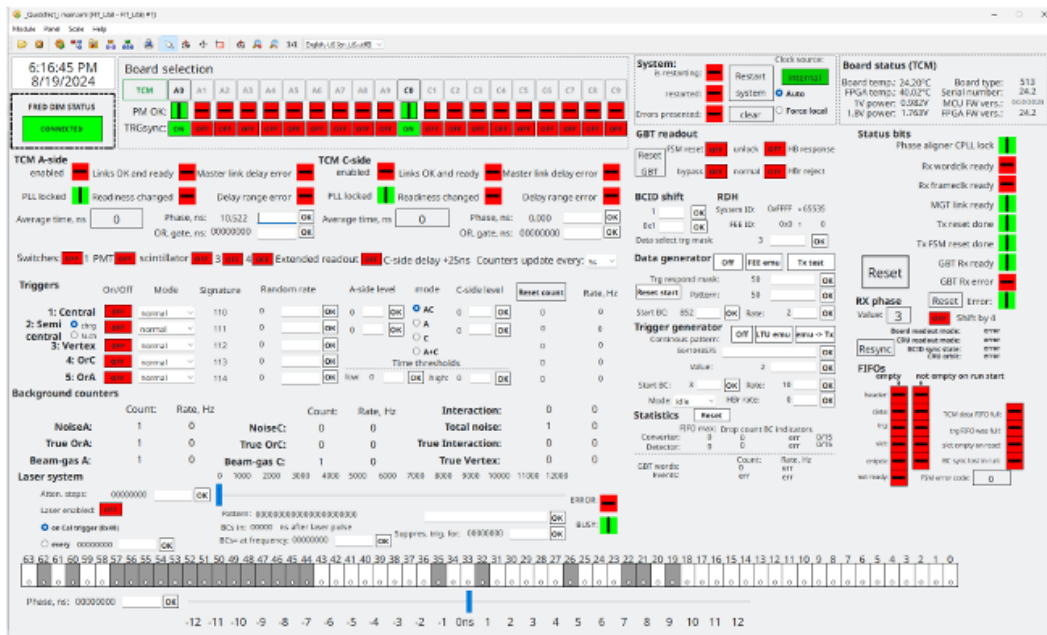
Podczas integracji systemu ALFRED w laboratorium ALICE-FIT należało wykonać kilkanaście pośrednich operacji. Należało między innymi skonfigurować oraz aktywować jego poszczególne komponenty, co miało na celu zagwarantowanie komunikacji z elektroniką detektora. Fundamentalne znaczenie miało dostosowanie struktury bazy danych oraz inicjalizacja wszystkich warstw tego systemu.

Podstawowa faza prac obejmowała dokładną konfigurację bazy danych, tak aby jej schemat odzwierciedlał specyfikę laboratoryjną, biorąc pod uwagę między innymi fizycznie podłączone płyty TCM, PMA0 oraz PMC0. Po wprowadzeniu i zweryfikowaniu niezbędnych rekordów w tabelach bazy danych, na komputerze dla systemu FRED zweryfikowano działanie serwera DIM DNS. W tym samym czasie, na drugim serwerze przewidzianym dla systemu ALFIPbus, zainstalowana i skonfigurowana została jego najnowsza dostępna wersja. Przed wykonaniem tych kroków należało odpowiednio skonfigurować sieć w ramach infrastruktury CERN, w celu umożliwienie komunikacji z serwerem DIM, oraz ustawieniem parametrów karty sieciowej dla protokołu IPbus, warunkującym poprawne połączenie z płytą TCM wykorzystywaną w laboratorium.

System ALFIPbus został uruchomiony ze zdefiniowanymi parametrami startowymi. Gdy w terminalu zostało wyświetlone potwierdzenie pomyślnego nawiązania komunikacji z elektroniką laboratoryjną, przystąpiono do kolejnego etapu, jakim stało się uruchomienie oprogramowania FRED. W tym celu niezbędne było skonfigurowanie pliku parametrów konfiguracyjnych. W momencie, gdy potwierdzono, że wymiana informacji pomiędzy systemem FRED a ALFIPbus odbywa się bez zakłóceń, a dane udostępniane poprzez serwisy DIM wykazują zgodność z wartościami referencyjnymi pochodzącymi z aplikacji Control Server, system został uznany za przygotowany do kolejnych etapów rozwoju i testów w warunkach laboratoryjnych.

Równocześnie z prowadzoną integracją, członek zespołu odpowiedzialny za rozwój systemu WinCC OA w laboratorium, stworzył panel testowy w ramach systemu SCADA, który opisany jest w jego pracy inżynierskiej[46]. Panel ten przedstawiono na rys. 5.8. Został on zaprojektowany w oparciu o główny ekran aplikacji Control Server i miał na celu usprawnienie oraz ułatwienie procesu testowania nowo wdrażanych funkcjonalności systemu w laboratorium ALICE-FIT.

Oprócz wykorzystania interfejsu SCADA, poprawność działania serwisów, w tym tych odpowiedzialnych za proces wgrywania konfiguracji do elektroniki, była dodatkowo weryfikowana za pomocą aplikacji WebDID, umożliwiającej bezpośrednią interakcję z serwerem DIM.



Rysunek 5.8. Wersja systemu SCADA stworzona w celach nawiązania komunikacji z systemem FRED, bazująca na ekranie głównym aplikacji Control Server

Po pomyślnym zakończeniu opisanych powyżej etapów, system pozwalał na przystąpienie do dalszych prac związanych z rozwojem systemu SCADA.

5.3. Integracja systemu w detektorze FV0

W kolejnej fazie rozwoju systemu FRED, która została przeznaczona do celu integracji dla detektora FV0, zaowocowała także zmianą jego architektury przez zespół ALICE-FIT DCS. Głównym założeniem tej zmiany było podniesienie elastyczności systemu poprzez rezygnację z architektury powiązanej z fizycznymi rejestrami elektroniki na rzecz podejścia w większym stopniu opartego na parametryzacji rejestrów. Ta zmiana koncepcji systemu pozwoliła na dużo prostszą i niezawodniejszą realizację serwisów odpowiedzialnych za dostarczanie danych w stałych interwałach czasowych. Dodatkowo, umożliwiło to redukcję liczby aktywnych serwisów DIM, z rzędu kilkunastu tysięcy do zaledwie kilkudziesięciu.

Nowa architektura serwisów DIM w systemie FRED, przystosowana dla detektora FV0, została zaprojektowana w następujący sposób:

- **PARAMETERS:** Serwis zapewniający możliwość zarówno odczytu, jak i zapisu wartości pojedynczych parametrów elektroniki.
- **STATUS:** Serwis dedykowany stałemu i cyklicznemu odczytowi zbioru parametrów, które w bazie danych zostały oznaczone jako wymagające regularnego odświeżania.
- **SET_PHASE_DELAY:** Serwis funkcjonalnie zbliżony do PARAMETERS, lecz wyspecjalizowany w obsłudze parametrów DELAY_A i DELAY_C dla płyty TCM. Wykonuje on dodatkowe operacje systemowe niezbędne do zapewnienia prawidłowego działania elektroniki po dokonaniu zmiany tych wartości.

- COUNTER_RATES: Serwis odpowiedzialny za pobieranie danych z rejestrów typu FIFO w stałych odstępach czasu, dostarczający aktualne wartości Counter'ów oraz obliczający na ich podstawie Rate'y w oparciu o informacje dotyczące interwału czasowego.
- RESET_SYSTEM: Serwis inicjujący procedurę restartu elektroniki.
- RESET_ERRORS: Serwis przeznaczony do kasowania flag błędów w systemie elektroniki odczytowej.
- RESET: Serwis o charakterze uniwersalnym, uruchamiający procedurę resetowania danego segmentu elektroniki w zależności od wartości parametru przesłanego w komendzie DIM.
- CONFIGURATIONS: Serwis realizujący proces wgrywania konfiguracji do elektroniki, bazując na nazwie konfiguracji przekazanej w komendzie.
- CONFIGURATION_DB_BROKER: Serwis pełniący rolę pośrednika w komunikacji pomiędzy systemem SCADA a bazą danych Oracle, pozwala na pobieranie, aktualizacje oraz dodawanie nowych konfiguracji i parametrów do nich.
- HISTOGRAMS: Serwis dedykowany procesowi histogramowania, obejmujący ustawianie i resetowanie rejestrów histogramów, ich odczyt, a także oferujący opcjonalną funkcję automatycznego odczytu danych w stałych interwałach czasowych.

Po zakończeniu implementacji systemu, należało przeprowadzić testowanie całego oprogramowania. W pierwszej kolejności, w laboratorium ALICE-FIT, skonfigurowano odrębną instancję bazy danych Oracle. Następnie dokonano niezbędnych modyfikacji w plikach konfiguracyjnych systemu FRED, to znaczy ustawiono definicje serwisów DIM, parametry komunikacji z systemem ALFIPbus oraz dane dostępowe do bazy danych i serwera DIM. Zaktualizowano również system ALFIPbus do jego najnowszej wersji, która została zmodyfikowana w trakcie równoległego rozwoju systemu FRED. Po skonfigurowaniu wszystkich plików, uruchomiono kompletny system ALFRED wraz z systemem SCADA w środowisku WinCC OA, a następnie przeprowadzono szczegółowe testy wszystkich zaimplementowanych funkcjonalności w warunkach laboratoryjnych.

Po uzyskaniu pomyślnych wyników testów laboratoryjnych, system został przeniesiony do sieci AliDCS na serwery przeznaczone dla detektora FV0. W tym celu przygotowane zostały dwie nowe maszyny: jedna z systemem CentOS 7 dla ALFIPbus, drugą z systemem Red Hat 9 dla FRED.

Pierwsze testy poprawności działania całego systemu zostały przeprowadzone w oparciu o obserwację zachowania systemu za pomocą aplikacji WebDID dostępnej w przeglądarce internetowej oraz interakcję z systemem SCADA w środowisku WinCC OA. Przeprowadzone testy (odczytu, zapisu, przeprowadzania operacji resetów oraz wiele innych) potwierdziły poprawność integracji systemu z infrastrukturą detektora FV0.

6. Rozwój i wdrożenie systemu SCADA

6.1. Wprowadzone Rozwiązania w Systemie SCADA WinCC OA

Rozwój systemu SCADA w środowisku WinCC OA, został podzielony na kilka etapów. W celu wykorzystania istniejących już schematów i wyeliminowania konieczności rozpoczęcia pracy od początku, wykorzystana została najnowsza wersja projektu systemu dla detektora FV0. Została ona scalona z projektem interfejsu użytkownika dla aplikacji Control Server. Dzięki temu zminimalizowano liczbę zmian w komunikacji z komponentami zewnętrznymi oraz systemami centralnymi eksperymentu. Aktualna w tamtym momencie wersja projektu zapewniała następujące funkcjonalności:

- Realizację komunikacji z kratą VME oraz kratą HV CAEN z wykorzystaniem protokołu OPC UA.
- Utrzymanie łączności z centralnymi systemami eksperymentu ALICE oraz akceleratora LHC. Obejmowało to akwizycję informacji dotyczących wykonywanych operacji i bieżących parametrów, a także dalszą publikację danych pochodzących z detektora.
- Zapewnienie połączenia z centralnym systemem zarządzania stanem detektora FSM.
- Umożliwienie komunikacji z aplikacją Control Server poprzez serwer DIM działający na localhost, co pozwala na odczyt parametrów konfiguracyjnych i statusów oraz na wysyłanie poleceń sterujących.

Przy wykorzystaniu tej wersji projektu można było zastosować sprawdzone już schematy tworzenia paneli użytkownika, strukturyzacji kodu oraz bibliotek programistycznych, a także definicji datapoint'ów.

Zanim przystąpiono do samych prac implementacyjnych, należało przeanalizować i stworzyć plan wdrożenia nowej wersji systemu, co obejmowało zrozumienie istniejącego projektu oraz przyjętych w nim standardów nazewnictwa i konwencji programistycznych. Jednym z głównych celów podczas rozwoju systemu było dążenie do jak największego ustandaryzowania tworzonego projektu oraz wykorzystywanych w nim manager'ów, tak aby projekt mógł zostać zaimplementowany na innych detektorach FIT. Podczas rozpoczynania prac, projekt musiał być zdolny do współpracy z pierwotną implementacją systemu FRED, ponieważ jego docelowa wersja nie była jeszcze gotowa do użytku w laboratorium. Z tego powodu, funkcja odpowiedzialna za nawiązywanie połączenia z serwerem DIM i pobieranie wartości z publikowanych serwisów systemu FRED została zaimplementowana w dwóch wariantach: pierwszy przetwarzał odpowiedzi publikowane na dedykowanych serwisach dla rejestrów FEE, podczas gdy drugi obsługiwał informacje publikowane na serwisach o nazwie STATUS. Ponadto, utworzono dwie główne grupy datapoint'ów: pierwsza grupa była przeznaczona do przechowywania danych pochodzących bezpośrednio z rejestrów systemu elektroniki odczytowej, gdy druga przechowywała

informację o odpowiedziach na serwisach DIM, a następnie zawierała osobne datapoint'y dla każdego parametru.

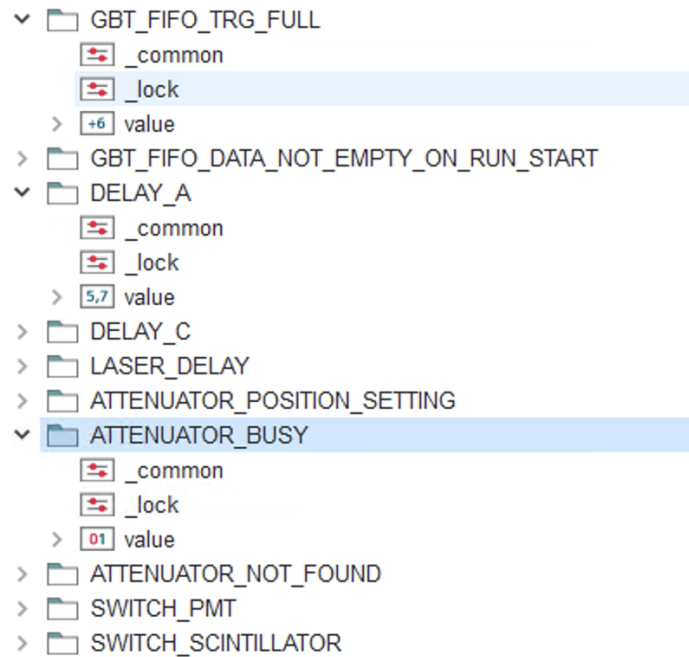
W celu utworzenia potrzebnych parametrów dla nowo implementowanej wersji oprogramowania FRED, nawiązano połączenie z bazą danych Oracle, która przechowywała definicje parametrów dla obu wersji systemu FRED. Następnie wyeksportowano nazwy wszystkich zdefiniowanych parametrów, i jednocześnie definiując ich typy danych. Zidentyfikowano następującą listę typów:

- **Boolean** (typ logiczny) – dla parametrów reprezentujących stan dla pojedynczego bitu.
- **Float** (liczba zmiennoprzecinkowa) – dla parametrów, dla których zdefiniowano wzór matematyczny przeliczający wartość z formatu elektronicznego na fizyczny.
- **Int** (liczba całkowita ze znakiem, 32-bitowa) – dla parametrów, które mogły przyjmować wartości ujemne.
- **UInt** (liczba całkowita bez znaku, 32-bitowa) – dla wszystkich pozostałych parametrów numerycznych.

Powyższa procedura została wykonana oddzielnie dla parametrów systemu elektronicznego dla płyty TCM oraz dla parametrów płyt PM. W kolejnym kroku, dysponując dwoma plikami tekstowymi zawierającymi listy parametrów, przystąpiono do napisania skryptu, który iterował kolejno po obu plikach i automatycznie tworzył dwa nowe typy datapoint'ów: TCM_FRED dla parametrów płyty TCM oraz PM_FRED dla parametrów płyt PM. Skrypt ten został następnie uruchomiony, na żądanie, jako osobny manager w systemie SCADA. W kolejnych etapach rozwoju projektu, skrypt ten był wykorzystywany do zbiorowego dodawania nowych datapoint'ów oraz definiowania ich typów.

Gdy utworzono typy datapoint'ów o nazwach TCM_FRED oraz PM_FRED przystąpiono do dodawania dla nich podtypów. Najpierw jednak pod kategorią TCM_FRED utworzono jeden datapoint o tej samej nazwie. Poniżej na rysunku 6.1 znajduje się lista niektórych podtypów tego datapoint'u.

Natomiast pod typem PM_FRED należało utworzyć datapointy, których nazwy odpowiadały identyfikatorom podłączonych płyt PM. Mając na uwadze to, że w laboratorium wykorzystywane są płyty PMA0 i PMC0, a w docelowym detektorze FV0 używane są płyty PMA0, PMA1, PMA2, PMA5, PMA6 i PMA7, utworzono datapointy o wszystkich wymienionych nazwach, stosując spójną konwencję nazewnictwa: <nazwa płyty PM (np. PMA0)>_FRED. Po utworzeniu podstawowych datapoint'ów dla parametrów FEE, dla niektórych z nich, takich jak LASER_DIVIDER na płycie TCM oraz GBT_CONTINUOUS_PATTERN na płytach TCM i PM, konieczne było stworzenie dodatkowych datapoint'ów typu Long (reprezentujących 63-bitowe liczby całkowite). W tym celu wykorzystano mechanizm datapoint function (funkcji wywoływanej automatycznie przy zmianie wartości jednego z pomocniczych datapoint'ów), który pozwolił na złączenie dwóch oddzielnych parametrów (dla LASER_DIVIDER były to: LASER_DIVIDER_LSB i LASER_DIVIDER_MSB) w celu



Rysunek 6.1. Część z utworzonych subparametrów dla datapoint'u TCM_FRED

stworzenia jednego, który był następnie wyświetlany i edytowany w interfejsie użytkownika.

Gdy został już przygotowany pełny zestaw parametrów, można było przystąpić do implementacji dedykowanego manager'a odpowiedzialnego za komunikację z serwerem DIM. Ze względu na fakt, iż na tym etapie możliwe było testowanie jedynie pierwotnej wersji systemu FRED, konfiguracja manager'a została dostosowana do jej specyfiki. W implementacji wykorzystano funkcje `appendService` oraz `appendCommand`, które umożliwiają podłączenie zdefiniowanego datapoint'u do konkretnego serwisu lub komendy DIM. Dzięki wykorzystaniu tego mechanizmu, podczas zmiany informacji publikowanej na serwisie DIM, wartość połączanego datapoint'u jest automatycznie odświeżana. Analogicznie, w przypadku komendy, zmiana wartości w odpowiednim datapoint'cie inicjuje wysłanie komendy DIM do systemu FRED. To podejście pozwala na sprawną wymianę danych pomiędzy systemami.

Właśnie z powodu tej sprawnej wymiany danych, w pierwotnej wersji FRED'a połączenie było nawiązywane bezpośrednio z datapoint'ami końcowymi, które później wyświetlane były w panelach. W nowej wersji jednak został utworzony specjalny typ datapoint'u o nazwie FRED, który zawierał na tym etapie prac podtypy o takich nazwach jak:

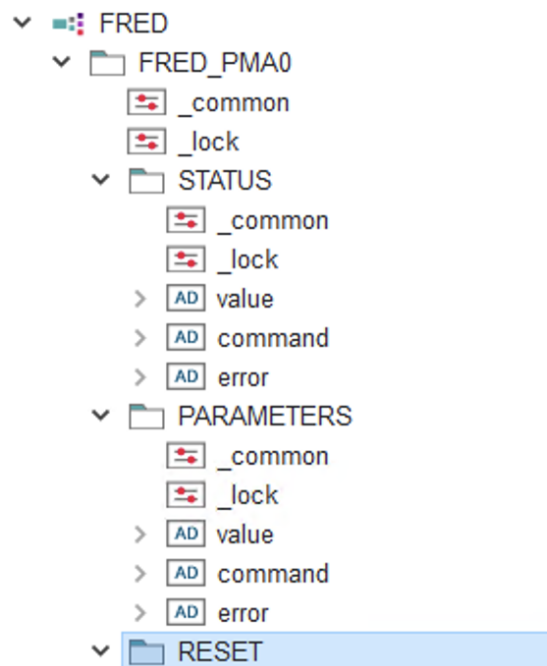
- STATUS
- PARAMETERS
- RESET
- CONFIGURATIONS

- RESET_ERRORS
- RESET_SYSTEM
- COUNTER_RATES
- SET_PHASE_DELAY

Każdy z tych parametrów zawierał kolejne podtypy, takie jak:

- value - przechowujący informacje z serwisu _ANS
- error - przechowujący informacje z serwisu _ERR
- command - przechowujący informacje dla komendy _REQ

Następnie dla każdej płyty tworzone były datapoint'y według schematu FRED_<nazwa płyty>. Poniżej na rysunku 6.2 znajduje się część utworzonych parametrów.



Rysunek 6.2. Część z utworzonych subparametrów dla datapoint'u FRED_PMA0

Posiadając dane z serwisów DIM w systemie WinCC OA, należało przetworzyć wiadomości przychodzące na powyższych serwisach do pojedynczych datapoint'ów w manage-rze DIM.

Gdy biblioteka w projekcie WinCC, odpowiedzialna za komunikację z serwerem DIM, została stworzona i zaimplementowana, można było przystąpić do jej uruchomienia i weryfikacji działania. W trakcie tego procesu wykryte zostało kilkanaście błędów implementacyjnych oraz brak implementacji dla niektórych serwisów. Wymagało to rozwiązania znalezionych problemów, aby móc dalej rozwijać system. Po pomyślnym przejściu tego etapu, możliwe stało się rozpoczęcie prac nad tworzeniem i rozwojem interfejsu użytkownika.

W celu rozwoju interfejsu użytkownika należało stworzyć plan, który pozwoliłby, aby przejście na docelową wersję systemu FRED, odbyło się bez konieczności gruntownej

przebudowy. Mając to na uwadze, została przyjęta spójna hierarchia paneli, które zostały zaprojektowane tak, aby funkcje wywoływane przy zmianie wartości danego rejestru dokonywały przeliczeń tych wartości. Gdy później miała zostać wdrożona nowa wersja systemu FRED, przeliczenia miały być bezpośrednio podstawione pod wartości pojedynczych parametrów przekazywanych z serwisu STATUS.

Kolejną rzeczą, która musiała być sprawdzona jak wygląda wydajność systemu było odświeżanie danych w interfejsach użytkowników. Wszystkie funkcje odpowiedzialne za wyświetlanie informacji były wywoływane poprzez wykorzystanie mechanizmu dpConnect, który pozwala na powiązanie funkcji zdefiniowanej w kodzie z listą monitorowanych parametrów (zmiana któregośkolwiek z tych parametrów wywołuje automatycznie powiązaną funkcję). Panele użytkownika zostały również pogrupowane tematycznie, aby uniknąć niepotrzebnego powielania powyższego mechanizmu. Dodatkowo, po konsultacjach z ekspertami, zaczęto stosować dodatkową technikę w celu optymalizacji kodu, polegającą na przekazywaniu większej liczby parametrów do pojedynczych funkcji i tym samym ograniczaniu ich liczby. Prowadziło to do tworzenia bardziej rozbudowanych funkcji, jednak często ich podstawowa funkcjonalność sprowadzała się jedynie do zmiany tekstu lub koloru etykiet, tym samym nie wymagały one dużych zasobów komputera.

Rozpoczynając prace nad rozwojem części odpowiedzialnej za interakcję z użytkownikiem, należało skupić się w pierwszej kolejności na interfejsie użytkownika dla aplikacji Control Server. Z tego powodu przeprowadzono analizę kodu źródłowego wspomnianej aplikacji w celu odnalezienia dokładnego sposobu przeliczania wartości dla danych rejestrów FEE lub sposobu wyświetlania wartości i informacji w konkretnym miejscu interfejsu. Konieczne było również wykorzystanie środowiska programistycznego QtCreator do wizualizacji projektu aplikacji Control Server, co znacząco ułatwiało odnalezienie funkcji przypisanych do danego identyfikatora dla wybranego komponentu graficznego. Podczas dalszego rozwoju systemu należało także zaimplementować dodatkowe funkcje pomocnicze, służące do weryfikacji i konwersji wartości pomiędzy różnymi systemami liczbowymi – dziesiętnym i szesnastkowym. Posiadając te funkcje, możliwe stało się dodanie funkcji walidujących, które sprawdzały, czy wprowadzana wartość nie wykracza poza dozwolony zakres. Zostały one umieszczone w bibliotekach z funkcjami pomocniczymi celem łatwiejszego ich używania w różnych miejscach. Do utworzonych funkcji należało podać zakres bitów na danym rejestrze. Zakres miał być zdefiniowany jako wartość minimalna i maksymalna, lecz gdy pola te były puste, był on obliczany automatycznie bazując na bicie początkowym i końcowym. Razem z zakresami wartości i bitów przechowano także informacje czy pole wyświetlane jest jako wartość szesnastkowa. Posiadając powyższą implementację, należało przejść do ich integracji z każdym przyciskiem i polem edytowalnym w panelu użytkownika, tak aby walidacja była dodana do projektu.

Jedną z kluczowych zalet tworzenia paneli użytkownika w systemie WinCC OA jest ich łatwa parametryzacja oraz możliwość kopiowania i ponownego wykorzystania. Tworząc

jeden panel, można przenosić go do paneli nadrzędnych, przekazując jedynie unikalny identyfikator jako parametr wejściowy. Mechanizm ten został z powodzeniem wykorzystany między innymi w panelu wyświetlającym informacje dla poszczególnych kanałów danej płyty PM, zostało to ukazane na rysunku 6.3. Takie podejście pozwoliło na uniknięcie konieczności przenoszenia i kopiowania tej samej logiki w kodzie dla każdego przycisku, pola tekstowego i etykiety dwanaście razy (dla każdego kanału), a jedynie na dynamiczne przekazanie identyfikatora kanału jako parametru.

PMAO

STA1

NO TRG

Time alignment

Threshold calibr

ADC delay

CFD threshold

ADC zero

CFD zero

ADC0

ADC1

Range correction

Base line

Mean charge

RMS

TDC RAW data

CFD count TRG count

Rate, HZ

Copy actual values to settings for this PM

Apply all settings for this PM

Reset counts

CH1	ON	896 400 2163 1500	10246 8000 300 300	144 144 30 80	2048 2048 2048 2048	39 0 74 0	1.00 1.00	7F 00	0 0 0 0	0 0
CH2	ON	909 800 1956 1500	10567 10567 300 300	333 200 -140 20	2048 2048 2048 2048	85 0 91 0	1.00 0.00	00 00	0 0 0 0	0 0
CH3	ON	863 863 2277 2083	10872 10872 300 300	286 238 20 20	2048 2048 2048 2048	30 0 40 0	1.00 1.41	07 00	0 0 0 0	0 0
CH4	ON	929 929 2133 1972	10750 10750 300 300	328 328 -70 -70	2048 2048 2048 2048	86 0 51 0	1.00 1.00	7F 00	0 0 0 0	0 0
CH5	ON	952 952 1955 1829	10920 10920 300 300	326 326 -5 -5	2048 2048 2048 2048	49 0 76 0	1.00 1.00	00 00	0 0 0 0	0 0
CH6	ON	931 931 2286 2163	11232 11232 300 300	322 322 95 95	2048 2048 2048 2048	49 0 93 0	1.00 1.41	00 00	0 0 0 0	0 0
CH7	ON	920 930 2513 2790	10683 10683 300 300	314 170 -80 -80	2048 2048 2048 2048	30 1 79 0	1.41 1.00	66 08	951.71M 63.76K	467822 20
CH8	ON	946 946 2252 2173	10315 10315 300 300	295 295 -60 -60	2048 2048 2048 2048	89 0 90 0	1.00 1.00	78 00	0 0 0 0	0 0
CH9	OFF	865 865 2480 2200	9002 9002 300 300	90 90 60 60	2048 2048 2048 2048	42 0 49 0	1.00 1.00	00 00	0 0 0 0	0 0
CH10	OFF	859 859 2575 2575	8502 8502 300 300	252 252 30 30	2048 2048 2048 2048	78 0 41 0	1.00 1.00	3C 00	0 0 0 0	0 0
CH11	OFF	859 859 2710 2710	9263 9263 300 300	334 334 220 220	2048 2048 2048 2048	38 0 30 0	1.00 1.00	00 00	0 0 0 0	0 0
CH12	OFF	860 860 2645 2645	8915 8915 300 300	182 182 270 270	2048 2048 2048 2048	31 0 53 0	1.41 1.41	7F 00	0 0 0 0	0 0

Rysunek 6.3. Pola parametrów dla kolejnych kanałów na płytach PM

Podczas prac pojawiły się również inne problemy. Jednym z głównych, który ujawnił się podczas implementacji interfejsu użytkownika odwzorowującego funkcjonalność aplikacji Control Server, była dynamiczna zmiana wyświetlanych danych dla poszczególnych płyt PM podczas przełączania między panelami za pomocą przycisków opisujących nazwy tych płyt. Ta część interfejsu ukazana jest na rysunku 6.4. Problem ten wynikał z faktu, że dla każdej płyty PM wykorzystywany był ten sam, pojedynczy panel, aby nie wykorzystywać kilkanaście razy tego samego pliku interfejsu.

TCM	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
PM OK:	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
TRGC:	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

Rysunek 6.4. Identyfikatory płyty TCM i płyt PM w systemie SCADA

Wymagało to zaimplementowania mechanizmu globalnego przekazywania informacji o zmianie wybranej przez użytkownika płyty do paneli znajdujących się na niższych

warstwach hierarchii. Po konsultacjach z ekspertami i wielu próbach znalezienia najlepszego sposobu implementacji tego komponentu, podjęto decyzję o całkowitej zmianie architektury panelu głównego bazującego na aplikacji Control Server. Postanowiono, że przyciski zawierające nazwy poszczególnych płyt systemu elektroniki odczytowej zostaną umieszczone na najwyższej warstwie interfejsu, co umożliwi bezpośrednie wywoływanie funkcji w panelach-dzieciach. Wymagało to jednak wprowadzenia dodatkowych modyfikacji w wielu interfejsach, tak aby posiadały one odpowiednie funkcje, które mogłyby być wywołane z zewnątrz w celu odświeżenia wyświetlanych danych. Konieczne stało się również implementowanie wywołań funkcji `dpDisconnect` w sytuacjach, gdy wcześniej dla danego obiektu wywołana była funkcja `dpConnect`. Należało także przechowywać informację o poprzednio wybranej nazwie płyty PM oraz o nowo przekazanej nazwie, w celu poprawnego odłączania i podłączania odpowiednich datapoint'ów.

W zbliżonym okresie, system FRED, tworzony równolegle przez zespół ALICE-FIT DCS, osiągnął poziom rozwoju pozwalający na jego pierwsze uruchomienie w laboratorium ALICE-FIT. Wymagało to jego przeniesienia na docelowe serwery, skompilowania kodu źródłowego oraz odpowiedniej konfiguracji połączenia z bazą danych. Dodatkowo, wymagane było dostosowanie samej bazy danych do specyfiki środowiska laboratoryjnego. Obejmowało to między innymi zmianę listy podłączonych płyt PM oraz uzupełnienie tabeli zawierającej parametry statyczne (między innymi takie jak częstotliwość zegara). W systemie WinCC OA należało zweryfikować poprawność działania Manager'a DIM, który wcześniej został zaimplementowany bazując na przekazanej wstępnej dokumentacji. Gdy te działania zostały przeprowadzone, można było uruchomić zintegrowany system ALFRED wraz z systemem SCADA, a następnie zweryfikowano poprawność połączenia pomiędzy nimi. Dysponując danymi publikowanymi na DIM, można było sprawdzić, czy wartości zapisywane w datapoint'ach utworzonych dla parametrów FEE, a zwracane przy użyciu serwisu STATUS, są poprawne i zgodne z oczekiwaniami. Jednocześnie okazało się, że utworzono bardzo duże zbiory datapoint'ów dla parametrów kolejnych kanałów dla płyt PM. Jak wyszło z analizy projektu, nie musiały być one przechowywane osobno, lecz mogły być przechowywane w jednej liście dla wszystkich kanałów i wszystkich płyt PM w podziale na typ parametru, tak były przekazywane wcześniej przez serwer DIM z aplikacji Control Server. Konieczne stało się zatem rozdzielenie funkcji odpowiedzialnych za przetwarzanie odpowiedzi na serwisie STATUS dla płyt PM i TCM, a jednocześnie rozbudowanie logiki pozwalającej na zapis tych danych. Dodatkowe utrudnienie implementacyjne pojawiło się w przypadku serwisu COUNTER_RATES, gdzie również podjęto decyzję o wykorzystaniu istniejących datapoint'ów. Poniżej znajduje się główny fragment biblioteki odpowiedzialny za inicjalizację serwisów oraz komend DIM dla systemu FRED.

Listing 1. Fragment kodu odpowiedzialny za inicjalizację serwisów oraz komend DIM

```
dyn_string board_names = makeDynString("PMA0", "PMA1", "PMA2",
"PMA5", "PMA6", "PMA7", "PMC0");
dyn_string services_endpoints = makeDynString("ANS", "ERR", "REQ");
```

6. Rozwój i wdrożenie systemu SCADA

```
dyn_string datapoints_endpoints = makeDynString("value", "error", "command");
dyn_string services_names = makeDynString("STATUS", "PARAMETERS", "RESET",
"CONFIGURATIONS", "CONFIGURATION_DB_BROKER", "RESET_ERRORS", "RESET_SYSTEM",
"COUNTER_RATES", "SET_PHASE_DELAY");

void appendFREDServices(){
    for(int i=0; i<=board_names.count(); i=i+1){
        string prefix="PM";
        string board_name;
        string board_dp;
        if(i==0){
            prefix="TCM";
            board_name="TCMO";
            board_dp="TCM";
        }
        else{
            board_name=board_names[i];
            board_dp=board_names[i];
        }
        for(int j=1; j<=services_names.count(); j++){
            for(int z=1; z<=2; z++){
                appendService(getSystemName()+"FRED_"+board_dp+"."+services_names[j]+"."
+datapoints_endpoints[z],
"FRED/"+prefix+"/"+board_name+"/"+services_names[j]+"_"
+services_endpoints[z]);
            }
            appendCommand(getSystemName()+"FRED_"+board_dp+"."+services_names[j]+"."
+datapoints_endpoints[3],
"FRED/"+prefix+"/"+board_name+"/"+services_names[j]
+"_"+services_endpoints[3]);
        }
    }

    appendCommand(getSystemName()+"FRED_ERRORS.RESTART_FRED", "FRED/EXIT");

    dpConnect("splitTCMStatusServices", getSystemName()+"FRED_TCM.STATUS.value");
    dpConnect("splitTCMStatusServices",
getSystemName()+"FRED_TCM.PARAMETERS.value");
    dpConnect("splitTCMCounterRatesServices",
getSystemName()+"FRED_TCM.COUNTER_RATES.value");
    dpConnect("splitConfigurationService",
getSystemName()+"FRED_TCM.CONFIGURATION_DB_BROKER.value");
    dpConnect("displayError", getSystemName()+"FRED_TCM.COUNTER_RATES.error");
    dpConnect("displayError", getSystemName()+"FRED_TCM.STATUS.error");
    dpConnect("displayError", getSystemName()+"FRED_TCM.PARAMETERS.error");
    dpConnect("displayError", getSystemName()+"FRED_TCM.RESET.error");
    dpConnect("errorOnConfigurationService",
getSystemName()+"FRED_TCM.CONFIGURATION_DB_BROKER.error");
    dpConnect("errorOnResetSystem",
getSystemName()+"FRED_TCM.RESET_SYSTEM.error");
    for(int i=0; i<=2; i++){
        dpConnect("displayError",
getSystemName()+"FRED_PMA"+i+".COUNTER_RATES.error");
        dpConnect("displayError", getSystemName()+"FRED_PMA"+i+".STATUS.error");
        dpConnect("displayError", getSystemName()+"FRED_PMA"+i+".PARAMETERS.error");
        dpConnect("displayError", getSystemName()+"FRED_PMA"+i+".RESET.error");
    }
    for(int i=5; i<=7; i++){
        dpConnect("displayError",
getSystemName()+"FRED_PMA"+i+".COUNTER_RATES.error");
        dpConnect("displayError", getSystemName()+"FRED_PMA"+i+".STATUS.error");
        dpConnect("displayError", getSystemName()+"FRED_PMA"+i+".PARAMETERS.error");
        dpConnect("displayError", getSystemName()+"FRED_PMA"+i+".RESET.error");
    }
}
```

```

}

string board_name=board_names[1];
dpConnect("splitPMStatusServices",
getSystemName()+"FRED_"+board_name+".STATUS.value");
dpConnect("splitPMStatusServices",
getSystemName()+"FRED_"+board_name+".PARAMETERS.value");
dpConnect("splitPMCounterRatesServices",
getSystemName()+"FRED_"+board_name+".COUNTER_RATES.value");
}

```

Gdy zmiany dla przetwarzania danych z serwera DIM zostały zintegrowane, przystąpiono do integracji systemu SCADA dla aktualnej wersji oprogramowania ALFRED. Oznaczało to w praktyce konieczność ponownej edycji większości paneli użytkownika, jednak zmiany te polegały głównie na podmianie wykorzystywanych datapoint'ów oraz na wprowadzeniu niewielkich modyfikacji w kodzie kolejnych paneli i bibliotek. Dodatkowo, należało wprowadzić zmiany dla paneli odpowiedzialnych za wyświetlanie wartości dla Counter'ów oraz Rate'ów, tak aby wykorzystywały one opisane wyżej listy parametrów.

Następnie można było przejść do weryfikacji brakujących datapoint'ów oraz parametrów FEE w bazie danych. Był to proces czasochłonny, który trwał praktycznie aż do momentu wdrożenia systemu na detektorze. Odnalezienie niektórych brakujących parametrów lub błędów w ich definicjach często było możliwe dopiero podczas testowania systemu przez eksperta. Niemniej jednak, duża część parametrów została zidentyfikowana już na samym początku procesu edycji paneli, i we współpracy z zespołem możliwe stało się sprawne edytowanie zawartości bazy danych.

W kolejnym kroku, gdy przetwarzanie i wyświetlanie danych działało poprawnie, stało się możliwe podłączenie kolejnego istotnego serwisu DIM, nazwanego PARAMETERS. Serwis ten pozwalał na wysyłanie poleceń odczytu lub zapisu wartości poszczególnych parametrów na żądanie użytkownika. Równocześnie z nim został podłączony drugi serwis, SET_PHASE_DELAY. Podłączenie tych serwisów oraz dostosowanie logiki przetwarzania odpowiedzi zwrotnej, tak aby odświeżana była wartość danego parametru w odpowiednim datapoint'cie, pozwoliło na dalszy rozwój interfejsu użytkownika o część odpowiedzialną za implementację funkcjonalności przycisków i wysyłanie komend sterujących.

Po zakończeniu tych prac, skupiono się na podłączeniu komend DIM realizujących funkcje resetowania, takie jak: RESET_SYSTEM, RESET_ERRORS oraz RESET. Pozwoliło to na odzwierciedlenie tej części interfejsu użytkownika, w nowym systemie SCADA przy użyciu systemu ALFRED. Po tym etapie konieczne było przeprowadzenie testowania całości systemu w laboratorium celem sprawdzenia, czy wszystkie etykiety i pola danych wyświetlają poprawne informacje. Proces weryfikacji można było przeprowadzić, mając równolegle uruchomioną aplikację Control Server, jednakże w takim przypadku nie było możliwe weryfikowanie wartości Rate'ów i Counter'ów pobieranych bezpośrednio z rejestrów FIFO. W tym celu wymagana była konsultacja z ekspertem detektora. Po przejściu

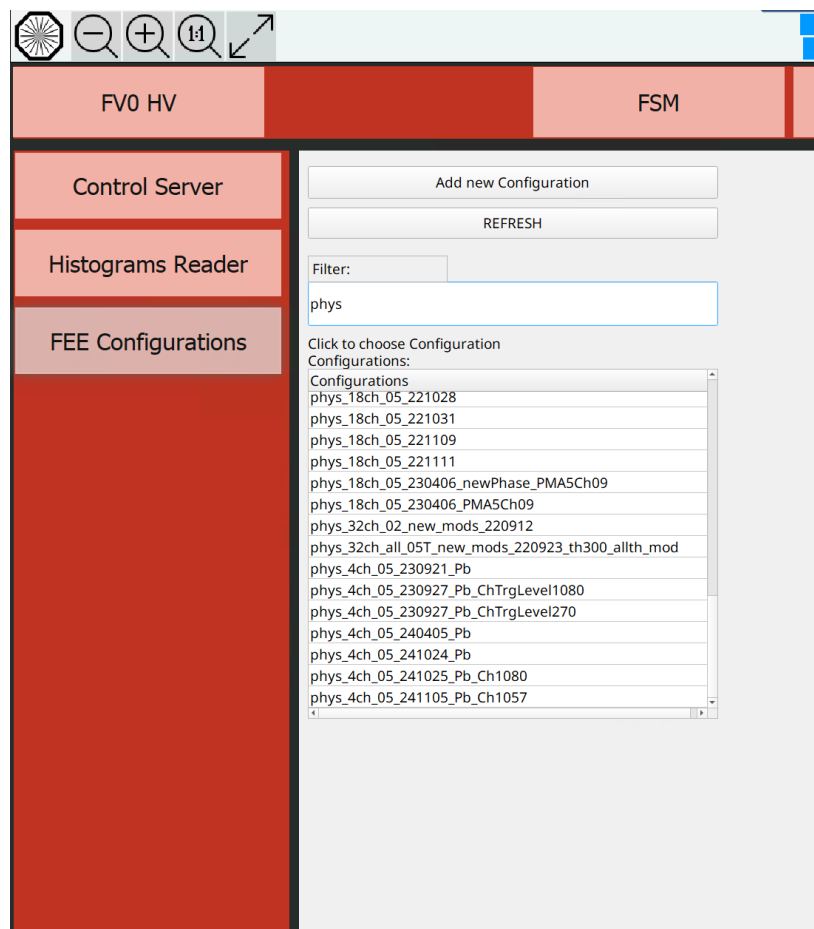
weryfikacji i rozwiązywaniu problemów, które pojawiły się podczas testowania, możliwe było rozpoczęcie implementacji kolejnych funkcjonalności.

Można było rozpocząć prace nad interfejsem do zarządzania konfiguracjami systemu elektroniki odczytowej, obejmującymi ich wyświetlanie, edytowanie, dodawanie do bazy danych oraz aplikowanie do FEE. Aplikacja Control Server pozwalała na przechowywanie konfiguracji w plikach tekstowych w formacie INI z rozszerzeniem `.cfg`, co miało zarówno swoje zalety, jak i pewne ograniczenia. Relacyjna baza danych w porównaniu do przechowywania danych w plikach pozwala na zapisywanie znacznie większej ilości szczegółowych informacji, takich jak to kto i kiedy wprowadził daną konfigurację do systemu, oraz pozwala ona na rozbitcie poszczególnych rejestrów na pojedyncze parametry. Tym samym, takie podejście jest znacznie bardziej bezpieczne, jeżeli chodzi o gromadzenie i uporządkowanie danych, i pozwala na edycję jedynie tych parametrów, które faktycznie tego wymagają, a nie tworzenie całego pliku na nowo.

Wraz z zespołem, podczas implementacji wyżej opisanej funkcjonalności, została podjęta decyzja o ograniczeniu bezpośrednich połączeń z bazą danych Oracle. Dostęp do niej miał mieć jedynie system FRED. Z tego powodu, na tym systemie dodatkowo został zaimplementowany dedykowany serwis DIM, pozwalający na przetwarzanie zapytań kierowanych do bazy danych od WinCC. Następnie zwracana była przetworzona odpowiedź do systemu SCADA. Serwis ten został nazwany `CONFIGURATION_DB_BROKER`, a implementacja przetwarzania odpowiedzi z systemu FRED w WinCC została zaimplementowana z wykorzystaniem biblioteki odpowiedzialnej za komunikację z serwerem DIM, podobnie do danych publikowanych na innych serwisach. Okazało się tutaj jednak, że serwis ten może być wykorzystywany przez wielu użytkowników jednocześnie oraz że mogą być zwracane całkowicie różne typy danych, takie jak lista konfiguracji, lista parametrów, a także odfiltrowane listy konfiguracji lub parametrów. Z tego powodu należało zaimplementować mechanizm kolejowania zapytań. Pozwoliło to na przechowywanie listy poleceń, które w kolejnych chwilach zostały wysłane przez kolejnych użytkowników. W następnym kroku przystąpiono do stworzenia panelu użytkownika dedykowanego wyświetlaniu listy zapisanych konfiguracji, to w nim też miała być osadzona istotna część kolejowania zapytań. Panel ten pozwalał na pobieranie listy przy inicjalizacji oraz po naciśnięciu przez użytkownika przycisku REFRESH. Jest on ukazany na rys. 6.5. Drugi z paneli ukazany na rys. 6.6, odpowiedzialny jest za prezentację listy parametrów dla wybranej konfiguracji, po jej dwukrotnym naciśnięciu. Podobnie jak poprzedni interfejs, pobiera on listę parametrów przy inicjalizacji i pozwala na edycję lub usunięcie wybranego parametru po dwukrotnym jego naciśnięciu. Na tym panelu należało również zaimplementować logikę wysyłania odpowiedniego polecenia przy użyciu komendy DIM do systemu FRED, który następnie, przy użyciu polecenia w języku SQL, wykonywał zmiany parametrów w bazie danych. Gdy funkcjonalność wyświetlania konfiguracji oraz ich edytowania została zaimplementowana, przystąpiono do stworzenia panelu odpowiedzialnego za dodawanie

nowych konfiguracji wraz z ich parametrami do bazy danych, interfejs ten został ukazany na rys. 6.7. Wymagało to w pierwszej kolejności stworzenia odpowiedniego datapoint'u oraz przygotowania struktury danych dla niego. Informacje, które przechowywał to między innymi lista parametrów, które powinny zostać wgrane do systemu FEE, wraz z dodatkową informacją, czy wartość danego parametru może zostać pobrana bezpośrednio z pola tekstowego w interfejsie, czy też musi zostać odczytana z aktualnego stanu systemu elektroniki odczytowej. W tym celu przygotowano dwie oddzielne listy: jedną dla płyty TCM, a drugą dla płyt PM. Struktura kolejnych wierszy na tych listach wyglądała następująco:

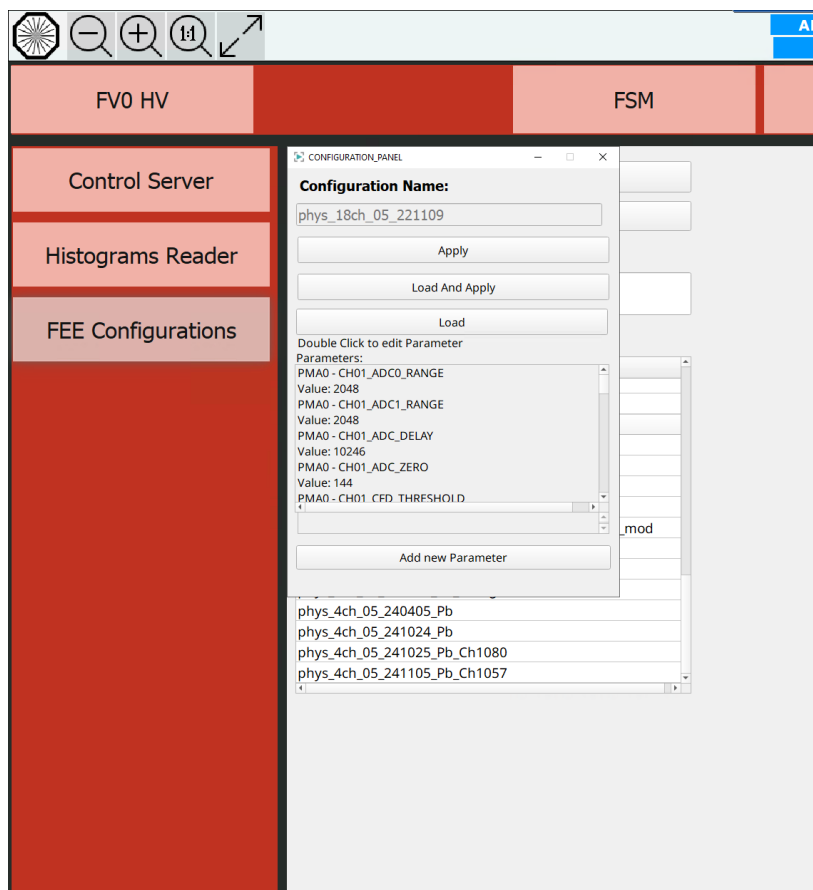
- „<nazwa parametru>,'0'” - dla parametrów posiadających pola edytowalne, „1'” - dla parametrów, których wartość mogła zostać odczytana jedynie z FEE>”



Rysunek 6.5. Ekran listy konfiguracji FEE

[32]

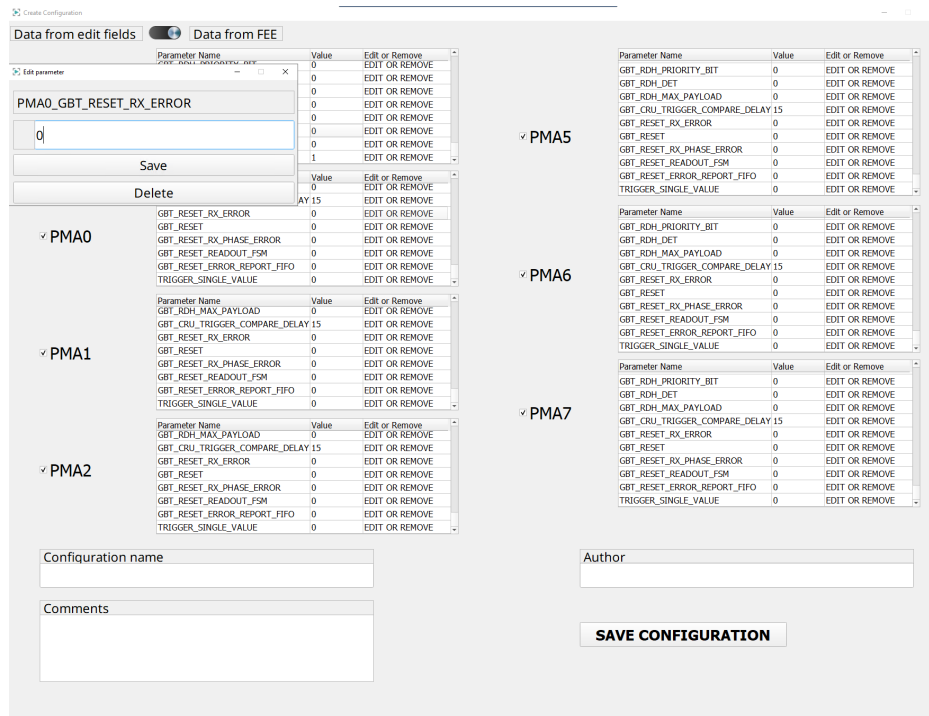
Powyższa lista parametrów została przygotowana, tak jak wcześniej, w oparciu o analizę kodu źródłowego aplikacji Control Server oraz zakresu rejestrów, który ta aplikacja wykorzystywała w celu zapisywania konfiguracji FEE. Lista parametrów, które zaliczały się do tego przedziału, została pobrana z bazy danych, a informacja o tym, czy dla danego parametru istnieje pole tekstowe w interfejsie, została oceniona poprzez weryfikację, czy istnieje odpowiedni datapoint połączony z polem edytowalnym. Gdy lista paramet-



Rysunek 6.6. Ekran parametrów konfiguracji FEE

[32]

trów była już gotowa, można było przejść do tworzenia właściwego panelu użytkownika w systemie SCADA. W tym celu wykorzystano standardowy komponent Table, pozwalający na dynamiczne tworzenie kolumn dla nazwy parametru, jego aktualnej wartości oraz przycisku umożliwiającego edycję. Dla płyty TCM oraz dla każdej z płyt PM stworzono osobną tabelę z odpowiednią listą parametrów. Dodatkowo, dla każdej płyty FEE dodano pole zaznaczenia (Checkbox), które pozwoliło użytkownikowi na elastyczne dodawanie konfiguracji tylko dla wybranych przez siebie płyt. Dodano również przełącznik, pozwalający na wybór trybu pobierania wszystkich parametrów z FEE lub, tak jak wspomniano wcześniej, pobieranie wartości tylko tam, gdzie parametry nie posiadają pól edytowalnych, a tam gdzie się da to z pól edytowalnych. W kolejnym kroku, dysponując listą parametrów w tabelach, po naciśnięciu przez użytkownika przycisku Edit or Remove otwierany był kolejny panel, pozwalający na edycję wybranego parametru. Przed jego utworzeniem należało jednak dodać definicje zakresów dla poszczególnych parametrów w projekcie w WinCC OA. W tym celu stworzono odpowiednie typy datapoint'ów o nazwach TCM_FRED_RANGES oraz PM_FRED_RANGES, posiadające następujące pola dla każdego parametru:



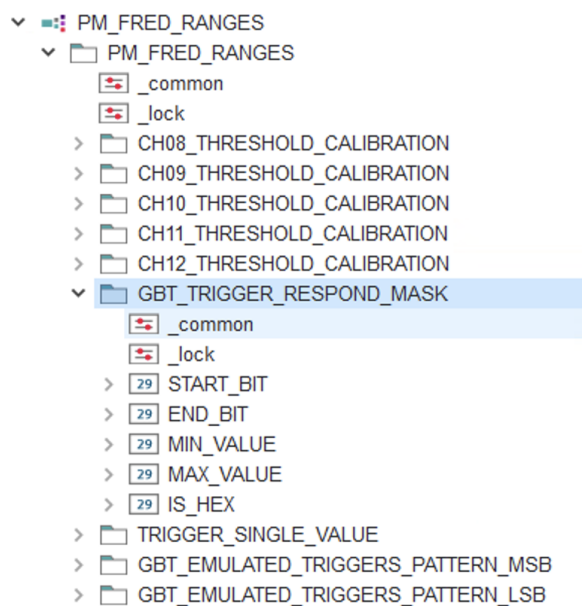
Rysunek 6.7. Ekran dodawania konfiguracji FEE

[32]

- IS_HEX - określający, czy dany parametr jest domyślnie wyświetlany w formacie szesnastkowym czy dziesiętnym
- START_BIT - określający bit startowy dla danego parametru w rejestrze elektroniki
- END_BIT - określający bit końcowy dla danego parametru w rejestrze elektroniki
- MIN_VALUE - określający minimalną dopuszczalną wartość dla parametru w rejestrze elektroniki
- MAX_VALUE - określający maksymalną dopuszczalną wartość dla parametru w rejestrze elektroniki

Dane te zostały uzupełnione w oparciu o wartości odczytane z bazy danych. Aby dodać je do systemu został wykorzystany stworzony na potrzeby tego zadania skrypt, który to umożliwił. Dzięki stworzonym datapoint'om, możliwe było dodanie opcji dynamicznego wyświetlania wartości w formacie szesnastkowym i dziesiętnym oraz zaimplementowanie mechanizmu ograniczania wprowadzanych wartości do maksymalnej i minimalnej dopuszczalnej wartości. Dodano również opcję usuwania danego parametru z listy parametrów tworzonej konfiguracji.

Przechodząc do dodawania nowych konfiguracji do bazy danych, stworzony został panel, do którego była możliwość wejścia z panelu FEE Configurations, gdzie wyświetlane były listy konfiguracji. Podczas dodawania nowej konfiguracji, użytkownik musiał uzupełnić dodatkowo pole nazwy konfiguracji oraz pole autora. Gdy te informacje zostały wprowadzone, wysyłana była odpowiednia komenda DIM, która inicjowała proces



Rysunek 6.8. Część z utworzonych subparametrów dla datapoint'u PM_FRED_RANGES

dodawania konfiguracji do bazy danych. Gdy z systemu FRED zwrócona została informacja o poprawnie wykonanej operacji, wysyłano kolejne polecenie, mające na celu dodanie wszystkich zdefiniowanych parametrów. Parametry te były pobierane z tabel wyświetlanych dla płyty TCM oraz dla poszczególnych płyt PM, które zawierały wszystkie zmiany wprowadzone przez użytkownika. Aby zsynchronizować operacje między systemem FRED a panelem użytkownika zostały stworzone datapoint'y pomocnicze, które pozwoliły stworzyć pewnego rodzaju maszynę stanów. Gdy wszystkie operacje zostały wykonane poprawnie, w interfejsie użytkownika pojawiał się komunikat o sukcesie, a okno dodawania konfiguracji było automatycznie zamykane po naciśnięciu przycisku OK.

Dysponując możliwością wyświetlania, dodawania oraz edytowania konfiguracji, należało zaimplementować opcję aplikowania wybranej konfiguracji do elektroniki oraz ładowania zapisanych parametrów konfiguracyjnych do pól edytowalnych w systemie SCADA. W tym celu wykorzystano serwis DIM o nazwie CONFIGURATIONS, na którym należało wysłać jedynie nazwę tej konfiguracji. Następnie system FRED aplikował konfigurację, pobierając jej parametry z bazy danych. W manager'ze odpowiedzialnym za komunikację poprzez DIM umieszczono także funkcję odpowiedzialną za przetwarzanie odpowiedzi zwrotnej na tym serwisie. W ten sposób, w panelu użytkownika, komenda ta jest wysyłana po naciśnięciu przycisku Apply lub Load&Apply, i gdy pojawia się informacja zwrotna o poprawnym jej zaaplikowaniu, wyświetlany jest komunikat o sukcesie. Dodatkowo, utworzono przycisk Load, który pozwala na załadowanie wartości dla kolejnych parametrów z wybranej konfiguracji do odpowiednich datapoint'ów systemu SCADA, które następnie pojawiają się w polach tekstowych interfejsu u wszystkich użytkowników. Ta sama operacja jest także automatycznie wykonywana poprzez przycisk Load&Apply.

Jedną z kolejnych funkcjonalności zaimplementowanych w systemie SCADA było

odtworzenie aplikacji HistogramReader. W tym celu został ustalony podział systemu na warstwę logiczną, realizowaną przez FRED, oraz warstwę prezentacji i zarządzania w WinCC OA. Do komunikacji pomiędzy tymi warstwami wykorzystano nowy serwis DIM o nazwie HISTOGRAMS. Pozwalał on na wysyłanie z systemu SCADA następujących komend:

- START – polecenie rozpoczynające proces ciągłego odczytywania histogramów i ich cyklicznego publikowania na wspomnianym serwisie DIM.
- STOP – polecenie zatrzymujące proces ciągłego odczytywania i publikowania histogramów z systemu elektronicznego.
- HISTOGRAMMING, <ON/OFF> – polecenie włączające lub wyłączające proces histogramowania bezpośrednio na elektronice poprzez ustawienie odpowiedniego bitu w rejestrze.
- READ – polecenie inicjujące pojedynczy, jednorazowy odczyt aktualnych danych histogramów z elektroniki.
- RESET – polecenie powodujące wyzerowanie zawartości rejestrów histogramów na elektronice.
- SELECT, [<wybrane typy histogramów: TIME, ADC0, ADC1>] – polecenie pozwalające na dynamiczne wybranie, które typy informacji z histogramów powinny być zwracane od systemu FRED do systemu SCADA w WinCC OA.
- BCID_FILTER, <OFF/0/1/. . . /3563> – polecenie umożliwiające kontrolę nad filtrem BCID (Bunch Crossing ID) w FEE.

Należy zaznaczyć, że ostatnie dwie z wymienionych komend (SELECT oraz BCID_FILTER) były możliwe do uruchomienia tylko dla płyt PM. Lista dostępnych płyt PM, prezentowana użytkownikowi do wyboru, była generowana automatycznie w oparciu o wartość parametru BITMASK_PM_MASK_SPI, który na kolejnych swoich bitach określa, która z płyt PM jest aktualnie uruchomiona i poprawnie skonfigurowana. Tym samym, po uruchomieniu procesu odczytu histogramów i wybraniu wszystkich dostępnych typów histogramów przy użyciu polecenia SELECT w komendzie DIM, dane do systemu WinCC OA były zwracane w następującym formacie:

```
[READOUT_ID]
CH<XX - numer kanału od 01 do 12>ADC0,<dane
dla kolejnych binów oddzielone przecinkami>
CH<XX - numer kanału od 01 do 12>ADC1,<dane
dla kolejnych binów oddzielone przecinkami>
CH<XX - numer kanału od 01 do 12>TIME,<dane
dla kolejnych binów oddzielone przecinkami>
...
READ_ELAPSED,<czas w ms>ms
PREV_ELAPSED,<czas w ms>ms
```

<dodatkowe informacje/błędy>

Natomiast dla płyty TCM zwracane były inne dane, które w aplikacji zostały określone jako Trigger per BC. W tym przypadku zwracane były trzy wiersze danych, gdzie pierwszy z typów danych był ustalany przez użytkownika i domyślnie był on wyłączony. Struktura publikowanej odpowiedzi dla płyty TCM wygląda następująco:

```
[READOUT_ID]
<wybrany typ danych>,<dane dla kolejnych binów oddzielone przecinkami>
OrA,<dane dla kolejnych binów oddzielone przecinkami>
OrC,<dane dla kolejnych binów oddzielone przecinkami>
READ_ELAPSED,<czas w ms>ms
PREV_ELAPSED,<czas w ms>ms
<dodatkowe informacje/błędy>
```

Dysponując powyższymi informacjami dotyczącymi formatu komunikacji z systemem FRED, należało w pierwszej kolejności opracować odpowiednią strukturę datapoint'ów oraz podzielić proces przetwarzania napływających danych na osobne procesy. W tym celu utworzono nowy, dedykowany typ datapointu o nazwie histograms, który przechowywał informacje o wartościach dla kolejnych zakresów czasu oraz energii dla poszczególnych kanałów w ramach jednej listy dla każdej płyty PM. Tym samym, każdy z modułów zawierał trzy odrębne listy danych z histogramów: TIME, ADC0 oraz ADC1.

Następnie należało opracować dokładny plan na przetwarzanie zapytań i odpowiedzi z serwisu DIM. Odpowiedź, która była zwracana z systemu FRED, najpierw była wstępnie przetwarzana na listę wartości przechowywaną jako pojedyncza zmienna typu string. Następnie z tej zagregowanej odpowiedzi wyciągano także informacje o czasach wykonania poszczególnych operacji. Dysponując tymi danymi, utworzono drugi manager, który dokonywał konwersji listy danych zapisanej w formacie tekstowym na zmienną typu dyn_int. Jest to operacja charakteryzująca się większą czasochłonnością, dlatego też została ona umieszczona w osobnym wątku, aby nie blokować głównego wątku przetwarzania serwisów DIM. Aby przeprowadzić tę konwersję, należało najpierw, wykorzystując funkcję split, zamienić tekst na wektor wartości tekstowych, a następnie, przy użyciu operacji rzutowania typów, zamienić go na wektor wartości liczbowych typu int. W kolejnym kroku należało, również przy użyciu rzutowania, zamienić wektor typu int na zmienną typu dyn_int. Główna różnica między tymi typami danych polega na dostępnych funkcjach wbudowanych oraz na sposobie indeksowania elementów: dla typu dyn_int indeksowanie rozpoczyna się od 1, natomiast dla standardowego wektora, od 0. Dodatkowo, typ dyn_int jest wspierany jako typ danych dla datapoint'u.

Gdy dane były zgromadzone i odpowiednio przetworzone, należało stworzyć panel użytkownika dedykowany zarządzaniu histogramami. W tym celu konieczne było wierne odtworzenie wyglądu i funkcjonalności aplikacji HistogramReader. Do implementacji logiki obsługi naciśnięcia poszczególnych przycisków wykorzystano komendy sterujące

opisane wyżej. W celu pobierania aktualnego statusu działania procesu histogramowania, utworzono dodatkowe datapointy, które pozwalały na sprawdzanie, która płyta elektroniki ma w danym momencie włączony tryb automatycznego odczytu (*autoread*). Do sprawdzania innych parametrów, takich jak informacja o tym, czy histogramowanie jest włączone na elektronice, czy aktywny jest filtr BCID, wykorzystano datapointy utworzone wcześniej dla standardowych parametrów elektroniki, zwracanych na serwisie STATUS.

W momencie gdy wyżej opisany panel dla histogramowania został ukończony i przetestowany, możliwe było przejście do tworzenia właściwego panelu odpowiedzialnego za wyświetlanie histogramów. W założeniach projektowych przyjęto, że funkcjonalności nowego systemu powinny być takie same lub większe niż te oferowane przez dotychczasową kombinację aplikacji Control Server i HistogramReader. Platforma WinCC OA jednak nie oferuje natywnie opcji wyświetlania wykresów histogramów wraz ze wszystkimi zaawansowanymi funkcjonalnościami, które dostarcza biblioteka QT5. Po licznych próbach implementacji z wykorzystaniem standardowych komponentów oraz po skontaktowaniu się z osobami odpowiedzialnymi za rozwój systemu WinCC OA, podjęto decyzję o wykorzystaniu możliwości integracji biblioteki QT5 bezpośrednio w środowisku WinCC OA. Jest to możliwe dzięki zastosowaniu komponentu o nazwie EWO (External Widget Object) [47]. Dzięki wykorzystaniu podstawowych skryptów, dostarczanych wraz z plikami instalacyjnymi WinCC OA, możliwe jest utworzenie szkicu panelu EWO do dalszego rozwoju. Następnie, gdy taka struktura kodu jest dostępna, można go swobodnie rozwijać o kolejne klasy i funkcjonalności. Tym samym, wykorzystując ten sam pakiet graficzny, który był używany przez aplikację HistogramReader, to jest QCustomPlot [48], możliwe było stworzenie w pełni funkcjonalnych wykresów o tych samych możliwościach interakcji.

Następnie skompilowany kod obiektu EWO, zapisany został w pliku z rozszerzeniem .dll, należało zmienić rozszerzenie na plik z rozszerzeniem .ewo, a potem przenieść go do odpowiedniego folderu źródłowego systemu WinCC OA. Dzięki temu możliwe stało się wykorzystywanie tego niestandardowego obiektu jako komponentu wbudowanego w system. Należało w tym celu wybrać ikonę EWO z listy dostępnych komponentów w trybie edycji panelu, a następnie wskazać odpowiedni plik .ewo. Gdy ta operacja została wykonana i obiekt ten został poprawnie umieszczony w docelowym panelu, sprawdzono komunikację pomiędzy EWO a WinCC OA. Po pomyślnym zakończeniu tego etapu, możliwe było przejście do weryfikacji działania całości interfejsu aplikacji HistogramReader w WinCC OA. Po pomyślnym przejściu powyższych testów, dodano również kilka nowych funkcjonalności w stosunku do oryginalnej aplikacji HistogramReader. Były to między innymi przyciski:

- **Save all...** – Przycisk pozwalający na jednoczesny zapis wszystkich aktywnych histogramów typu Time oraz Charge (ADC0, ADC1) w jednym, wybranym przez użytkow-

nika folderze, z automatycznym tworzeniem podfolderów TIME i ADC. Dane z każdej płyty PM zapisywane są w obu podfolderach w postaci plików w formacie .csv.

- **Load...** – Przycisk pozwalający na odczyt i załadowanie do wizualizacji histogramów z wcześniej utworzonego folderu, zapisanego za pomocą funkcji `Save all...`

Wszystkie wyżej wymienione funkcjonalności zostały szczegółowo zweryfikowane i sprawdzone, a wszystkie błędy, które pojawiły się podczas procesu integracji i tworzenia głównego interfejsu, zostały rozwiązane.

Jak wspomniano wcześniej, w systemie bazującym na aplikacji Control Server brakowało możliwości automatyzacji wielu procesów, a które teraz mogły zostać zaimplementowane w systemie ALFRED. Jednym z nich było stworzenie dedykowanego panelu umożliwiającego automatyczne wgrywanie konfiguracji zarówno dla elektroniki, jak i dla systemu wysokiego napięcia, a następnie automatyczne zbieranie danych z histogramów dla różnych wartości zmian wysokiego napięcia, zgodnie z zadaniem przez użytkownika minimalnym i maksymalnym zakresem oraz określonym krokiem zmian. Po uruchomieniu tego panelu, użytkownik w pierwszej kolejności musiał dodać konfiguracje elektroniki, które były pobierane dynamicznie z bazy danych poprzez system FRED. Następnie należało wybrać odpowiednie konfiguracje wysokiego napięcia, które były pobierane z pamięci podręcznej cache w systemie WinCC OA. Jest to opcja, którą oferuje pakiet JCOP, aby zapisywać konfiguracje w bazie danych Oracle lub właśnie w pamięci podręcznej. Obie listy musiały zawierać tę samą liczbę konfiguracji, aby możliwe było rozpoczęcie procedury automatycznego skanowania. W kolejnym kroku użytkownik musiał uzupełnić pola definiujące minimalny i maksymalny zakres zmian napięcia oraz wartość kroku. Tym samym, dla każdej zdefiniowanej kombinacji konfiguracji elektroniki i wysokiego napięcia, były one najpierw aplikowane na detektorze. Dopiero gdy napięcie się ustabilizowało zmieniane było wysokie napięcie przy użyciu wartości podanych przez eksperta. Proces był rozpoczynany od dodania do wartości bazowej minimalnego zakresu (który zgodnie z założeniami powinien być wartością ujemną), a z każdą kolejną iteracją do wartości wysokiego napięcia na każdym kanale dodawany był zdefiniowany krok. Dla przykładu, gdy użytkownik podał następujące wartości: minimalny zakres = -10, maksymalny zakres = 10, krok = 5, po zaimplementowaniu wybranej konfiguracji napięcie na kanałach było najpierw obniżane o 10V, następnie zbierane i zapisywane były dane z histogramów, a potem napięcie było zwiększane o 5V i dane były również zbierane i zapisywane. Ta procedura była powtarzana łącznie 5 razy, tak jak jest to pokazane na wyrażeniu poniżej.

$$(10 - (-10))/5 + 1 = 5$$

Takie rozwiązanie pozwoliło w znaczący sposób ograniczyć możliwość wystąpienia potencjalnego błędu ludzkiego, w sytuacji gdy ekspert musiał ręcznie modyfikować wartości wysokiego napięcia dla kolejnych kanałów kraty CAEN. Pozwoliło to również na pełną automatyzację procedury samego zapisu danych z histogramów do plików w formacie

.csv. Jest to ta sama funkcjonalność, która została zaimplementowana na panelu aplikacji HistogramReader jako przycisk Save All... Pozwala ona na jednoczesny zapis do różnych plików danych ze wszystkich aktywnych płyt PM, z zachowaniem podziału na kategorie TIME i CHARGE.

Gdy wyżej opisane narzędzie zostało pomyślnie przetestowane i sprawdzone na detektorze, kolejnym etapem prac stała się implementacja dedykowanego okna kalibracji elektroniki. Funkcjonalność ta, pochodząca z aplikacji HistogramReader, miała na celu umożliwienie precyzyjnego dostrajania kluczowych parametrów pracy elektroniki, takich jak TIME_ALIGNMENT, ADC_DELAY oraz CFD_ZERO. Proces kalibracji składał się z kilkunastu etapów, wymagał uprzedniego zaaplikowania odpowiedniej konfiguracji laserowej dla systemu elektroniki odczytowej oraz aktywacji kraty wysokiego napięcia. Następnie, wspomniane parametry były iteracyjnie modyfikowane aż do osiągnięcia odpowiednich wartości spełniających kryteria. Tym samym w kodzie aplikacji HistogramReader umieszczonych było wiele wykorzystywanych algorytmów, a poziom ich zrozumienia nie był niski. Z tego też powodu implementacja tej części w systemie ALFRED wymagała odłożenia w czasie. Gdy poziom rozwoju systemu już na to pozwalał, kluczowym krokiem stało się przeniesienie tych algorytmów do warstwy systemu FRED. Proces ten wymagał ścisłej współpracy z zespołem odpowiedzialnym za implementację tej części logicznej aplikacji w celu ustalenia formatu wymiany danych. Dopiero po zakończeniu tych prac przygotowawczych można było przystąpić do właściwej implementacji funkcjonalności w środowisku WinCC OA.

Prace implementacyjne rozpoczęto od dodania wymaganych datapoint'ów w systemie WinCC OA, przeznaczonych do przechowywania:

- Danych diagnostycznych (log'ów) z podziałem na poszczególne kanały elektroniki oraz kategorie: errors (błędy), warnings (ostrzeżenia) i infos (informacje).
- Danych do generowania wykresów dla procesu kalibracji parametru CFD_ZERO.
- Danych dla wykresu tłumika lasera (*Attenuator*) wykorzystywanego podczas kalibracji CFD_ZERO.

Po utworzeniu struktur danych, kolejnym krokiem było nawiązanie komunikacji ze stworzonym dla tych operacji serwisem DIM. Zaimplementowano również dedykowanego manager'a, odpowiedzialnego za przetwarzanie napływających danych i ich odpowiednie mapowanie na wcześniej zdefiniowane datapoint'y. Na tym etapie kluczowe było dogłębne zrozumienie formatu i struktury przesyłanych danych, aby zapewnić ich poprawne przetwarzanie i ewentualną transformację przed zapisem. Następnie przystąpiono do tworzenia interfejsu użytkownika w systemie WinCC OA. Przy wykorzystaniu standardowych komponentów dostępnych w WinCC OA, stworzono panel na bazie ekranu początkowego zakładki „Calibration” z aplikacji HistogramReader.

Jednak to prezentacja wykresów jest główną funkcjonalnością z perspektywy użytkownika podczas kalibracji parametru CFD_ZERO. Są to wykresy trójwymiarowe (3D), na

których intensywność sygnału jest reprezentowana za pomocą skali kolorów. Dodatkowo, wykres tłumika lasera (*Attenuator*) jest wykresem punktowym. Ze względu na specyfikę tych wizualizacji oraz ograniczenia standardowych komponentów WinCC OA, zdecydowano o przeniesieniu tej części panelu do dedykowanego obiektu EWO. Wymagało to, analogicznie do wcześniejszych prac, wykorzystania skryptu do generowania struktury panelu EWO, a następnie zaimplementowania interfejsu użytkownika zgodnie z wzorcem z aplikacji HistogramReader. Komunikacja pomiędzy obiektem EWO a systemem WinCC OA została zrealizowana za pomocą funkcji `invokeMethod`. Poniżej przedstawiono fragment kodu metody odpowiedzialnej za odbieranie danych dla wykresu tłumika oraz czyszczenie wykresu:

```
QVariant CalibrationsCFDZero::invokeMethod(const QString &name,
QList<QVariant> &values,
QString &error)
{
    else if (name == "setValuesAttenuator") {
        QString value = values[0].toString();

        QVector<QVector<uint>> result;
        QStringList rows = value.split('\n', Qt::SkipEmptyParts);
        for (const QString& row : rows) {
            QStringList valuesList = row.split(',', Qt::SkipEmptyParts);
            if (valuesList.size() != 3) {
                error = "Invalid number of values in row: " + row;
                qWarning() << error;
                return QVariant();
            }
            baseWidget->addPoint(valuesList[0].toDouble(),
                                valuesList[1].toDouble(),
                                valuesList[2]=="1");
        }
        baseWidget->replot();
        return QVariant();
    }
    else if (name == "clearValues") {
        baseWidget->clear();
        baseWidget->replot();
        return QVariant();
    }
    return QVariant();
}
```

Listing 2. Fragment metody `invokeMethod` obiektu EWO dla kalibracji CFD_ZERO

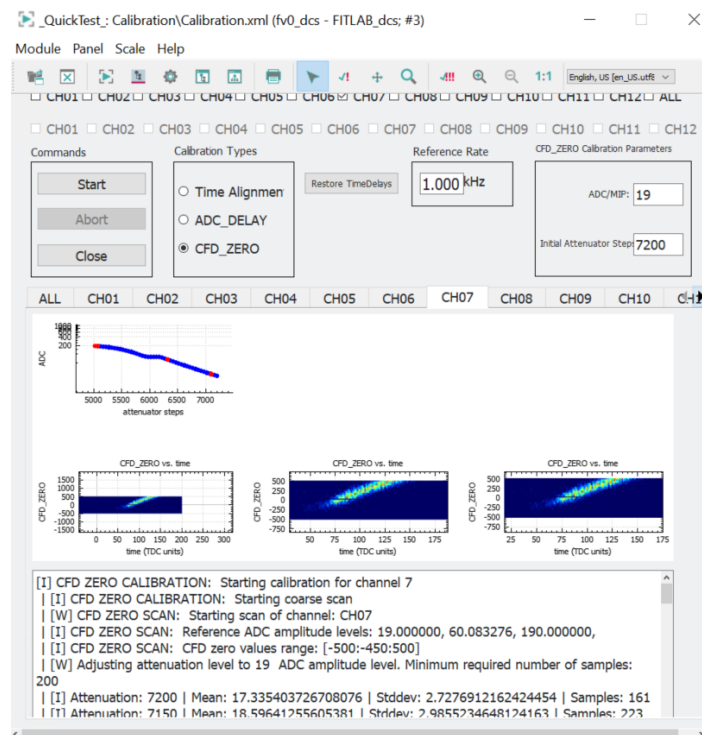
Po skompilowaniu obiektu EWO i zmianie rozszerzenia na `.ewo`, plik przeniesiono do odpowiedniego folderu źródłowego systemu WinCC OA. Umożliwiło to umieszczenie niestandardowego komponentu w panelu kalibracji.

Ostatnim elementem do zaimplementowania był panel wyświetlający log'i informujące o przebiegu procesu kalibracji z systemu FRED. Początkowa koncepcja zakładała wykorzystanie standardowego, bazowego pola tekstowego oferowanego przez system SCADA. Niestety, takie rozwiązanie okazało się nieefektywne – każda nowa iteracja i pojawienie się nowych danych wymagało odświeżenia całej zawartości pola tekstowego, a nie jedynie do-

dania nowej linii. Powodowało to zauważalne migotanie ekranu w obszarze wyświetlania log'ów, szczególnie pod koniec kalibracji, gdy zebrało się dużo linii tekstu.

Aby wyeliminować ten problem, stworzono dodatkowy, osobny interfejs EWO, zawierający wyłącznie pole tekstowe i dedykowane mechanizmy komunikacji z WinCC OA. Kluczową zaletą tego podejścia była możliwość przekazywania pojedynczych linii tekstu, co eliminowało konieczność odświeżania całego tekstu w stałych interwałach czasowych. Podobnie jak poprzedni obiekt EWO, również ten komponent został skompilowany, a jego rozszerzenie zmienione, po czym został przeniesiony do plików systemu WinCC OA.

Implementacja opisanych powyżej komponentów pozwoliła na stworzenie i umieszczenie paneli oraz nawiązanie pełnej komunikacji z logiką systemu FRED. Stworzony w ten sposób panel kalibracji w systemie ALFRED-WinCC OA w pełni odzwierciedla działanie z aplikacji HistogramReader, dostarczając ekspertom zintegrowane narzędzie do kalibracji elektroniki detektora. Poniżej na rysunku 6.9 znajduje się wyżej opisany panel kalibracji.



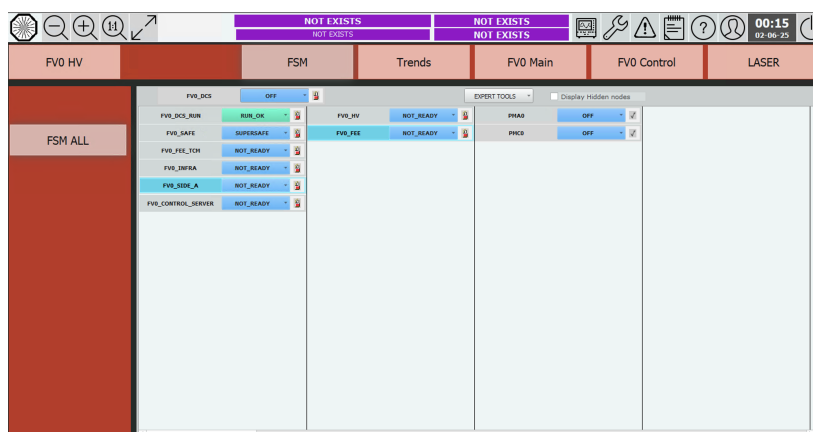
Rysunek 6.9. Panel kalibracji systemu elektroniki odczytowej dla parametru CFZ_ZERO

6.2. Integracja systemu w laboratorium ALICE-FIT

W laboratorium detektora ALICE-FIT system SCADA oparty na WinCC OA mógł być uruchamiany na dowolnym urządzeniu wyposażonym w system Windows oraz zainstalowaną platformę WinCC OA. Pojawił się jednak problem operacyjny związany z ograniczeniem serwera DIM, który pozwalał na połączenie tylko jednego komputera w roli klienta dla systemu FRED w danym momencie. W konsekwencji, gdy prace wymagały jednoczesnego dostępu z kilku stanowisk lub gdy inna osoba potrzebowała skorzystać z systemu SCADA,

konieczne było wylogowanie aktywnego użytkownika za pomocą narzędzia WebDID. W celu usprawnienia pracy, zespół Central ALICE DCS udostępnił jeden z komputerów dla rozwoju systemu DCS dla detektorów FIT. Na tej maszynie uruchomiony został projekt WinCC OA, a następnie nawiązano komunikację z kratą VME w laboratorium, w której umieszczone są moduł TCM oraz płyty PM. Połączenie to zrealizowano z wykorzystaniem protokołu OPC UA, co umożliwiło zdalne zarządzanie zasilaniem tego urządzenia oraz odczyt jego parametrów. Kolejnym krokiem było ustanowienie stałego połączenia z serwerem DIM dla systemu FRED, działającego w laboratorium. Dzięki temu, że projekt został uruchomiony na stałe i połączenie z tej maszyny nie było przerywane, rozwój systemu oraz jego weryfikację mogła przeprowadzać więcej niż jedna osoba.

W kolejnych etapach prac przystąpiono do konfiguracji maszyny stanu detektora, FSM. Choć system ten jest centralnie zarządzany przez zespół Central ALICE DCS, istnieje możliwość jego lokalnego utworzenia i uruchomienia z wykorzystaniem systemu DIM. Taka instancja FSM została uruchomiona w laboratorium i poddana odpowiedniej konfiguracji. Konieczne było zaktualizowanie listy płyt PM tak, aby odpowiadała faktycznie uruchomionym jednostkom, oraz wprowadzenie modyfikacji w funkcji odpowiedzialnej za weryfikację poprawności ich działania. Modyfikacja ta była podyktowana specyfiką środowiska laboratoryjnego, gdzie niektóre parametry zwracane przez protokół GBT, dotyczące statusu połączenia, sygnalizują błędy. Dla celów laboratoryjnych, gdzie nie prowadzi się akwizycji danych do późniejszej analizy fizycznej, tego typu błędy nie są uznawane za krytyczne. Należało także skonfigurować nowy obiekt kraty VME oraz nawiązać z nią połączenie poprzez OPC UA. Wcześniej zostało to skonfigurowane ręcznie w celach weryfikacji połączenia. W tym celu w oknie edycji urządzeń połączonych z systemem, należało dodać nowe urządzenie, które wcześniej podczas nawiązywania połączenia poprzez OPC UA zostało nazwane FITFEElab. Następnie zweryfikowano połączenie datapoint'ów dla skonfigurowanego urządzenia kraty VME. Uruchomienie tak skonfigurowanego systemu FSM umożliwiło częściowe testowanie również tego segmentu całego projektu w warunkach laboratoryjnych, jest on ukazany na rys. 6.10.



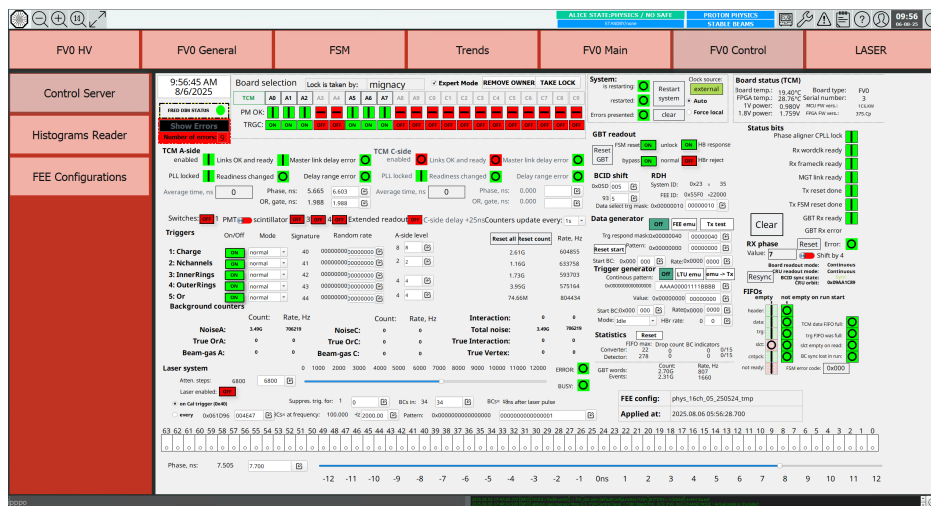
Rysunek 6.10. Panel systemu FSM wraz z kategorią odpowiedzialną za stan płyt PM

6.3. Integracja Systemu dla detektora FV0

Proces integracji systemów ALFRED oraz WinCC OA w sieci wewnętrznej AliDCS dla detektora FV0 został rozpoczęty od otrzymania dedykowanych komputerów dla systemów AlfIPbus i FRED, a także niezbędnej bazy danych Oracle dla systemu FRED. Podjęto decyzję o umieszczeniu systemu WinCC OA na tym samym komputerze Worker Node, na którym operował poprzedni projekt wraz z aplikacją Control Server. Pozwoliło to znacząco uprościć proces przełączania pomiędzy projektami, eliminując konieczność ponownej konfiguracji dla komunikacji z systemami zewnętrznymi oraz urządzeniami, która realizowana była między innymi z wykorzystaniem protokołu OPC UA.

Gdy stan rozwoju systemów osiągnął odpowiedni poziom do rozpoczęcia prac na detektorze FV0, dokonano przeniesienia projektu WinCC OA wraz z wymaganymi pakietami. Systemy AlfIPbus oraz FRED zostały również przeniesione, skonfigurowane oraz uruchomione w sieci wewnętrznej na wcześniej przygotowanych serwerach.

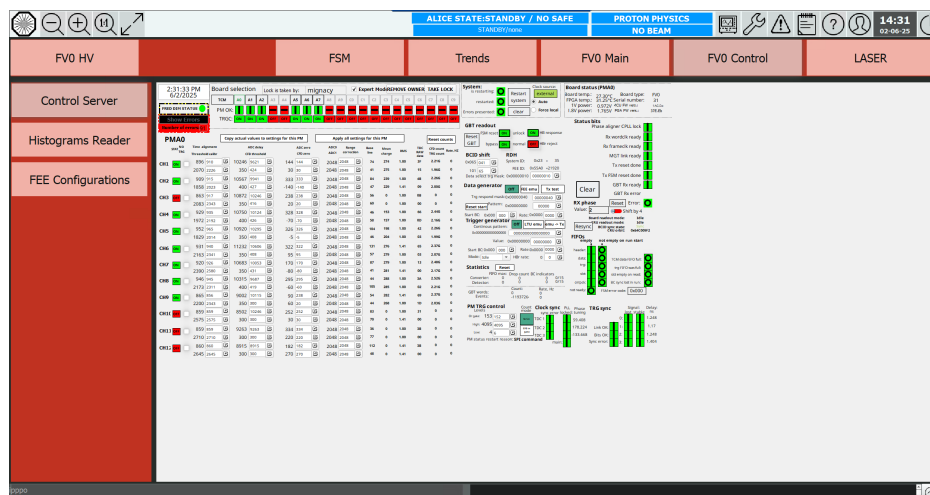
Po pomyślnej konfiguracji wszystkich komponentów, ALFIPbus i FRED zostały uruchomione i gdy działały poprawnie, projekt w systemie WinCC OA został uruchomiony. Nawiązano komunikację z systemami centralnymi eksperymentu, urządzeniami zewnętrznymi oraz z serwerem DIM. Pierwszym krokiem testowym było uruchomienie systemu elektroniki odczytowej detektora i gruntowne sprawdzenie całej ścieżki komunikacyjnej w systemie ALFRED. Poniżej ukazane są dwa zrzuty ekranu prezentujące działanie systemu SCADA wraz z systemem ALFRED.



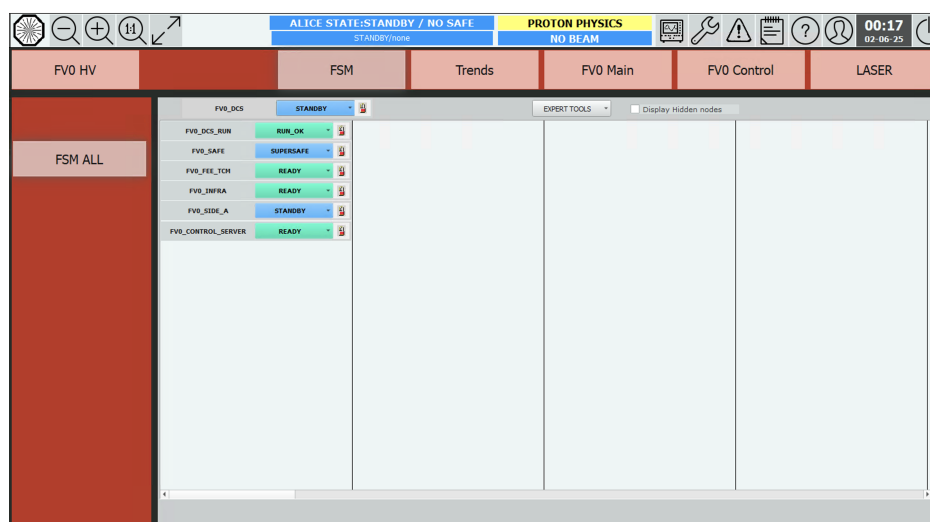
Rysunek 6.11. Ekran płyty TCM dla detektora FV0

Pozytywna weryfikacja tego etapu pozwoliła na przejście do rozwiązywania zidentyfikowanych problemów.

Jednym z głównych wyzwań okazała się konfiguracja systemu FSM dla detektora FV0 w docelowej sieci AliDCS. Na rysunku 6.13 ukazany jest ekran tego systemu.



Rysunek 6.12. Ekran płyty PM dla detektora FV0



Rysunek 6.13. System FSM dla detektora FV0

Pełne przetestowanie i zweryfikowanie działania FSM w środowisku laboratoryjnym było niemożliwe ze względu na istotne różnice sprzętowe: w laboratorium dostępne były jedynie dwie płyty PM, podczas gdy w detektorze FV0 zainstalowanych jest ich sześć. Dodatkowo, krata wysokiego napięcia CAEN w detektorze FV0 posiadała całkowicie inną kombinację kanałów i płyt. Te rozbieżności wymuszały wprowadzenie znaczących zmian w konfiguracji FSM przygotowanej dla laboratorium, co skutkowało odmiennym zachowaniem systemu.

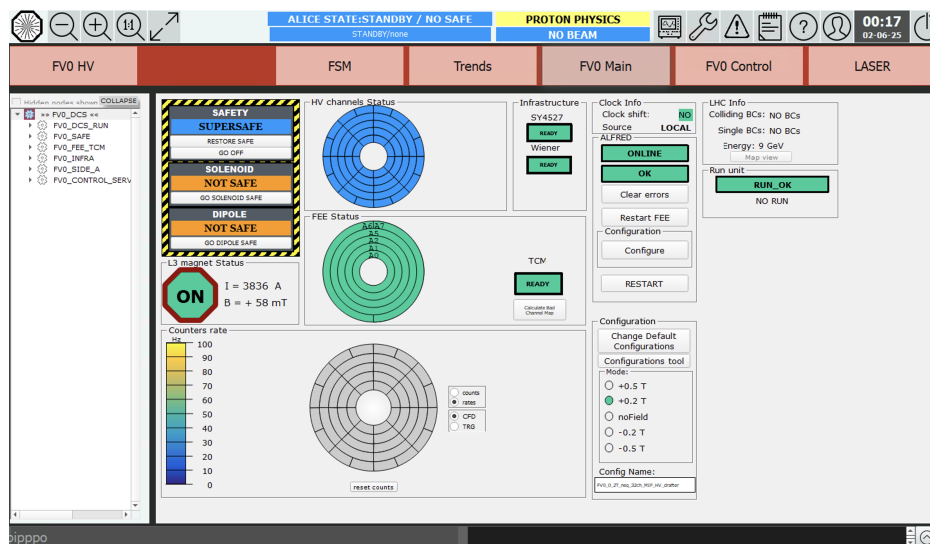
Pierwsze zmiany w FSM objęły dostosowanie części odpowiedzialnej za komunikację z aplikacją Control Server oraz mechanizmów sprawdzania jej statusu. Celem stało się zapewnienie, aby ten moduł FSM współpracował z nowym systemem ALFRED. Według nowej implementacji, status działania kraty VME weryfikowany jest poprzez zewnętrznego manager'a, który oblicza czas, jaki upłynął od ostatniej odpowiedzi na jednym z serwisów DIM o nazwach STATUS lub COUNTER_RATES. Manager ten został uruchomiony jako

osobny wątek w projekcie WinCC OA. Skonfigurowano odpowiednie datapoint'y, a za pomocą funkcji *datapoint function*, kopiując informacje z datapoint'ów używanych do komunikacji z ALFRED, zminimalizowano modyfikacje w istniejącym kodzie FSM oraz dzięki temu istniała opcja ewentualnego powrotu do aplikacji Control Server w przypadku wystąpienia poważniejszych komplikacji.

Po ustanowieniu komunikacji pomiędzy FSM a systemem ALFRED, przystąpiono do dalszego rozwoju tej części projektu. Konieczne było zmodyfikowanie procedur wykonywanych podczas zmiany stanów detektora, głównie podczas przejść pomiędzy stanami OFF ↔ STANDBY oraz STANDBY ↔ READY. Początkowo funkcje te zostały zmodyfikowane tak, aby wykonywały te same operacje co dla Control Server, lecz z wykorzystaniem systemu ALFRED. Jednakże takie podejście prowadziło do problemów: detektor potrzebował różnych okresów czasu na załączenie elektroniki, co bardzo często skutkowało błędami związanymi z aplikowaniem konfiguracji i czyszczeniem błędów po uruchomieniu. Po wielu próbach i modyfikacjach, wspólnie z zespołem podjęto decyzję o przeniesieniu tej części logiki do nowego serwisu DIM w systemie FRED o nazwie MANAGER. Serwis ten umożliwiał uruchomienie systemu FRED wraz z podaniem nazwy konfiguracji, która miała zostać zaaplikowana. Pozwalał również na jego „uśpienie”, jednak tutaj operacja była wykonywana także przez zewnętrznego manager'a na wypadek, gdyby wyłączenie nastąpiło w sposób niekontrolowany (np. podczas awarii, braku zasilania). Po zaimplementowaniu tej funkcjonalności, odpowiednio skonfigurowano komunikację DIM oraz zmodyfikowano kod FSM odpowiedzialny za konfigurację elektroniki po jej uruchomieniu.

Po zakończeniu implementacji tej części systemu, przeprowadzono weryfikację całości kodu, włączając w to fragmenty odpowiedzialne za komunikację z kratą wysokiego napięcia. Następnie, podczas testów z rzeczywistym detektorem FV0, zweryfikowano poprawność działania całego systemu.

Prace nad zarządzaniem stanem detektora nie zakończyły się jednak w tym miejscu. Należało także zaimplementować część systemu odpowiedzialną za stan bezpieczeństwa dla detektora FV0. Jest to komponent powiązany z FSM, lecz działający niezależnie i równolegle. Wymagało to wprowadzenia zmian w operacjach zmieniających stan bezpieczeństwa detektora, takich jak przejścia pomiędzy stanami SUPERSAFE – SAFE – NOT SAFE. Dzięki doświadczeniu zdobytemu podczas pracy z systemem FSM, sprawnie zaimplementowano operacje Restore From Magnet SAFE oraz Restore From Dipole SAFE, dostosowując je do komunikacji z systemem ALFRED. Również w tym przypadku przeprowadzono testy działania systemu na detektorze. Poniżej na rys. 6.14 znajduje się ekran główny projektu SCADA wraz z systemem DSS.



Rysunek 6.14. Panel główny systemu SCADA dla detektora FV0

Podczas operacji Restore From Magnet SAFE ujawnił się specyficzny błąd polegający na braku wyczyszczenia flagi błędu na niektórych płytach PM po zaaplikowaniu konfiguracji. Problem ten wymagał licznych konsultacji z ekspertami. Ustalono, że błąd ten był powodowany błędnym czasem oczekiwania na wyczyszczenie błędów po wgraniu konfiguracji, operacja była wykonywana zbyt szybko.

W toku dalszych prac, wspólnie z zespołem, podjęto dodatkowe decyzje o zwiększeniu ilości informacji przekazywanych z systemu WinCC OA do systemu FRED, aby móc rozwiązywać problemy jak powyższy. Jedną z takich operacji było wysyłanie komendy START_OF_RUN podczas rozpoczynania zbierania danych w eksperymencie ALICE. W tym celu ponownie wykorzystano serwis MANAGER, do którego system WinCC OA przesyłał informacje o aktualnej konfiguracji dla danego natężenia pola magnetycznego solenoidu ALICE.

Podczas intensywnego użytkowania i testowania systemu przez ekspertów pojawiły się cenne sugestie dotyczące możliwych poprawek oraz wprowadzenia nowych funkcjonalności.

W pierwszej kolejności skupiono się na eliminacji zidentyfikowanych błędów. Jednym z nich było nieprawidłowe działanie przycisków Copy Actual Values oraz Apply Actual Values w interfejsie użytkownika. Przyciski te powinny kopiować i implementować informacje dla wszystkich pól tekstowych danej płyty PM, a nie, jak miało to miejsce, jedynie dla parametrów przechowywanych w zmiennych dla kolejnych kanałów na tej płycie.

Zasugerowano również wprowadzenie mechanizmu ograniczającego możliwość modyfikacji parametrów w ekranach aplikacji Control Server oraz HistogramsReader przez więcej niż jednego użytkownika jednocześnie. Miało to na celu:

- Na ekranie aplikacji HistogramsReader: ograniczenie błędów, które mogłyby pojawić

się w części histogramów, gdyby dwóch użytkowników próbowało odczytywać dane z różnych płyt PM (system FRED nie był przystosowany do takiej pracy jednocześnie).

- Na ekranie aplikacji Control Server: uniknięcie nieporozumień, gdy więcej niż jedna osoba mogłaby wprowadzać zmiany, potencjalnie psując pracę innych.

W odpowiedzi na te sugestie, utworzono dwa datapoint'y zawierające w nazwie słowo Lock, przechowujące nazwę użytkownika, który aktualnie posiada dostęp do edycji, można było założyć, że jest to coś podobnego do Locka z FSM. Nazwa zalogowanego użytkownika systemu operacyjnego pobierana jest w WinCC OA za pomocą funkcji `getOSUser()`. Gdy użytkownik nie posiada aktywnej blokady:

- Na ekranie aplikacji Control Server: ma on podgląd do wszystkich płyt PM oraz płyty TCM, jednak z zablokowaną możliwością wprowadzania jakichkolwiek zmian.
- Na ekranie aplikacji HistogramsReader: widzi dokładnie ten sam ekran i te same dane, które wyświetla użytkownik aktualnie posiadający blokadę.

W systemie brakowało jeszcze mechanizmu ukazania uśpienia systemu FRED w sytuacji, gdy system elektroniczny detektora zostaje wyłączony, Ekran aplikacji Control Server miał również przechodzić do stanu uśpienia, aby było to widoczne dla użytkownika. Podczas implementacji tej funkcjonalności w projekcie WinCC OA został stworzony dedykowany Manager, ustawiony w trybie ciągłego działania (tryb *Always*). Manager ten, wykorzystując funkcję `dpConnect`, monitorował datapoint'y odpowiedzialne za stan kraty VME oraz systemu FRED. Po spełnieniu odpowiednich warunków, sygnalizujących wyłączenie elektroniki, wysyłana była komenda DIM STOP na serwis MANAGER systemu FRED.

Gdy system był używany i testowany, zidentyfikowano problem nadmiernego zużycia zasobów przez panel odpowiedzialny za wyświetlanie histogramów oraz przez funkcje przetwarzające dane napływające z systemu FRED. Chociaż automatyczne pobieranie danych histogramów można zatrzymać za pomocą przycisku `autoread`, eksperci często o tym zapominali, co prowadziło do niepotrzebnego odczytu i przetwarzania danych przez wiele godzin. Aby zaradzić tej sytuacji, opcja ta została zablokowana i połączona z operacją histogramowania w systemie elektroniki odczytowej. Tym samym gdy histogramowanie było włączone, rozpoczynał się odczyt danych, a następnie gdy system elektroniczny zakończył zbieranie danych, proces odczytu też był zatrzymywany. Zostało to zaimplementowane na poziomie projektu WinCC OA.

Kolejnym istotnym etapem prac związanym z integracją systemu SCADA dla detektora FV0 była konfiguracja archiwizacji danych do centralnej bazy danych, zarządzanej przez zespół Central ALICE DCS. System WinCC OA natywnie wspiera funkcjonalność archiwizacji danych do skonfigurowanej bazy, a baza Oracle może zostać także połączona z wykorzystaniem framework'u JCOP. Dla projektu FV0 połączenie z docelową bazą danych zostało nawiązane już wcześniej przez osoby odpowiedzialne za rozwój tego projektu z użyciem aplikacji Control Server. Dotychczas archiwizowane dane obejmowały przede wszystkim wartości *Rate'ów* dla płyt PM, temperatury płyt w modułach FEE oraz wybrane

statusy i stan działania kraty wysokiego napięcia CAEN. Głównym celem rozbudowy listy przechowywanych parametrów była integracja z systemem Grafana, który umożliwia monitorowanie publikowanych informacji spoza sieci wewnętrznej AliDCS oraz spoza sieci GPN CERN. W tym celu wyeksportowano listę nazw potencjalnych parametrów do archiwizacji w celu konsultacji z ekspertem detektora, który dokonał selekcji kluczowych danych. Następnie zaproponowano różne typy odświeżania informacji w bazie danych (Smoothing), takie jak: porównanie starej i nowej wartości (*Old/new comparison*), porównanie starej i nowej wartości z uwzględnieniem czasu (*Old/new comparison AND Time*), zmiana wartości (*Value changed* – z możliwością zdefiniowania progu zmiany relatywnej lub absolutnej) oraz zmiana wartości z uwzględnieniem czasu (*Value changed AND TIME*). Po ponownej konsultacji i zatwierdzeniu przygotowanej listy parametrów oraz ich konfiguracji przez zespół centralny, można było przystąpić do implementacji mechanizmów archiwizacji w projekcie WinCC OA.

Część z wybranych parametrów, w szczególności te dotyczące kolejnych kanałów dla płyt PM, została skonfigurowana do archiwizacji z wykorzystaniem funkcji *Mass Parameters*, oferowanej przez framework JCOP. Parametry te zostały wcześniej zdefiniowane jako typ urządzenia w projekcie WinCC OA przez osoby pierwotnie tworzące ten projekt. Pozostałe parametry nie zostały skonfigurowane w ten sam, zautomatyzowany sposób, ponieważ wprowadzanie zmian w istniejących typach urządzeń w trakcie działania projektu i aktywnego zbierania danych mogłoby prowadzić do niepożądanych komplikacji na poziomie globalnym eksperymentu. Z tego względu archiwizacja dla tych parametrów została ustawiona ręcznie, poprzez indywidualne odnalezienie odpowiednich datapoint'ów w oknie konfiguracyjnym *Para* systemu WinCC OA.

7. Podsumowanie

Przedmiotem niniejszej pracy była kompleksowa implementacja i wdrożenie platformy ALFRED, stanowiącej modernizację systemu kontroli i sterowania dla detektorów ALICE-FIT w eksperymencie ALICE w CERN. W ramach pracy, autor, we współpracy z zespołem ALICE-FIT DCS, wniósł kluczowy wkład w rozwój i integrację nowej architektury systemu DCS. Prace te zaowocowały publikacjami naukowymi, których współautorem jest autor niniejszej pracy.

Głównym zadaniem była zmiana systemu z dotychczasowych, dedykowanych aplikacji (Control Server, HistogramReader) do ustandaryzowanego systemu ALFRED, opartego na warstwach FRED i ALF. Kluczowym wyzwaniem była adaptacja tej architektury do specyfiki detektorów FIT, które komunikują się z elektroniką FEE za pomocą protokołu IPbus, a nie standardowego GBT. Wymagało to rozwoju specjalistycznego komponentu ALFIPbus oraz rozbudowy systemu FRED, aby przenieść i zintegrować całą logikę oprogramowania, w tym obsługę złożonych operacji na rejestrach, przetwarzanie danych i histogramów oraz zarządzanie konfiguracjami.

Równolegle, autor opracował w środowisku SIMATIC WinCC OA kompleksowy system SCADA. Aplikacja ta, stanowiąca nadrzędną warstwę nowej architektury, zapewnia pełną wizualizację i kontrolę nad systemem elektronicznym, zastępując wszystkie funkcje poprzednich narzędzi. Zaimplementowano zaawansowane panele do zarządzania konfiguracjami FEE w oparciu o bazę danych Oracle, odtworzono i rozszerzono funkcjonalność aplikacji HistogramReader, włączając w to wizualizację wykresów w obiektach EWO, oraz stworzono nowe, zautomatyzowane narzędzia do kalibracji parametrów detektora oraz zbierania danych.

Stworzony i zintegrowany system został pomyślnie wdrożony i przetestowany, najpierw w laboratorium ALICE-FIT, a następnie w docelowym środowisku na detektorze FV0. Wdrożenie to nie tylko ujednoliciło system sterowania ze standardami eksperymentu ALICE, ale także wyeliminowało problemy związane ze stabilnością poprzedniej wersji systemu i zwiększyło elastyczność całego oprogramowania. Docelowo, opracowana platforma ma zostać zaimplementowana do sterowania całym zespołem poddetektorów FIT, co przyczyni się do zwiększenia niezawodności i efektywności zbierania danych fizycznych w eksperymencie.

Bibliografia

- [1] CERN. "About CERN". (2025), adr.: <https://home.cern/about>.
- [2] *Convention for the establishment of a European organization for nuclear research: Paris, 1st July, 1953 : as amended. Convention pour l'établissement d'une Organisation européenne pour la Recherche nucléaire. Paris, le 1er juillet 1953 : telle qu'elle a été modifiée.* Geneva: CERN, 1971. adr.: <https://cds.cern.ch/record/330625>.
- [3] G. Aad i in., "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC", *Phys. Lett. B*, t. 716, s. 1–29, 2012. DOI: 10.1016/j.physletb.2012.08.020. arXiv: 1207.7214 [hep-ex].
- [4] T. J. Berners-Lee Sir., "Information Management: A Proposal", mar. 1989.
- [5] CERN. "CERN people". (2025), adr.: <https://home.cern/about/who-we-are/our-people>.
- [6] "LHC Machine", *JINST*, t. 3, L. Evans i P. Bryant, red., S08001, 2008. DOI: 10.1088/1748-0221/3/08/S08001.
- [7] L. Arnaudon i in., "Linac4 technical design report", grud. 2006.
- [8] K. Hanke, "Past and present operation of the CERN PS Booster", *Int. J. Mod. Phys. A*, t. 28, s. 1330019, 2013. DOI: 10.1142/S0217751X13300196.
- [9] D. J. Simon, "The CERN PS complex: a versatile particle factory", 1996. adr.: <https://cds.cern.ch/record/305985>.
- [10] CERN. "CERN Accelerator Complex". (2025), adr.: <https://home.cern/science/accelerators/accelerator-complex>.
- [11] D. (CERN). "LEIR OPERATIONS FOR THE LHC AND FUTURE PLANS". (), adr.: <https://cds.cern.ch/record/1702922/files/mopm2ha01.pdf>.
- [12] J. Coupard i in., "LHC Injectors Upgrade, Technical Design Report: v.2: Ions", kw. 2016. DOI: 10.17181/CERN.L6VM.UOMS.
- [13] I. PAN. "LHC – Wielki Zderzacz Hadronów (polski podręcznik)". (), adr.: https://ppss.ifj.edu.pl/materials_2016/LHC-booklet-Polish.pdf.
- [14] D. J. Simon, "The CERN PS complex: a versatile particle factory", *Conf. Proc. C*, t. 960610, S. Myers, A. Pacheco, R. Pascual, C. Petit-Jean-Genaz i J. Poole, red., s. 295–297, 1996.
- [15] T. C. accelerator complex. "CERN". (), adr.: <https://cds.cern.ch/record/2197559>.
- [16] "A vacuum as empty as interstellar space", 2012. adr.: <https://cds.cern.ch/record/1997379>.
- [17] G. Aad i in., "The ATLAS Experiment at the CERN Large Hadron Collider", *JINST*, t. 3, S08003, 2008. DOI: 10.1088/1748-0221/3/08/S08003.
- [18] S. Chatrchyan i in., "The CMS Experiment at the CERN LHC", *JINST*, t. 3, S08004, 2008. DOI: 10.1088/1748-0221/3/08/S08004.
- [19] A. A. Alves Jr. i in., "The LHCb Detector at the LHC", *JINST*, t. 3, S08005, 2008. DOI: 10.1088/1748-0221/3/08/S08005.

- [20] J. Adam i in., “Enhanced production of multi-strange hadrons in high-multiplicity proton-proton collisions”, *Nature Phys.*, t. 13, s. 535–539, 2017. DOI: 10 . 1038 / nphys41111. arXiv: 1606 . 07424 [nucl-ex].
- [21] W. H. Trzaska, “New Fast Interaction Trigger for ALICE”, *Nucl. Instrum. Meth. A*, t. 845, G. Badurek, T. Bergauer, M. Dragicevic i in., red., s. 463–466, 2017. DOI: 10 . 1016 / j . nima . 2016 . 06 . 029.
- [22] A. 2. detector systems. “ALICE (CERN)”. (), adr.: https://alice-collaboration.web.cern.ch/2024_ALICE_LS2_upgrade_paper.
- [23] K. Roslon, “ALICE Fast Interaction Trigger upgrade”, *Nucl. Instrum. Meth. A*, t. 1079, s. 170 609, 2025. DOI: 10 . 1016 / j . nima . 2025 . 170609. arXiv: 2503 . 05518 [hep-ex].
- [24] ALICE-FIT. “ALICE-FIT Detectors”. (), adr.: https://alice-collaboration.web.cern.ch/menu_proj_items/FIT.
- [25] C. Ghabrous Larrea, K. Harder, D. Newbold i in., “IPbus: a flexible Ethernet-based control system for xTCA hardware”, *JINST*, t. 10, nr. 02, s. C02019, 2015. DOI: 10 . 1088 / 1748 - 0221 / 10 / 02 / C02019.
- [26] P. Buncic, M. Krzewicki i P. Vande Vyvre, “Technical Design Report for the Upgrade of the Online-Offline Computing System”, kw. 2015.
- [27] “Upgrade of the ALICE Readout & Trigger System”, P. Antonioli, A. Kluge i W. Riegler, red., 2013.
- [28] Kvapil, Jakub, Bhasin, Anju, Bombara, Marek i in., “ALICE Central Trigger System for LHC Run 3”, *EPJ Web Conf.*, t. 251, s. 04 022, 2021. DOI: 10 . 1051 / epjconf / 202125104022. adr.: <https://doi.org/10.1051/epjconf/202125104022>.
- [29] P. P. Nikiel, B. Farnham, V. Filimonov i S. Schlenker, “Generic OPC UA Server Framework”, *J. Phys. Conf. Ser.*, t. 664, nr. 8, s. 082 039, 2015. DOI: 10 . 1088 / 1742 - 6596 / 664 / 8 / 082039.
- [30] M. Slupecki, “Fast Interaction Trigger for ALICE upgrade”, *Nucl. Instrum. Meth. A*, t. 1039, s. 167 021, 2022. DOI: 10 . 1016 / j . nima . 2022 . 167021.
- [31] C. Gaspar i M. Donszelmann, “DIM: A distributed information management system for the DELPHI experiment at CERN”, w *8th Real-Time Computer Applications in Nuclear, Particle and Plasma Physics (RT 93)*, czer. 1993, s. 156–158.
- [32] Siemens. “WinCC OA”. (), adr.: <https://www.winccoa.com>.
- [33] J. Mejía Camacho, S. Rodríguez Ramírez, J. Cabanillas Noris i in., “Forward Diffractive Detector control system for Run 3 in the ALICE experiment”, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, t. 1050, s. 168 146, 2023, ISSN: 0168-9002. DOI: <https://doi.org/10.1016/j.nima.2023.168146>. adr.: <https://www.sciencedirect.com/science/article/pii/S0168900223001365>.
- [34] M. Tkáčik, J. Jadlovský, S. Jadlovská, L. Koska, A. Jadlovská i M. Donadoni, “FRED—Flexible Framework for Frontend Electronics Control in ALICE Experiment

- at CERN”, *Processes*, t. 8, nr. 5, 2020, ISSN: 2227-9717. adr.: <https://www.mdpi.com/2227-9717/8/5/565>.
- [35] P. Moreira i in., “The GBT Project”, w *Topical Workshop on Electronics for Particle Physics*, CERN, 2009. DOI: 10.5170/CERN-2009-006.342.
 - [36] K. Roslon, “Integrating IPbus ALFRED into the ALICE-FIT setup”, sty. 2025. arXiv: 2501.04685 [physics.ins-det].
 - [37] W. Pierozak. “ALFIPbus”. (), adr.: <https://github.com/wpierozak/ALFIPbus>.
 - [38] L. R. Dalesio, J. O. Hill, M. Kraimer i in., “The Experimental Physics and Industrial Control System architecture: Past, present, and future”, *Nucl. Instrum. Meth. A*, t. 352, s. 179–184, 1994. DOI: 10.1016/0168-9002(94)91493-1.
 - [39] M. R. K. (N. Laboratory). “EPICS Input / Output Controller (IOC) Application Developer’s Guide”. (), adr.: <https://epics.anl.gov/EpicsDocumentation/AppDevManuals/AppDevGuide/3.12BookFiles/AppDevGuide.book.html>.
 - [40] J. O. Hill. “EPICS R3.14 Channel Access Reference Manual”. (), adr.: <https://epics.anl.gov/base/R3-14/12-docs/CAref.html>.
 - [41] TANGO. “TANGO Controls”. (), adr.: <https://www.tango-controls.org/>.
 - [42] J. M. Chaize, A. Gotz, W. D. Klotz, J. Meyer, M. Perez i E. Taurel, “TANGO - an object oriented control system based on CORBA”, *Conf. Proc. C*, t. 991004, D. Bulfone i A. Daneels, red., s. 475–479, 1999.
 - [43] CERN. “SCA Communication Understanding”. (), adr.: https://indico.cern.ch/event/504723/contributions/1182041/attachments/1237063/1816948/sca_communication.pdf.
 - [44] O. Holme, M. Gonzalez Berges, P. Golonka i S. Schmeling, “The JCOP framework”, *Conf. Proc. C*, t. 051010, WE2.1–60, 2005.
 - [45] T. C. Project. “The CentOS Project”. (), adr.: <https://www.centos.org/>.
 - [46] P. Kinasz, “Opracowanie w oprogramowaniu WinCC O.A. systemu SCADA (ang. Supervisory Control And Data Acquisition) do obsługi stacji laboratoryjnej pod-detektorów FIT (ang. Fast Interaction Trigger) w eksperymencie ALICE (ang. A Large Ion Collider Experiment)”, Praca inżynierska, Praca inżynierska, Politechnika Warszawska, 2025.
 - [47] Siemens. “EWO (External Widget Object)”. (), adr.: https://www.winccoa.com/documentation/WinCCOA/latest/en_US/Native_GEDI/EWO.html.
 - [48] E. Eichhammer. “QCustomPlot”. (), adr.: <https://www.qcustomplot.com>.

Wykaz symboli i skrótów

ADC – ang. *Analog Digital Converter*
ALF – ang. *ALICE Low FrontEnd*
ALFRED – ang. *ALICE Low Level FrontEnd Device*
ALICE – ang. *A Large Ion Collider Experiment*
ATLAS – ang. *A Toroidal LHC Apparatus*
BCID – ang. *Bunch Crossing ID*
CA – ang. *Channel Access*
CERN – fr. *Organisation Européenne pour la Recherche Nucléaire*
CMS – ang. *Compact Muon Solenoid*
CORBA – ang. *Common Object Request Broker Architecture*
CRU – ang. *Common Readout Unit*
CTP – ang. *Central Trigger Processor*
DCS – ang. *Detector Control System*
DIM – ang. *Distributed Information Management*
DNS – ang. *Domain Name Server*
EiTI – Wydział Elektroniki i Technik Informacyjnych
PW – Politechnika Warszawska
EPICS – ang. *Experimental Physics and Industrial System*
EMCal – ang. *Electromagnetic Calorimeter*
EWO – ang. *External Widget Object*
FEE – ang. *FrontEnd Electronics*
FIFO – ang. *First In First Out*
FIT – ang. *Fast Interaction Trigger*
FLP – ang. *First Level Processor*
FRED – ang. *Front-End Device*
FSM – ang. *Finite State Machine*
GBT – ang. *GigaBit Transceiver*
GPN – ang. *Global Private Network*
GUI – ang. *Graphical User Interface*
HV – ang. *High Voltage*
IC – ang. *Internal Communication*
IOC – ang. *Input Output Controller*
IP – ang. *Interaction Point*
ITS – ang. *Inner Tracking System*
JCOP – ang. *JOint Controls Project*
LHC – ang. *Large Hadron Collider*
LHCb – ang. *Large Hadron Collider beauty Experiment*
LINAC3 – ang. *Linear Accelerator 3*

LINAC4 – ang. *Linear Accelerator 4*
LTU – ang. *Local Trigger Unit*
MAPI – ang. *Message Application Interface*
MCP - PMT – ang. *Micro Channel Plate - Photomultiplier Tube*
OPC UA – ang. *Open Protocol Communication Unified Architecture*
PM – ang. *Process Module*
PS – ang. *Proton Synchrotron*
PSB – ang. *Proton Synchrotron Booster*
RF – ang. *Radio Frequency*
SCA – ang. *Slow Control Adapter*
SCADA – ang. *Supervisory Control and Data Acquisition*
SPS – ang. *Super Proton Synchrotron*
SWT – ang. *Single Word Transaction*
TCM – ang. *Trigger&Clock Module*
TPC – ang. *Time Projection Chamber*
TRD – ang. *Transition Radiation Detector*
UDP – ang. *User Datagram Protocol*
WN – ang. *Worker Node*
WWW – ang. *World Wide Web*

Spis rysunków

1.1	Schemat ukazujący cały kompleks akcelacyjny CERN [15]	7
1.2	Schemat multidetektora ALICE [22]	10
1.3	Ilustracja elementów detektora FIT z pokryciem pseudopospieszności oraz odległością od punktu interakcji [24]	11
1.4	System elektroniki odczytowej (FEE) detektora FV0 w eksperymencie ALICE .	12
1.5	Schemat centralnego systemu trigger'ów ukazujący wejścia do CTP (Central Trigger Processor) i typy połączeń między CTP i FEE (FrontEnd Electronics) detektora. [28]	13
2.1	Schemat architektury DCS FIT [30]	15
2.2	Ekran panelu TCM i PMA0 w aplikacji Control Server	16
2.3	Ekran zakładki Time i Charge dla płyty PMA0 w aplikacji HistogramReader . .	17
2.4	Ekran główny systemu SCADA	18
2.5	Ekran systemu FSM dla systemu SCADA	18
2.6	Schemat działania systemu DIM [31]	19
2.7	Schemat działania systemu ALFRED w celu komunikacji z FEE [36]	20
3.1	Schemat architektury systemu ALFRED dostosowanego pod FEE w detektorach ALICE-FIT [36]	21
4.1	Schemat wymiany danych z wykorzystaniem interfejsu MAPI [34]	25

4.2	Struktura systemu WinCC OA podzielonej na kolejne jednostki i managerów [32]	27
5.1	Zapis log'ów po uruchomieniu systemu FRED w laboratorium	28
5.2	Struktura ramki SWT w celu wymiany danych pomiędzy FRED i ALFIPbus w laboratorium ALICE-FIT	30
5.3	Podgląd danych w narzędziu WebDID dla jednego z rejestrów	31
5.4	Ręczna deklaracja serwisów na początku rozwoju systemu	31
5.5	Część serwisów oraz komend dla płyty PMA0	32
5.6	Schemat komunikacji pomiędzy systemami WinCC OA, FRED oraz ALF z wykorzystaniem serwera DIM [34]	33
5.7	Schemat bazy danych dla parametrów elektroniki	35
5.8	Wersja systemu SCADA stworzona w celach nawiązania komunikacji z systemem FRED, bazująca na ekranie głównym aplikacji Control Server	37
6.1	Część z utworzonych subparametrów dla datapoint'u TCM_FRED	41
6.2	Część z utworzonych subparametrów dla datapoint'u FRED_PMA0	42
6.3	Pola parametrów dla kolejnych kanałów na płytach PM	44
6.4	Identyfikatory płyty TCM i płyt PM w systemie SCADA	44
6.5	Ekran listy konfiguracji FEE	49
6.6	Ekran parametrów konfiguracji FEE	50
6.7	Ekran dodawania konfiguracji FEE	51
6.8	Część z utworzonych subparametrów dla datapoint'u PM_FRED_RANGES . .	52
6.9	Panel kalibracji systemu elektroniki odczytowej dla parametru CFD_ZERO . .	59
6.10	Panel systemu FSM wraz z kategorią odpowiedzialną za stan płyt PM	60
6.11	Ekran płyty TCM dla detektora FV0	61
6.12	Ekran płyty PM dla detektora FV0	62
6.13	System FSM dla detektora FV0	62
6.14	Panel główny systemu SCADA dla detektora FV0	64

Spis tabel

Spis załączników

Kod źródłowy systemu FRED – plik ZIP

Projekt systemu SCADA w WinCC OA dla detektora FV0 – plik ZIP

Projekt systemu SCADA w WinCC OA dla laboratorium ALICE-FIT – plik ZIP