

**ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Теплоенергетичний**

Автоматизації теплоенергетичних процесів

До захисту допущено:

Завідувач кафедри

_____ Володимир ВОЛОЩУК

«__» _____ 20__ р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Автоматизоване управління технологічними процесами»

спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»

на тему: «Розробка ПОТ рішень на базі PLCnext»

Виконав:

студент IV курсу, групи ТО-81

Панасюк Володимир Ігорович _____

Керівник:

Доцент кафедри АТЕП КП

Пупена Олександр Миколайович _____

Консультант з розділу «Охорона праці»:

Професор, д. т. н.

Глива Валентин Анатолійович _____

Рецензент:

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент Панасюк В.І. _____

Київ – 2022 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Теплоенергетичний
Автоматизації теплоенергетичних процесів

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 151 «Автоматизація та комп'ютерно-інтегровані технології»

Освітньо-професійна програма «Комп'ютерно-інтегровані технологічні процеси і виробництва»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Володимир ВОЛОЩУК

«___» _____ 20__р.

ЗАВДАННЯ

на дипломний проєкт студенту

Панасюку Володимирі Ігоровичу

1. Тема проєкту «Розробка IoT рішень на базі PLCnext», керівник проєкту Пупена Олександр Миколайович, доцент кафедри АТЕП КПІ, затверджені наказом по університету від «8» травня 2022р.№ 965-с
2. Термін подання студентом проєкту 15 червня 2022 року
3. Вихідні дані до проєкту наявність контролеру PLCnext.
4. Зміст пояснювальної записки 1.Постановка задачі цифровізації, 2.Проектування ПТК, 3.Розробка програмного забезпечення бібліотеки, 4. Імітаційне моделювання і аналіз функціонування ПТК, 5. Охорона праці.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) 1.Схема структурна ПТК

2.Схема функціональної структури ПТК

6. Консультанти розділів проекту

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| Охорона праці | Глива В.А., професор | | |

7. Дата видачі завдання

2 травня 2022 року

Календарний план

| № з/п | Назва етапів виконання дипломного проекту | Термін виконання етапів проекту | Примітка |
|-------|--|---------------------------------|----------|
| 1 | Постановка задачі цифровізації. Обґрунтування та аналіз. | 3.05.2022 | |
| 2 | Схема структурна ПТК | | |
| 3 | Схема функціональної структури ПТК | | |
| 5 | Вибір платформи для розробки | 10.05.2022 | |
| 6 | Розробка імітаційної моделі на PLCnext Engineer. | | |
| 7 | Розробка НМІ для імітаційної моделі | | |
| 8 | Ознайомлення з PACFramework | | |
| 9 | Розгортання рівнів PACFramework | 25.05.2022 | |
| 10 | Розроблення НМІ для PACFramework | | |
| 11 | Розроблення інтеграцій ПоТ | | |
| 12 | Перевірка роботи ПоТ | | |
| 13 | Передзахист ДП | 15.06.2022 | |
| 14 | Захист ДП | 21.06.2022 | |

Студент

Володимир ПАНАСЮК

Керівник

Олександр ПУПЕНА

Анотація

Дипломний проєкт виконаний на тему «Розробка ПоТ рішень на базі PLCnext». Результатом виконання кваліфікаційної роботи є розроблене рішення ПоТ з використанням PLCnext. Використання та адаптування бібліотеки каркасу з відкритим кодом PACFramework до PLCnext дало можливість стандартизувати рівень Edge в ПоТ рішеннях та покращити контроль і обслуговування виробничого обладнання та зменшити витрати часу на рутинну роботу персоналу.

Всі задачі, що були поставлені на початку роботи виконані, а саме: ознайомлення з новим середовищем розробки PLCnext Engineer, написання імітаційного об'єкту з НМІ для наладки, ознайомлення з технологією PACFramework та його розгортання та обмін даними через OPC UA між PLCnext та Node-Red.

Ключові слова: PLCnext, ПоТ, PACFramework, Node-Red.

ABSTRACT

The diploma project was made on the topic "Development of IIoT solutions based on PLCnext". The result of the qualification work is a developed IIoT solution using PLCnext. The use and adaptation of the PACFramework open source framework library to PLCnext has made it possible to standardize the Edge level in IIoT solutions and to improve the control and maintenance of production equipment and reduce the time spent on routine staff work.

All the tasks set at the beginning of the work have been completed, namely: familiarization with the new PLCnext Engineer development environment, writing a simulation object with HMI for configuration, familiarization with PACFramework technology and its deployment and data exchange via OPC UA between PLCnext and Node- Red.

Keywords: PLCnext, IIoT, PACFramework, Node-Red.

**Пояснювальна записка
до дипломного проекту
на тему: «Розробка IoT рішень на базі PLCnext»**

Київ – 2022 року

ЗМІСТ

| | | |
|--|--|----|
| ВСТУП | | 11 |
| 1. ПОСТАНОВКА ЗАДАЧІ ЦИФРОВІЗАЦІЇ | | 13 |
| 1.1. Огляд і аналіз актуальних задач автоматизації | | 13 |
| 1.2. Техніко-економічне обґрунтування задачі автоматизації | | 15 |
| 1.3. Опис функціональної структури задачі автоматизації | | 16 |
| 1.4. Опис вимог до реалізації функцій задачі автоматизації | | 18 |
| 2. ПРОЄКТУВАННЯ ПТК | | 20 |
| 2.1. Обґрунтування і вибір платформи і структури ПТК | | 20 |
| 2.2. Загальний опис каркасу | | 22 |
| 2.3. Опис класу PLC частини ПТК каркасу | | 27 |
| 2.4. Опис класів LVL0 частини ПТК каркасу | | 32 |
| 2.5. Опис класів LVL1 частини ПТК каркасу | | 35 |
| 2.6. Опис класів LVL2 частини ПТК каркасу | | 37 |
| 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ БІБЛІОТЕКИ | | 40 |
| 3.1. Загальний опис програмного забезпечення | | 40 |
| 3.2. PLCFN | | 41 |
| 3.3. Рівень каналів | | 44 |
| CHAIFN | | 45 |
| CHAOFN | | 47 |
| CHDIFN | | 49 |
| CHDOFN | | 51 |
| 3.4. Рівень технологічних змінних | | 54 |
| AIVARFN | | 54 |
| AOVARFN | | 65 |
| DIVARFN | | 70 |
| DOVARFN | | 75 |

| | | | | | | | | | |
|-------------------------------------|--------------|----------|--------|------|--|--|-------|---------|--|
| | | | | | | | | | |
| | | | | | | | | | |
| Змн. | Лист | № докум. | Підпис | Дата | | | | | |
| Розроб. | Панасюк В.І. | | | | | | | | |
| Перевірив | Пупена О.М. | | | | | | | | |
| | | | | | | | | | |
| Н. Контр. | Некрашевич | | | | | | | | |
| Затверд. | Волощук | | | | | | | | |
| Розробка IoT рішень на базі PLCnext | | | | | | Літ. | Арквш | Арквштв | |
| | | | | | | 7 | 134 | | |
| | | | | | | «КПІ ім. Ігоря Сікорського», ТЕФ, ТО-81 | | | |

| | |
|--|-----|
| 3.5. Рівень пристроїв | 79 |
| VLVDFN | 80 |
| VLVAFN | 85 |
| DRVFN | 87 |
| 3.6. Розроблення інтеграції ПоТ | 92 |
| 4. ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ І АНАЛІЗ ФУНКЦІОНУВАННЯ ПТК | 95 |
| 4.1. Розроблення імітаційної моделі | 96 |
| Танки | 97 |
| Дозатори | 98 |
| Трьохходовий клапан | 98 |
| Вся установка | 100 |
| Людино-машинний інтерфейс імітаційної установки | 103 |
| 4.2. Обробка каналів | 103 |
| aichs | 103 |
| dichs | 104 |
| aochs | 105 |
| dochs | 105 |
| moduls | 107 |
| 4.3. Обробка технологічних змінних | 109 |
| aivars | 109 |
| divars | 109 |
| aovars | 111 |
| dovars | 111 |
| 4.4. Обробка пристроїв | 112 |
| resolution | 112 |
| actrs | 113 |
| 4.5. Формування карти ПЛК та ініціалізування | 114 |
| plcmaps | 114 |
| INITVARS | 114 |
| 4.6. Основна програма | 117 |
| 4.7. Перевірка роботи з використанням людино-машинного інтерфейсу | 117 |
| 4.8. Перевірка роботи ПоТ | 120 |
| 5. ОХОРОНА ПРАЦІ | 123 |
| 5.1. Технічні рішення та організаційні заходи з безпечної експлуатації засобів автоматизації, технологічного обладнання та засобів обчислювальної техніки. | 124 |

5.2. Вимоги гігієни праці та виробничої санітарії при використанні засобів обчислювальної техніки 127

5.3. Пожежна безпека та профілактика 132

ВИСНОВКИ 134

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 135

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 9 |

Перелік умовних скорочень

ТОУ – технологічний об’єкт управління

ВМА – Вимірювальна апаратура

ВКА – Виконавча апаратура

ПЛК – Програмно логічний контролер

IoT Gateway – інтелектуальний центральний центр для пристроїв Інтернету речей

Edge – периферія IoT

Network – мережне підключення

Apps – застосунки націлені на моніторинг стану системи

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 10 |

ВСТУП

Метою роботи є розробка концепції побудови ІоТ рішень на базі PLCnext з використанням каркасу PACFramework.

ІоТ – це основна технологія для ініціатив Індустрії 4.0, в якій використовуються підключені інтелектуальні датчики, виконавчі механізми та багато іншого, з можливістю віддаленого контролю та управління в автоматизованому режимі через інтернет. Технологія дозволяє збирати, аналізувати та обмінюватися даними, що підвищує продуктивності та ефективності, а також сприяє іншим економічним перевагам.

Для вирішення питань інтеграції в роботі використовується каркас PACFramework – програмний каркас для рівня ПЛК з відкритим кодом, який призначений для полегшення та прискорення побудови прикладних програм на базі мов IEC61131-3 для контролерів великої та середньої каналності з урахуванням більшості типових вимог та інтеграцію в ІоТ. Програми розроблені на базі PACFramework також полегшують роботу налагоджувального та експлуатаційного персоналу, зменшуючи час на боротьбу з несправностями шляхом підвищення ситуаційно обізнаності та надаючи додаткові можливості налагодження. Сьогодні PACFramework також супроводжується додатковими утилітами для автоматизації розгортання (PACFramework-tools) та інтеграції в ІоТ рішення (PACFramework IoT Gateway). Використання PACFramework дасть можливість стандартизувати типові об'єкти що у свою чергу спростить розгортання ІоТ рішень.

До початку даної кваліфікаційної роботи бібліотека PACFramework була доступна тільки для деяких платформ Schneider Electric та Siemens. Тому головні задачі які були поставлені для даної роботи це:

- розібратися в структурі ІоТ рішень
- розібратися з платформою PLCnext та способи інтеграції його в рішення ІоТ
- розібратися в PACFramework, як способі реалізації представлення сутностей в архітектурі ІоТ

- адаптувати існуючі бібліотеки PACFramework до PLCnext з перевіркою портабельності коду
- перевірити роботу PACFramework на імітаційній моделі
- інтегрування PLCnext в рішення ІоТ (OPC UA, MQTT)

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 12 |

1. ПОСТАНОВКА ЗАДАЧІ ЦИФРОВІЗАЦІЇ

1.1. Огляд і аналіз актуальних задач автоматизації

У промисловому виробництві використання технології IoT дає можливість «речам» (обладнанню, механізмам, машинам тощо) об'єднуватися в єдину мережу для ідентифікації один одного, характеристики свого стану, передачі даних один одному та їх обробки без втручання людини. І що ще важливіше, IoT збирає масу даних та аналізує їх, щоб допомогти керівникам чи власникам бізнесу швидше приймати обґрунтовані рішення.

В результаті підприємство, що впровадило ПоТ, може користуватися такими перевагами, які забезпечують конкурентні переваги:

- підвищення операційної ефективності
- покращена операційна та ланцюгова прозорість
- скорочення часу простою, підвищення продуктивності
- оптимізовані виробничі процеси
- покращена якість продукції
- покращене керування запасами
- підвищений рівень автоматизації
- зниження витрат та ризиків
- підвищення гнучкості бізнесу

Системи ПоТ зазвичай є багаторівневою модульною архітектурою цифрових технологій (Рис. 1.1).

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 13 |

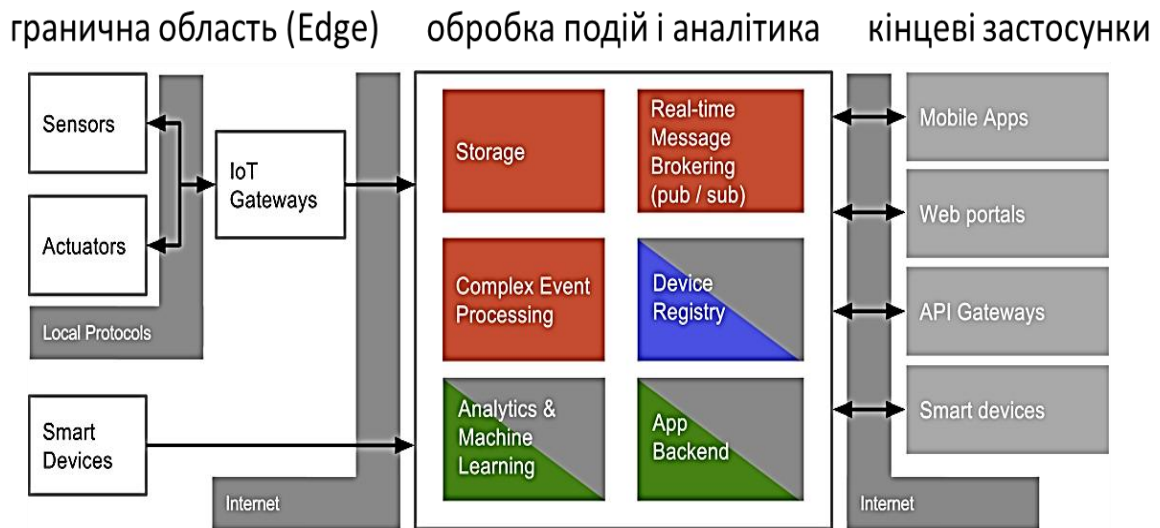


Рисунок 1.1 - Приклад архітектури ІоТ

Взаємодія з «речами» відбувається через датчики (sensors) та виконавчі механізми (Actuators). Ці датчики разом з усією інфраструктурою для інтеграції з рівнем обробки подій через мережу Internet формують так звану граничну область (Edge).

Події, тобто дані, що поступають з граничної області зберігаються і обробляються відповідно до задачі (рівень обробки подій і аналітики, event processing, Platform). На цьому рівні події зберігаються (storage), обробляються (Event Processing), перенаправляються потрібним застосункам (Real-Time Message Brokering, Stream Processing). Додатково на цьому рівні відбувається адміністрування та керування пристроями з граничної області (Device Registry, Edge Device Management). Події обробляються з використанням аналітичних сервісів (Analytics) на основі них проводиться машинне навчання (Machine Learning), що дозволяє зробити певні висновки про об'єкт. Цей рівень як правило реалізований з використанням хмарних (Cloud) або туманних (Fog) обчислень. Отримання результатів, контроль, віддалене керування та адміністрування системи проводиться через кінцеві застосунки з використанням Internet (доповненої і віртуальної реальності (AR/VR) тощо) .

1.2. Техніко-економічне обґрунтування задачі автоматизації

Впровадження системи дасть можливість зменшити витрати часу на: налагодження системи, вияв факту несправності та усунення причин, навчання персоналу, розроблення ПЗ.

Для кращої ситуаційної обізнаності на об'єкті необхідно формувати контекстну інформацію з використанням структурованих змінних, які містять достатню кількість даних про технологічну змінну. Наприклад, поточне значення величини, наявність тривоги, стан технологічного параметру, мінімум/максимум величини тощо.

При нештатних ситуаціях на об'єкті, час на вияв факту несправності та усунення причин може призупинити процес виробництва, що в свою чергу призводить до втрати грошей. Використовуючи структуровані дані, які базуються на стано-орієнтованому підході, можна отримати додаткову інформацію. Ця інформація, може вказати на місце та причини несправності, що пришвидшує її усунення.

Налагодженні системи часто потребує велику кількість рутинної роботи: перевірка імітованих сигналів датчиків, зміна стану вхідної змінної незалежно від значення фізичного каналу, виконання операцій форсування, зміни значення невикористаного каналу, тощо. Деякі процеси налагодження потребують участь розробника ПЗ. Вирішити проблем налагодження може програмна імітація та форсування, які повинні бути доступні з засобів НМІ. Це прискорює наладку системи та наладка не залежить від розробника ПЗ.

При створенні великих проектів, варто автоматизувати процес розроблення. Першим кроком до цього є стандартизація представлення програмних об'єктів, другим - створення програмного забезпечення для автоматизації перетворення вихідних даних проекту (перелік технологічних змінних, VM, тощо) в код програми. Також побудувавши правильну архітектуру проекту, при потребах зміни алгоритму роботи програми, потрібно змінити код в кількох місцях, що в свою чергу пришвидшує розробку ПЗ.

Також потрібно не забувати про навчальний персонал. Навчання роботи персоналу не завжди можливе на реальних об'єктах. Тому як варіант вирішення цієї

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 15 |

проблеми є використання імітаційного моделювання в програмі контролеру та ознайомлення з концепціями, які використовують на виробництві.

1.3. Опис функціональної структури задачі автоматизації

Архітектура проекту (Рис. 1.2) складається з трьох основних доменів - застосунки, мережі та периферії.

Застосунки (Apps) - застосунки, які націлені на моніторинг стану системи та керування через інтернет (доповнена і віртуальна реальності (AR/VR)).

Мережне підключення (Network) побудовано на основі протоколів IP, деякі з яких є звичайними протоколами, такими як HTTP, а інші є більш спеціалізованими протоколами, розробленими спеціально для використання додатків на основі IoT, які передбачають збір і розповсюдження великої кількості даних (MQTT).

Периферія (Edge) складається з «речей» IoT, таких як датчики, виконавчі механізми та контролери. Також IoT Gateway, який забезпечує обмін даними між доменами архітектури використовуючи популярні протоколи обміну даними (OPC UA, MQTT).

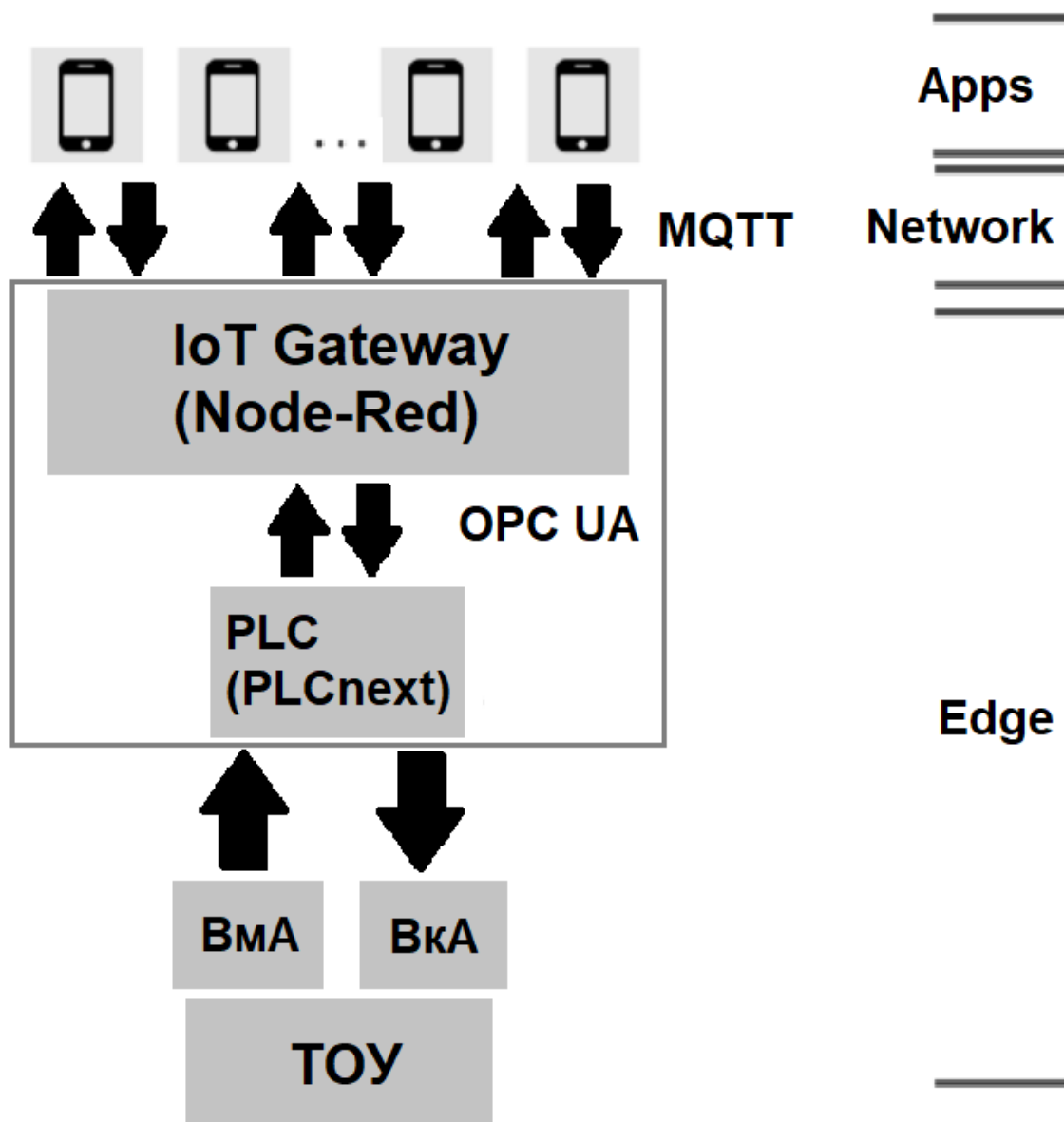


Рисунок 1.2 - Архітектура ІоТ

ТОУ – технологічний об'єкт управління

ВмА – Вимірювальна апаратура

ВкА – Виконавча апаратура

PLC – Програмно логічний контролер

IoT Gateway – інтелектуальний центральний центр для пристроїв Інтернету речей (IoT)

Edge – периферія IoT

Network – мережне підключення

Apps – застосунки націлені на моніторинг стану системи

1.4. Опис вимог до реалізації функцій задачі автоматизації

Головні вимоги, які повинна мати система:

- вимоги до обслуговування:
 - висока спостережливість роботи системи
 - продуманий механізм функціонування тривоги та подій
 - зменшення затрат часу на налагодження системи
 - зменшення затрат часу на виявлення несправностей та їх усунення
 - зменшення затрат часу на навчання персоналу
- можливість взаємодії з іншими застосунками (наприклад, доповнена і віртуальна реальності (AR/VR))
- обмін даними з IoT Gateway, використовуючи найпоширеніші протоколи обміну даними.
- локальний веб-інтерфейс

Вимоги до демонстраційного проекту.

Для того щоб перевірити рішення необхідно розробити демопроект.

Пропонується розробити імітаційну модель рецептурно-змішувального комплексу для приготування продукції з дозуванням за рівнем компонентів у дозаторі та періодичним змішуванням компонентів (Рис. 1.3) та можливістю доступу до веб-інтерфейсу моделі.

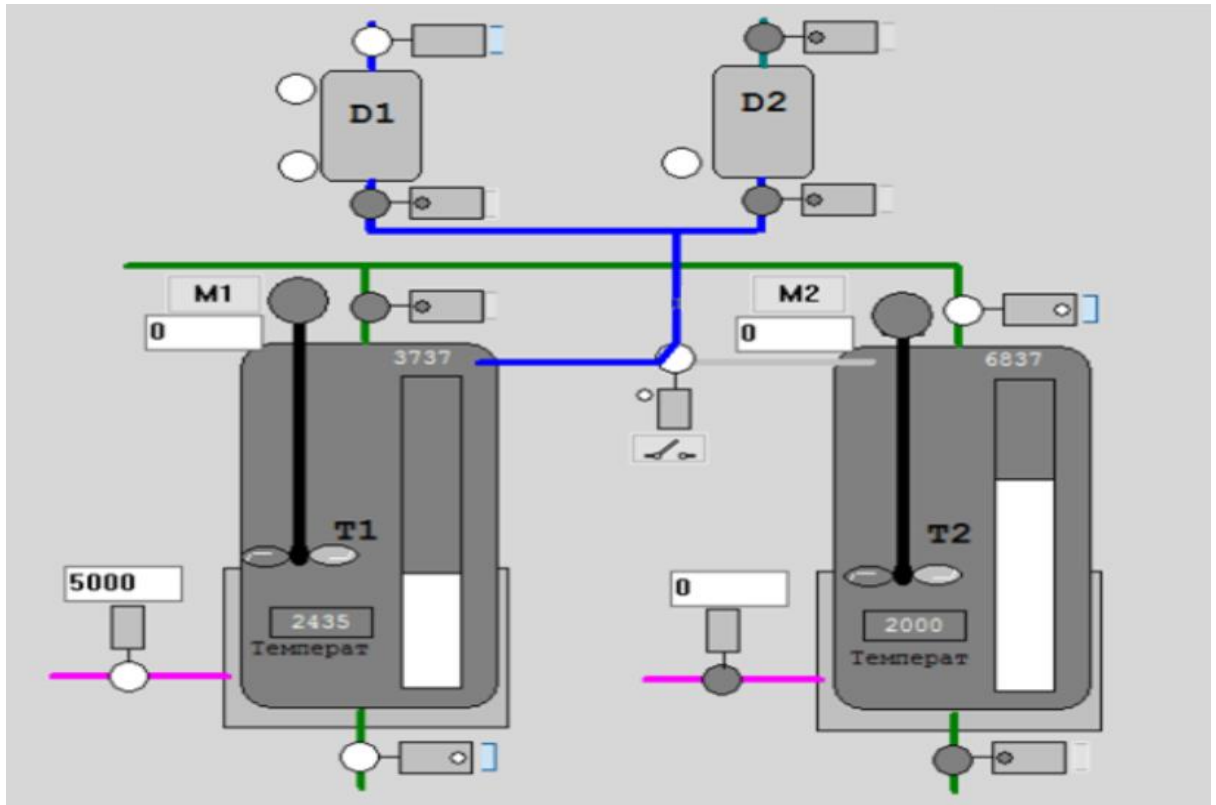


Рисунок 1.3 - Зображення імітованої установки

Технологічна установка приготування складається з трьох основних елементів системи: танки, дозатори та трьохходовий клапан перемикання трубопроводу подачі з дозаторів.

Танки - ємності в яких готуються продукти за різними рецептами (T1, T2). Танки забезпечені запірними клапанами набору та зливу (з датчиками кінцевого положення), датчики рівня та температури, регулюючий клапан подачі теплоагенту у теплообмінний кожух, привід мішалки.

Дозатори - мірні ємності, які забезпечують подачу дози компоненту (D1 та D2). Дозатори забезпечені сигналізаторами верхнього та нижнього рівнів, запірними клапанами набору та зливу.

Трьохходовий клапан перемикання трубопроводу подачі з дозаторів на один із танків. Нормальному стан положення перший танк. Клапан має датчики кінцевого положення першого та другого танків.

2. ПРОЄКТУВАННЯ ПТК

2.1. Обґрунтування і вибір платформи і структури ПТК

Компанія Phoenix Contact випустила програмний застосунок для прискорено розробки програм з використанням кращих інженерних інструментів під назвою PLCnext Engineer (див. рис. 2.1). Під час проходження переддипломної практики було ознайомлено з цим рішенням та вирішено використовувати це середовище для розробки даного дипломного проекту.

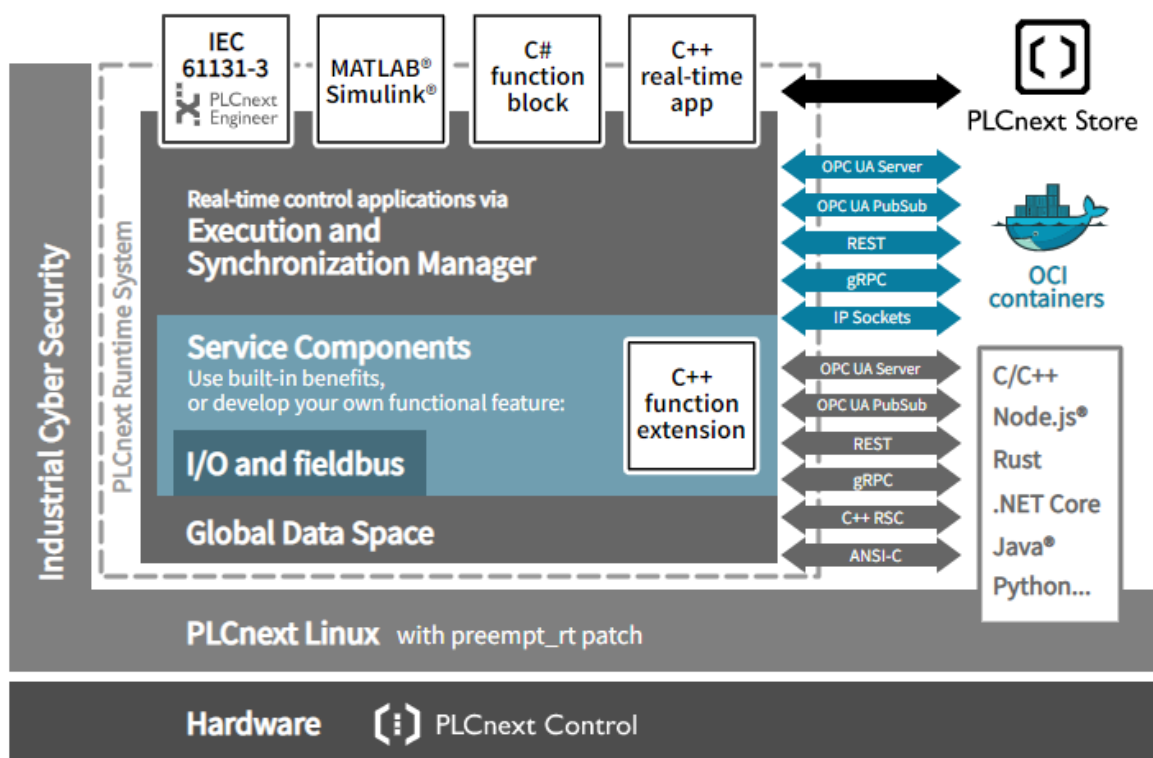


Рисунок 2.1 - PLCnext Engineer у структурі PLCnext Technology

Особливості програмної платформи PLCnext Engineer, яку можна гнучко та індивідуально розширювати за допомогою функціональних доповнень це:

- Безпечне програмування, сертифіковане MEK 61508;
- Конструктивна безпека із контролерами сімейства PLCnext Control;
- Єдине управління у всьому світі завдяки стандартизованому програмуванню згідно MEK 61131;
- Одна програма для всіх інжинірингових завдань;

- Скорочення витрат на оплату праці та навчання завдяки оптимізованому інтерфейсу користувача;
- Економія часу та засобів завдяки комплексному програмуванню в одній оболонці;
- Гнучкі інжинірингові можливості завдяки інтеграції індивідуальних функціональних доповнень до безкоштовної базової версії;
- Спрощення інжинірингового процесу завдяки використанню модулів автоматизації та об'єктно-орієнтованому програмуванню.

Програмне забезпечення також пропонує зручну роботу при комбінуванні програм мовою високого рівня або моделей MATLAB® Simulink® та стандартної автоматизації. Простий спосіб конфігурування мереж PROFINET. Прискорена розробка програм, використовуючи готові рішення та програмні модулі з магазину PLCnext Store.

Основним критерієм вибору середовища розробки було наявність імітатора контролера, який з'явився у 2022 році, для моделювання без апаратного контролера. Моделювання PLCnext Engineer пропонує широкий спектр можливостей для моделювання функцій, компонентів та моделей проекту. Тобто розробника програм більше не залежать від цільового обладнання. Це надає наступні можливості:

- Впливати на перебіг програмного коду шляхом примусового введення змінних даних процесу або налагодження вихідних сигналів;
- Моделювати та випробувати частину HMI проекту PLCnext Engineer з повним набором функцій;
- Використовувати веб-керування вашої імітованої апаратної цілі, щоб додати нових користувачів, а потім установити ролі користувачів, щоб перевірити поведінку вашого проекту щодо різних аутентифікацій користувачів;
- Налаштувати, підготувати та перевірити підключення до сервера OPC UA за допомогою клієнта OPC UA на локальному хості (127.0.0.1).

Вирішення проблеми розробки IoT компанія Phoenix Contact пропонує за допомогою бібліотеки "The IoT Framework". Ця бібліотека є безкоштовною для завантаження з магазину PLCnext Store. IoT Framework включає функції для

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 21 |

нормалізації даних, зберігання та візуалізації. Завдяки високій модульності його можна легко розширити за допомогою вхідних і вихідних програм для підключення ОТ та ІТ.

Наступні конектори вже включені в систему ІоТ Framework:

- MSSQL
- MySQL
- OPC UA
- PN-Machine manager (2904472)
- AXL F DI (цифровий введення вхідних даних)
- AXL F AI (аналоговий вхідний/вихідний)
- AXL F PM (вимірювання потужності)
- MQTT
- AWS-Хмара
- NodeRed
- Empro

ІоТ Framework розроблений тільки для ПЛК PLCnext. При інтегруванні в систему ІоТ ПЛК інших виробників можуть виникнути ускладнення приведення до єдиної платформи. Крім того ІоТ Framework не означає правила семантики сутностей, що не стандартизує їх представлення і ускладнює інтеграцію рішень навіть в межах одного бренду. Тому було вирішено використовувати концепцій PACFramework з його функціоналом (п. 2.2) для структурування та обробки даних. З подальшою відправкою даних на ІоТ Gateway, який базується на Node-RED, та спілкується з PLCnext по протоколу OPC UA. З можливістю налагодження системи через веб-інтерфейс.

2.2. Загальний опис каркасу

PACFramework - функціональний каркас для розробки прикладного програмного забезпечення для промислових контролерів(PLC/PAC), але не обмежених ними, з урахуванням типових вимог до систем керування, сучасних світових стандартів (ISA, IEC, ISO) та тенденцій (Industry 4.0, ІоТ).

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 22 |

Повний опис каркасу доступний на репозиторії <https://github.com/pupenasan/PACFramework>. У цьому та наступних підрозділах наводиться тільки та частина, що задіяна в імплементуванні в проєкті.

PACFramework - є open source проєктом, дає можливість швидко розробляти програмне забезпечення для PLC/PAC та HMI з функціоналом, достатнім для АСУТП різного призначення. Реалізація може бути на будь-яких програмних пристроях для керування та контролю.

Передумови створення та основні ідеї каркасу:

1. Проблеми інтеграції з MES/MOM та іншими підсистемами;
2. Низька спостережність роботи об'єкта навіть при достатній кількості вимірювальних даних;
3. «статична» діагностика процесу без прив'язки до типу продукції та особливостей умов;
4. Погана реалізація самодіагностики та не врахування відмов в самій системі АСУ ТП;
5. Недостатньо продуманий механізм функціонування тривог та подій;
6. Складність, значні витрати часу на налагодження системи;
7. Складність, значні витрати часу на вияв факту несправності та усунення причин;
8. Складність, значні витрати ресурсів на навчання персоналу.

Ієрархія обладнання в PACFramework складається хоча би з трьох рівнів (див. рис. 2.2). Всі елементи вищих рівнів можуть взаємодіяти з рівнями нижчих рівнів (змінювати значення, перемикати в різні режими, змінювати налаштування тривог), але другий рівень не може безпосередньо взаємодіяти з нульовим рівнем, в той же час нульовий рівень не знає від кого він залежить.

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 23 |

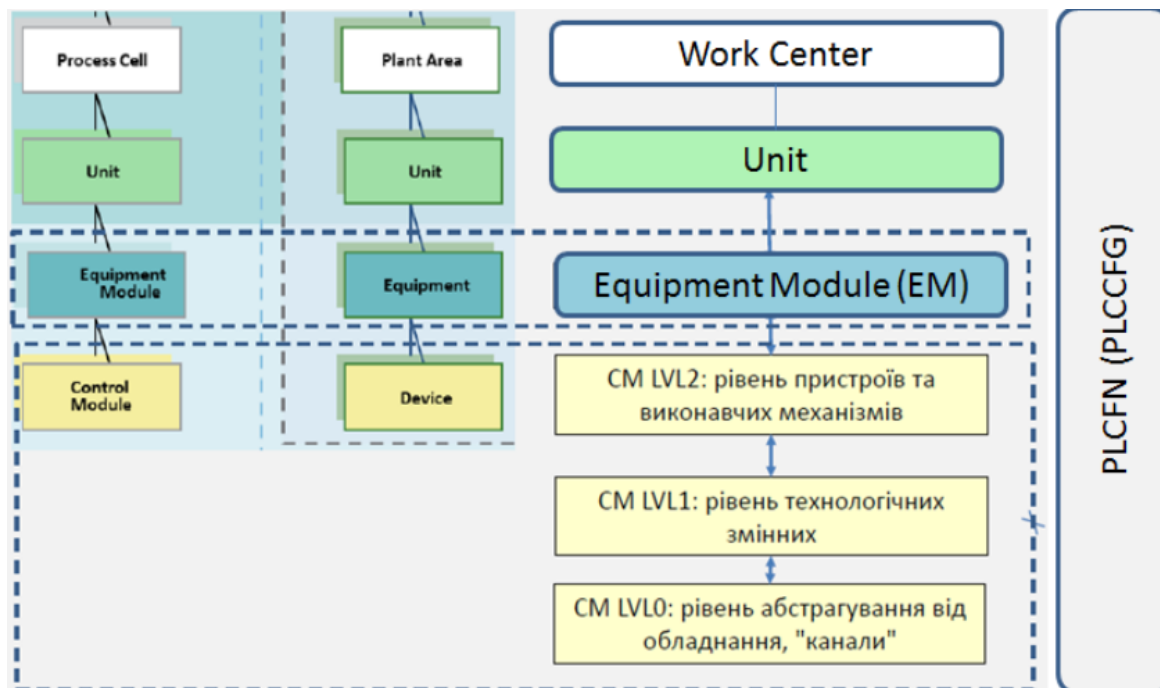


Рисунок 2.2 - Ієрархія Control Module (CM).

Нульовий рівень (LVL0) або рівень каналів ПЛК - канали контролеру. Виконує наступні функції: діагностика каналу, прив'язки логічних каналів до фізичних, форсування входів/виходів.

Елементи нульового рівня представляють масиви існуючих каналів контролеру (абстрагуючись від конкретики реальних приладів). Кожен елемент масиву прив'язується по унікальному ідентифікатору до каналу на програмному рівні. Прив'язуючи до фізичних значень каналу, виконуються наступні функції: діагностика для вищих рівнів (наприклад ознаки достовірності, та причини відмови чи несправності), форсування значень, достовірність прив'язки технологічної змінної до каналу (див. рис. 2.3).

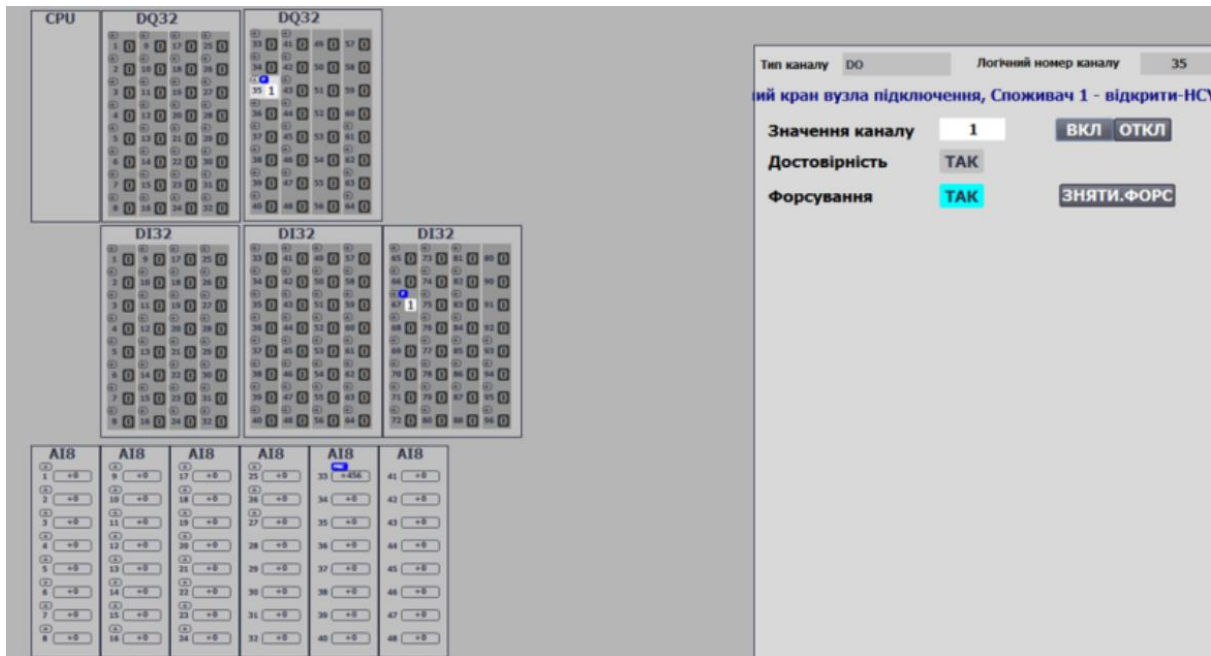


Рисунок 2.3 - Приклад використання функцій каналів на НМІ.

Перший рівень (LVL1) або рівень технологічних змінних - технологічні змінні для повної обробки інформації з процесу, включаючи прив'язку до каналу, фільтрацію, масштабування, інверсію і т.п.; для зручності відлагодження процесу; для функцій імітаційного моделювання; для функцій технологічної сигналізації;

Технологічна змінна може бути прив'язана до каналу того ж типу по її номеру. Тобто привязка відбувається динамічно, що дає змогу змінювання фізичного підключення конкретного пристрою у випадку поломки (може бути на програмному рівні). Привязуючи канали до технологічних змінних ми не лише отримуємо діагностичну інформацію, а також отримуємо додатковий функціонал, який реалізований на цьому рівні: перевірка на достовірність змінної (при виході за діапазон вимірювання), діагностика роботи каналу, масштабування, фільтрація, інвертування, форсування (ручний режим), конфігурування та обробка тривог (див. рис. 2.4, рис. 2.5).

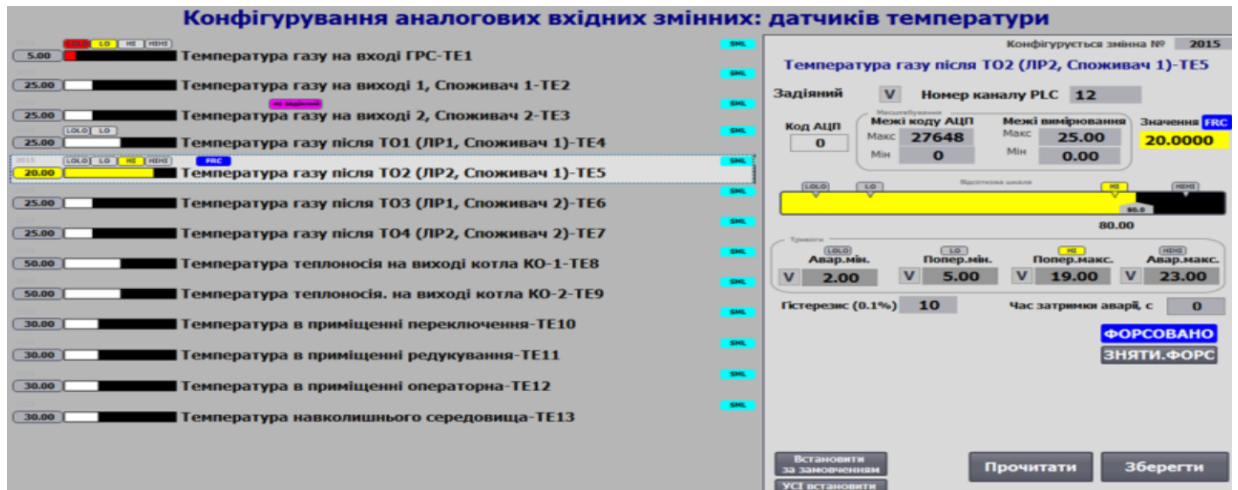


Рисунок 2.4 - Приклад використання функцій аналогових вхідних змінних на НМІ.

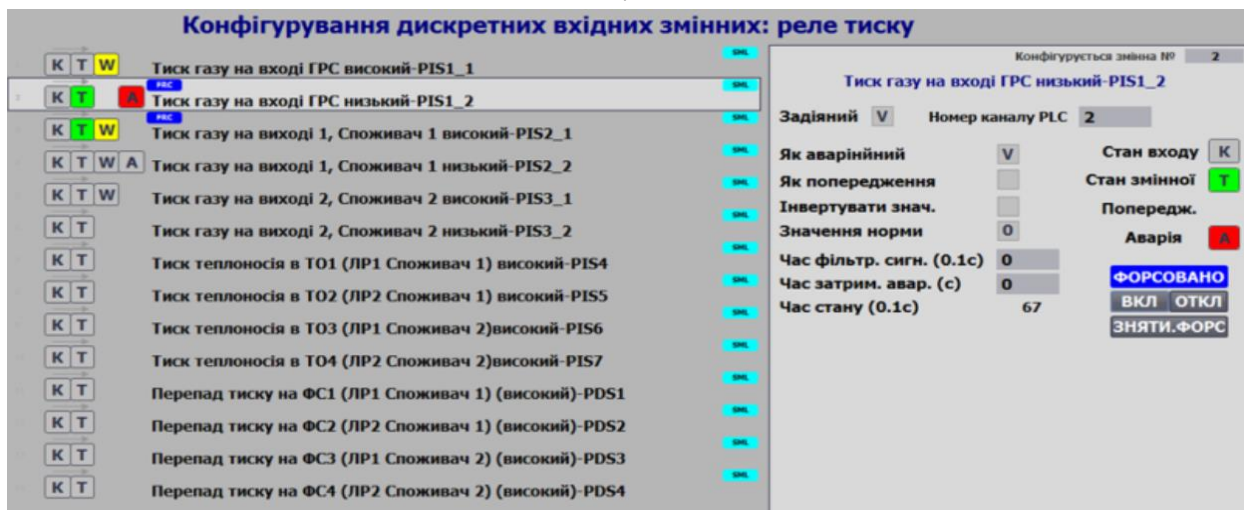


Рисунок 2.5 - Приклад використання функцій аналогових вихідних змінних на НМІ.

Другий рівень (LVL2) або рівень пристроїв - рівень пристроїв та виконавчих механізмів. Використовується: для зручності налагодження процесу, для функцій імітаційного моделювання, для функцій технологічної сигналізації, для ведення статистики:

- 1) Виконавчі механізми (запірні клапани, регулюючі клапани, двигуни, насоси);
- 2) Контури регулювання та управління: для функцій керування зі зворотним зв'язком;
- 3) Інші пристрої, що включають кілька технологічних змінних та мають окремо виділені стани.

Елементи другого рівня передбачають двосторонню взаємодію з технологічними змінними (LVL1) запис та читання. Це передбачає наступні можливості цього рівня:

- 3) Записує загальний час з початку контролера, астрономічний час, початок доби, години;
- 4) Скидає тривоги, лічильники тривог та деякі біти статусних слів;
- 5) Показує плинний час.

Клас PLC повинен реалізовувати одну із головних функцій каркасу - PLCFN. Реалізація PLCFN залежить від середовища програмної розробки. Також функція повинна запускатися першою в основній задачі, для правильної роботи.

Для ПЛК створюється змінна типу PLC_CFG (див. таблиця 1). Вона використовується як аргумент для усіх функцій та функціональних блоків, які реалізовані в каркасі. У випадках розподіленого керування для обміну між ПЛК, змінна типу PLC_CFG може спростити координацію між ними.

На початку роботи функції PLCFN усі біти полів ALM1, ALM2, та деякі біти STA скидаються. Це розроблено для того, щоб будь який з елементів структури програми міг змінити значення біту, тим самим сигналізуючи про твердження "хоча б один".

Таблиця 2.1 - Структура змінної типу PLC_CFG.

| Ім'я параметру | Тип параметру | Номер біта в 16-бітній структурі | Опис |
|----------------|---------------|----------------------------------|--|
| ID | UINT | | Унікальний ідентифікатор, наприклад для ідентифікації PLC |
| CLSID | UINT | | 16#21xx |
| STA | UINT | | Може бути набір біт типу PLC_STA |
| ACON2ERR | BOOL | 0 | 1 - є помилка зв'язку з сусідні ПЛК в парі (для теплого резервування) |
| APLC2STOP | BOOL | 1 | 1 - спарений ПЛК в стопі (для теплого резервування) |
| BLK | BOOL | 2 | 1 – є хоча б один заблокований ВМ |
| ALDIS | BOOL | 3 | 1 – є хоча б одна відключена тривога |
| DIOON | BOOL | 4 | 1 - йде опитування по мережі (MODBUS або аналогічний) засобів віддаленого вводу/виводу |
| DIOERR | BOOL | 5 | 1 - є помилка на DIO |
| DBLCKALL | BOOL | 6 | 1 - усі приводи розблоковані |
| FRC | BOOL | 7 | 1 – хоча б одна змінна форсована (або ручний режим) на будь якому рівні |

| | | | |
|--------|------|----|--|
| SMLALL | BOOL | 8 | 1 – все в режимі в імітації, примушує усі СМ перейти в режим імітації |
| DISP | BOOL | 9 | 1 – хоча б один елемент в режимі ручного управління |
| FRC2 | BOOL | 10 | 1 – є хоча б один форсований керівний елемент (рівня 2), клапан, двигун |
| FRC1 | BOOL | 11 | 1 – хоча б одна змінна форсована (рівня 1) |
| SCN1 | BOOL | 12 | 1 – перший скан |
| FRC0 | BOOL | 13 | 1 – хоча б одна змінна форсована (рівня 0) |
| SML | BOOL | 14 | 1 – хоча б один об'єкт в режимі імітації |
| CMDACK | BOOL | 15 | 0 - підтвердження команди, команда пройшла через весь цикл задачі і отримана усіма |
| CMD | UINT | | Команди з НМІ:16#0100 – прочитати конфігурацію;16#0101 – записати конфігурацію; 16#301 - включити режим деблокування усіх ВМ; 16#302 - відключити режим деблокування усіх ВМ; 16#300 - переключити режим деблокування усіх ВМ; 16#0111 – синхронізувати час з верхнім рівнем; 16#0301 – вимкнути сирену; 16#0302 – увімкнути сирену; 16#4101 – записати конфігурація за замовчуванням для усіх змінних DIVAR; 16#4102 – записати конфігурація за замовчуванням для усіх змінних AIVAR; 16#4103 – записати конфігурація за замовчуванням для усіх змінних DOVAR; 16#4104 – записати конфігурація за замовчуванням для усіх змінних AOVAR;16#4301 – форсувати всі об'єкти LVL0; 16#4302 – дефорсувати всі об'єкти LVL0;16#4303 – форсувати всі об'єкти LVL1;16#4304 – дефорсувати всі об'єкти LVL1 |
| CMDPRG | UINT | | Команди з програмного керування (побітові) |
| PRM1 | UINT | | дискретні параметри (проектно залежні) |
| PRM2 | UINT | | дискретні параметри (проектно залежні) |
| PLS | UINT | | Може бути набір біт типу PLS |
| P100MS | BOOL | 0 | імпульс на один цикл основної задачі з періодичністю 100 мс (адекватно працюватиме тільки для циклів <50ms) |
| P200MS | BOOL | 1 | 200 мс (адекватно працюватиме тільки для циклів <100ms) |
| P500MS | BOOL | 2 | 500 мс (адекватно працюватиме тільки для циклів <250ms) |
| P1S | BOOL | 3 | 1 с |
| P2S | BOOL | 4 | 2 с |
| P5S | BOOL | 5 | 5 с |
| P10S | BOOL | 6 | 10 с |
| P60S | BOOL | 7 | 1 хв |
| M1S | BOOL | 8 | меандр з періодом 1 с (0.5 с + 0.5 с) |

| | | | |
|----------|-------|----|---|
| M2S | BOOL | 9 | меандр з періодом 2 с (1 с + 1 с) |
| NEWMIN | BOOL | 12 | 1 (на один цикл основної задачі) – початок хвилини |
| NEWHR | BOOL | 13 | 1 (на один цикл основної задачі) – початок години |
| NEWDAY | BOOL | 14 | 1 (на один цикл основної задачі) – початок доби |
| NEWSHIFT | BOOL | 15 | 1 (на один цикл основної задачі) – початок зміни |
| ALM1 | INT | | Може бути типу PLC_ALM1 |
| ALM | BOOL | 0 | 1, є хоча б одна тривога аварійного рівня |
| NWALM | BOOL | 1 | 1, нова тривога аварійного рівня |
| ALMNACK | BOOL | 2 | 1, є неквітовані тривоги |
| WRN | BOOL | 3 | 1, є хоча б одна тривога попереджувального рівня |
| NWWRN | BOOL | 4 | 1, нова тривога попереджувального рівня |
| WRNNACK | BOOL | 5 | 1, є неквітовані попередження |
| BAD | BOOL | 6 | 1, є хоча б одна тривога недостовірності |
| NWBAD | BOOL | 7 | 1, нова тривога недостовірності |
| BADNACK | BOOL | 8 | 1, є неквітовані тривоги недостовірності |
| EMCYSTP | BOOL | 9 | 1, аварійний зупин (грибок) |
| STP2RUN | BOOL | 10 | 1, момент переходу з аварійного зупину в роботу |
| CON2ERR | BOOL | 11 | 1 - є помилка зв'язку з сусідні ПЛК в парі (для теплового резервування) |
| PLC2STOP | BOOL | 12 | 1 - спарений ПЛК в стопі (для теплового резервування) |
| DIOERR | BOOL | 13 | 1 - є помилка DIO |
| PLCERR | BOOL | 14 | 1 – є апаратна помилка (ПЛК) |
| CONHIERR | BOOL | 15 | 1 - є помилка зв'язку з верхнім рівнем (ПЛК) |
| ALM2 | INT | | використовується за необхідності |
| STEP1 | INT | | крок основної програми |
| T_STEP1 | INT | | час кроку основної програми обробки, в (с) |
| MSG | UDINT | | для формування повідомлень, може бути як бітовим (32 повідомлення), так і числовим (за номером); передбачається, що повідомлення буде скидуватися в 0 автоматично через певний час (не менш ніж 2 інтервали опитування з HMI) |
| TQ | UDINT | | загальний час з початку 1-го циклу контролера (в секундах) |
| TQM | UDINT | | загальний час роботи ПЛК з моменту пуску (в хвилинах), потребує збереження в Retentive |
| DICNT | UINT | | кількість наявних каналів DI |

| | | | |
|-------------|--------------------|--|---|
| DOCNT | UINT | | кількість наявних каналів DO |
| AICNT | UINT | | кількість наявних каналів AI |
| AOCNT | UINT | | кількість наявних каналів AO |
| NOW | ARRAY[0..3] of INT | | плинний час астрономічний час в BCD форматі: NOW[0] seconds,-- (16#ss,--);NOW[1] hour, minute (16#hhmm);NOW[2] month, day (16#mmdd); NOW[3] year (16#yyyy) |
| SHIFTPARA | ARRAY[0..3] of INT | | Години передачі змін (2 або 3 зміни):SHIFT[0] – кількість змін; SHIFT[1] – передача 1-ї зміни (BCD 16#hhmm); SHIFT[2] – передача 2-ї зміни (BCD 16#hhmm); SHIFT[3] – передача 3-ї зміни (BCD 16#hhmm) |
| SHIFTNMB | UINT | | номер активної зміни |
| CNTALM | UINT | | лічильник кількості активних тривог "аварія" |
| CNTWRN | UINT | | лічильник кількості активних тривог "попередження" |
| CNTBAD | UINT | | лічильник кількості активних тривог "недостовірність" |
| CNTFRC | UINT | | лічильник кількості форсованих об'єктів |
| CNTMAN | UINT | | лічильник кількості ВМ в ручному режимі |
| TSK_LTIME | UINT | | плинний (останній) час задачі в мілісекундах |
| TSK_MAXTIME | UINT | | максимальний час задачі в мілісекундах |
| STA_PERM | UINT | | повторює STA, який був на початку виклику функції |
| ALM1_PERM | UINT | | повторює ALM1, який був на початку виклику функції |
| CNTALM_PERM | UINT | | кількість активних тривог "аварія" |
| CNTWRN_PERM | UINT | | кількість активних тривог "попередження" |
| CNTBAD_PERM | UINT | | кількість активних тривог "недостовірність" |
| CNTFRC_PERM | UINT | | кількість форсованих об'єктів |
| CNTMAN_PERM | UINT | | кількість ВМ в ручному режимі |
| MODULSCNT | INT | | кількість модулів |
| NOWns | UDINT | | нс (наносекунд) для плинного астрономічного часу |
| TQMS | UDINT | | мілісекундний лічильник, скидається при старті або після переповнення |

2.4. Опис класів LVL0 частини ПТК каркасу

Загальний опис класів LVL0.

Рівень каналів забезпечує абстрагування від конкретики пристроїв. Тобто реалізація цього рівня залежить як від вибраної платформи так і способу реалізації. Кожен елемент масиву каналів контролеру ідентифікується за унікальним номером, а прив'язка до фізичного каналу відбувається жорстко на програмному рівні. Елементи типу канал прив'язуються до фізичних каналів контролеру, а також виконують наступні функції:

- 1) Надають діагностичну інформацію на рівні які знаходяться вище по ієрархії;
- 2) Забезпечують форсування значення вхідного/вихідного каналів;
- 3) Демонстрація достовірності прив'язки технологічної зміни до каналу.

Передбачається п'ять основних класів, які реалізують функції відповідних каналів:

- 1) CHDI – дискретні вхідні канали;
- 2) CHDO – дискретні вихідні канали;
- 3) CHAI – аналогові вхідні канали;
- 4) CHAO – аналогові вихідні канали;
- 5) CHCOM – комунікаційні канали.

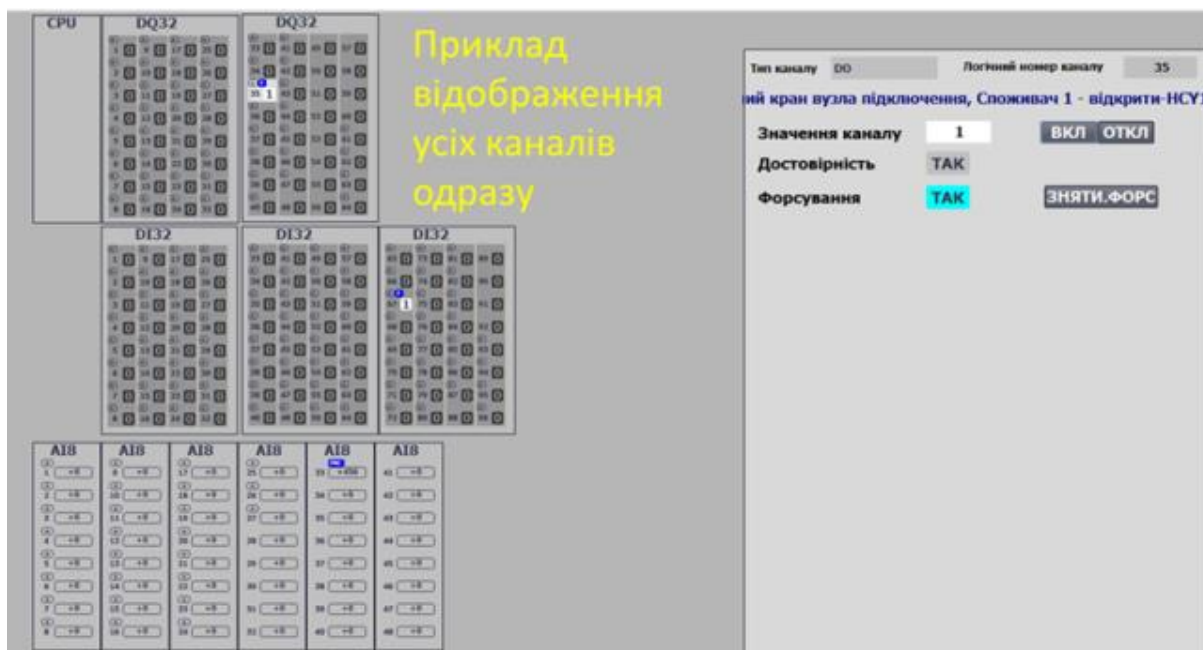
Мнемосхеми з відображенням стану каналів в каркасі зветься картою ПЛК. На карті ПЛК доступні функції, які реалізовані на рівні каналів. У випадку апаратної помилки канал підсвічується відповідним чином.

Для економії змінних рекомендується:

- 1) Використовувати буферну змінну типу CH_BUF на всі канали (зміна типу CH_BUF має структуру аналогічну до полів типу CH_CFG (див. таблиця 2));
- 2) Використовувати клас MODULES - представляє фізичний модуль на ПЛК, що може мати до чотирьох груп каналів, які представлені у вигляді Submodule (підмодуль).

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 32 |

НМІ рівню каналів (карта ПЛК) демонструє стан каналів модулів, яка зміна використовує канал. Також є можливість форсування значення якогось виходу. Приклади обох варіантів показані на рис. 2.7.



Приклад відображення каналів по модулям окремо

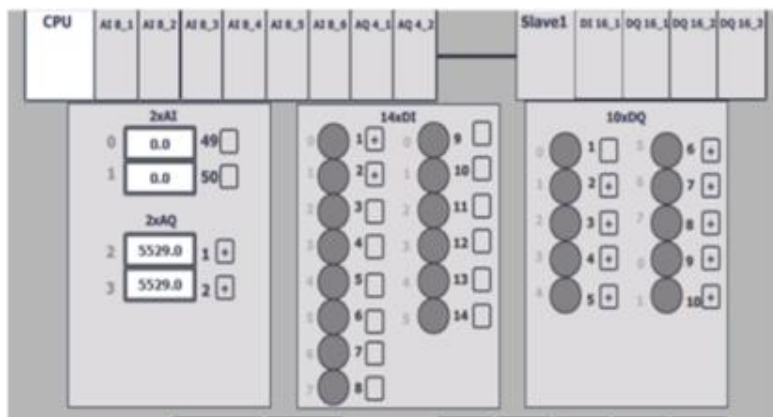


Рисунок 2.7 - Приклад відображення карт ПЛК

Таблиця 2.2 - Структура змінної типу CH_CFG.

| Ім'я параметру | Тип параметру | Номер біта в 16-бітній структурі | Опис |
|----------------|---------------|----------------------------------|--|
| ID | UINT | | Унікальний ідентифікатор - номер змінної. Кількість доступних каналів є останнім номером. Нумерація логічна, логіка нумерації вибирається при розробці проекту. ID = 0 зарезервований для помилкових операцій. Реальні канали нумеруються з 1. |
| CLSID | UINT | | - CHDI (CLSID=16#001x) – дискретні вхідні канали,- CHDO (CLSID=16#002x) – дискретні вихідні канали; - |

| | | | |
|-----------------------|------|----------|---|
| | | | CHAI (CLSID=16#003x) – аналогові вхідні канали;- CHAO (CLSID=16#004x) – аналогові вихідні канали. |
| STA | UINT | | Може бути набір біт типу CH_STA |
| VRAW | BOOL | 0 | для дискретного сигналу: - для CHDI значення з дискретного входу; - для CHDO значення на дискретний вихід; - в режимі FRC=1 може змінюватися із-зовні для CHDO; для CHAI: RAWINT>0; для CHAO: RAWINT>0 |
| VALB | BOOL | 1 | значення дискретного сигналу для CM верхнього рівня: - для CHDI об'єкт DIVAR зчитує його; - для CHDO об'єкт DOVAR записує в нього; - в режимі FRC=1 може змінюватися із-зовні для CHDI; для CHAI: CHCFG.VAL>0; для CHAO: CHCFG.VAL>0 |
| BAD | BOOL | 2 | 1 – є якась помилка каналу (BRK або SHRT або інша) |
| b3 | BOOL | 3 | резерв |
| PNG | BOOL | 4 | 1 – прийшов PING-запит від власника, обнуляється каналом (PONG-відповідь) |
| ULNK | BOOL | 5 | 1 – якщо канал використовується в технологічній змінній (прив'язаний) DIVAR/AIVAR/DOVAR/AOVAR |
| MERR | BOOL | 6 | 1 – є помилка на всьому модулі (діагностична інформація) |
| BRK | BOOL | 7 | 1 – помилка обриву каналу |
| SHRT | BOOL | 8 | 1 – помилка КЗ або перевантаження |
| NBD | BOOL | 9 | 1 – фізично канал не існує; наприклад, для відображення в буфері модуля MODULS для НМІ; якщо зручно використовувати неіснуючі канали для кратності адресації |
| STAb10 | BOOL | 10 | резерв |
| INIOTBUF | BOOL | 11 | 1 – змінна СНзавантажена в буфер IoT CH_BUF |
| INBUF | BOOL | 12 | 1 – змінна СНзавантажена в буфер CH_BUF |
| FRC | BOOL | 13 | 1 – значення форсоване |
| SML | BOOL | 14 | 1 – значення імітується (верхнім рівнем) |
| CMDLOAD | BOOL | 15 | 1 - запит на зчитування в буфер (тільки з НМІ) |
| CMD | UINT | | команда:16#1 - записати 1/MAX; 16#2 - записати 0/MIN;16#3 - для дискретних змінних: TOGGLE перемикання 0<->1; 16#3 - для аналогових змінних: MIDDLE виставлення половини з діапазону MIN/MAX; 16#0100 - прочитати конфігурацію в буфер; 16#301 - форсувати один/усі об'єкти типу; 16#302 - дефорсувати один/усі об'єкти типу; 16#0300 - перемкнути форсування |
| VAL | INT | | значення: - для CHAI значення з аналогового входу для AIVAR;- для CHAO значення на аналоговий вихід з |
| ТО8151.0014.001.АТХ.П | | | |
| Зм. | Лист | № докум. | Підпис |
| | | | Дата |
| | | | Арк |
| | | | 34 |

| | | | |
|---------|------|---|--|
| INVERSE | BOOL | 2 | 1 – інвертувати сире значення |
| NRMVAL | BOOL | 5 | значення норми |
| QALENBL | BOOL | 6 | 1 – задіяти тривогу недостовірності каналу |
| DSBL | BOOL | 7 | 1 – змінна не задіяна |
| CHID | UINT | | Логічний номер дискретного каналу, до якого прив'язана змінна, 0 - немає прив'язки |
| STEP1 | UINT | | номер кроку |
| T_DEASP | UINT | | Час затримки тривоги в секундах |
| T_FLTSP | UINT | | Заданий час фільтрації в мілісекундах |
| TMP | UINT | | резерв |
| T_STEP1 | TIME | | Плинний час кроку в мс |
| T_PREV | TIME | | час в мс з попереднього виклику, береться зі структури PLC_CFG.TQMS |

Приклад конфігурування та спостереження за технологічними змінами (структура яких наведено у таблиці 3) на НМІ показаний на рис. 2.8 та рис. 2.9. Такі вікна називаються картою технологічних змінних.

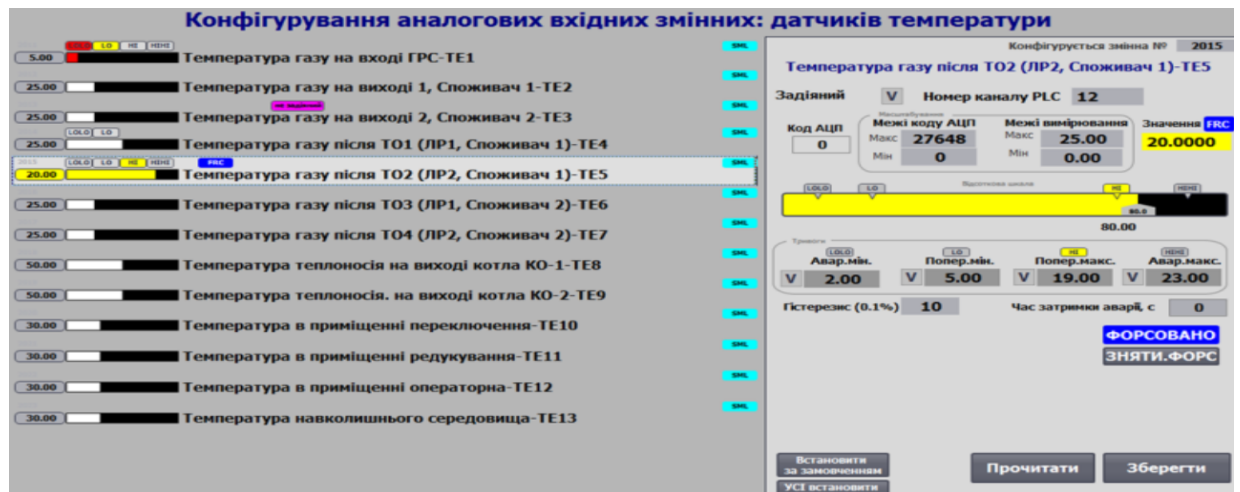


Рисунок 2.8 - Приклад використання функцій аналогових вхідних змінних на НМІ.



Рисунок 2.9 - Приклад використання функцій аналогових вихідних змінних на НМІ.

2.6. Опис класів LVL2 частини ПТК каркасу

Загальний опис класів LVL2.

Рівень пристроїв представлений чотирма основними класами, які реалізують функціонал цього рівню:

- 1) VLVSrv - клапани типу МЕО;
- 2) DRV - двигуни;
- 3) VLVD - клапани з дискретним керуванням.

Оскільки рівень пристроїв взаємодіє з рівнем технологічних змінних, елементи керування виконавчим механізмом подаються як змінні типу DIVAR та DOVAR. Якщо якісь елементи не використовуються, то їх значення подаються як пусті з індексом 0.

Таблиця 2.4 - Структура змінної типу VLVD_CFG.

| Атрибут | Тип | Біт | Опис |
|---------|------|-----|---|
| ID | | | |
| CLSID | | | 16#2011 – одно стабільний режим 16#2012 – бістабільний режим |
| STA | | | |
| CMD | | | |
| PRM | UINT | b0 | |

| | | | |
|--------|------|------------|--|
| | | 2 ZOPNENBL | X2 - =1 є датчик відкритості |
| | | 3 ZCLSENBL | X3 - =1 є датчик закритості |
| | | 4 BISTABLE | X4 - =1 |
| | | b5 | X5 – =1 клапан без нормального положення (додатково використовується змінна "закрити") |
| | | 6 MANCFG | X6 – =1 ручне конфігурування типу клапану та типу керування (при зміні клапану в режимі роботи) |
| | | b7 | X7 – =1, використання клапану (0 – резерв) |
| | | 8 AUTOACK | X8 - =1, автоматичне підтвердження тривоги |
| MSG | UINT | | Номер останнього діагностичного повідомлення |
| TDEASP | UINT | | Час затримки тривоги в секундах |
| STEP1 | UINT | | крок: 0-невизначений, 2-відкривається, 3-закривається, 1-зупинений в невизначеності, 4 – зупинений у відкритому стані, 5 – зупинений в закритому стані, 6 – заблокований |
| TSTEP1 | UINT | | Час кроку в с |
| TOPNSP | UINT | | Максимальний час відкриття в секундах |
| CNTPER | UINT | | Кількість змін положення |
| CNTALM | UINT | | Кількість аварій |

4) VLVA - клапани з аналоговим керуванням.

Таблиця 2.5 - Структура змінної типу VLVA_CFG.

| Атрибут | Тип | Біт | Опис |
|---------|------|------------|--|
| ID | UINT | | |
| CLSID | UINT | | 16#2020 |
| STA | UINT | | |
| CMD | UINT | | |
| VAL | REAL | | значення |
| PRM | UINT | b0 | |
| | | 2 ZOPNENBL | |
| | | 3 ZCLSENBL | |
| | | b5 | X5 – використати обмеження по швидкості |
| | | 6 MANCFG | |
| | | b7 | X7 – =1, використання клапану (0 – резерв) |

| | | | |
|---------|------|--------------|---|
| | | 8 AUTOACK | X8 – =1, автоматичне підтвердження тривоги |
| | | 11 SPDLMENBL | X11 – =1, включити обмеження по швидкості |
| MSG | UINT | | Номер останнього діагностичного повідомлення |
| TDEASP | UINT | | Час затримки тривоги в секундах |
| STEP1 | UINT | | Крок |
| VALPREV | REAL | | Значення VAL на попередньому перерахунку (2 секунди) |
| MVAL | REAL | | Значення VAL у ручному/форсованому режимі перезаписує VAL |
| CNTPER | UINT | | Кількість змін положення |
| CNTALM | UINT | | Кількість аварій |
| T_STEP1 | UINT | | Час кроку в с |
| T_OPN | UINT | | Час повного відкриття клапану при обмеженні на швидкість |

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Лист | № докум. | Підпис | Дата |

ТО8151.0014.001.АТХ.П

Арк

39

3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ БІБЛІОТЕКИ

3.1. Загальний опис програмного забезпечення

У даному розділі наведена програмна реалізація бібліотеки PACFramework для ПЛК PLCnext. Програма розроблена в середовищі PLCnext Engineer. Основна мова – ST.

Для кожних рівнів каркасу було розроблені типи даних відповідно до вимог каркасу, які наведені на рис.3.1.

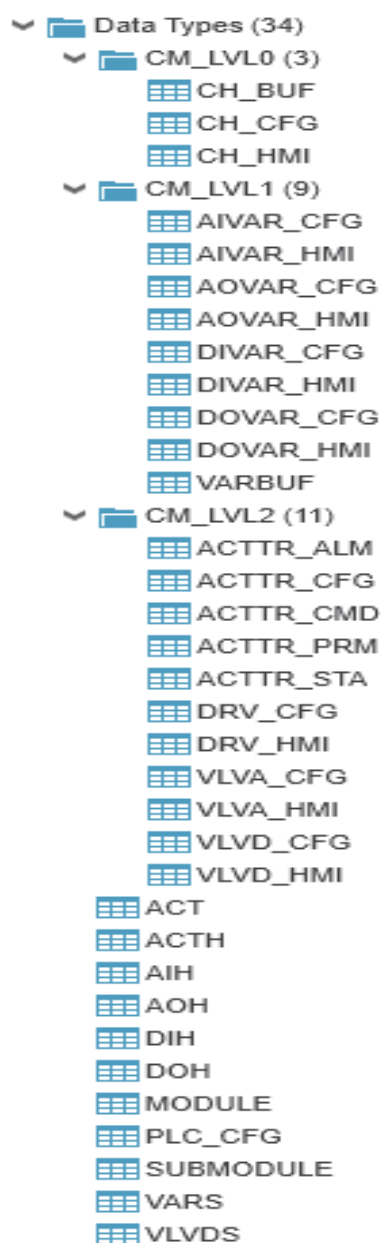


Рисунок 3.1 - Розроблені типи даних каркасу

Розгортання каркасу розпочалось з написання головної функції PLCFN, яка наведена в п.3.2. Далі було розроблено функціональні блоки для рівнів каналів (LVL0), код для яких наведено в п.3.3. Розроблені функціональні блоки для рівню технологічних змінних (LVL1) наведені в п.3.4 а для рівню пристроїв (LVL2) в п.3.5.

3.2. PLCFN

```

IF PLC.STA1.CMDACK THEN
    PLC.CMD := 0;
END_IF;

PLC.STA1.CMDACK := (PLC.CMD <> 0 AND NOT PLC.STA1.CMDACK);

IF NOT FirstScan AND NOT NextScan THEN
    FirstScan := TRUE;
    NextScan:=TRUE;
ELSE
    FirstScan := FALSE;
END_IF

    PLC.STA1.SCN1 := FirstScan;

IF PLC.STA1.SCN1 THEN (*при першому скані*)
    PLC.ID := 1;          (*Ідентифікатор завжди 1*)
    PLC.CLSID := 16#2100; (*Клас завжди 16#2100*)
    PLC.TQ := 0;
    TMBITS_CUR := 0;
    TMBITS_PREV := 0;
    PLC.TQMS := 0;
END_IF;

(*16#301 - включити режим деблокування усіх ВМ;
16#302 - відключити режим деблокування усіх ВМ;
16#300 - переключити режим деблокування усіх ВМ;*)
CASE PLC.CMD OF
    16#300: (*768*)
        PLC.STA1.DBLCALL := NOT PLC.STA1.DBLCALL;
    16#301: (*769*)
        PLC.STA1.DBLCALL := true;
    16#302: (*770*)
        PLC.STA1.DBLCALL := false;
    ELSE
        ;
END_CASE;

PLC.STA_PERM.%X0 := PLC.STA1.ACON2ERR;
PLC.STA_PERM.%X1 := PLC.STA1.APLC2STOP;
PLC.STA_PERM.%X2 := PLC.STA1.BLK;
PLC.STA_PERM.%X3 := PLC.STA1.ALDIS;
PLC.STA_PERM.%X4 := PLC.STA1.DIOON;
PLC.STA_PERM.%X5 := PLC.STA1.DIOERR;
PLC.STA_PERM.%X6 := PLC.STA1.DBLCALL;
PLC.STA_PERM.%X7 := PLC.STA1.FRC0 OR PLC.STA1.FRC1 OR PLC.STA1.FRC2;
PLC.STA_PERM.%X8 := PLC.STA1.SMLALL;
PLC.STA_PERM.%X9 := PLC.STA1.DISP;
PLC.STA_PERM.%X10 := PLC.STA1.FRC2;
PLC.STA_PERM.%X11 := PLC.STA1.FRC1;
PLC.STA_PERM.%X12 := PLC.STA1.SCN1;

```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 41 |

```

PLC.STA_PERM.%X13 := PLC.STA1.FRC0;
PLC.STA_PERM.%X14 := PLC.STA1.SML;
PLC.STA_PERM.%X15 := PLC.STA1.CMDACK;

PLC.ALM1_PERM.%X0 := PLC.ALM.ALM;
PLC.ALM1_PERM.%X1 := PLC.ALM.NWALM;
PLC.ALM1_PERM.%X2 := PLC.ALM.ALMNACK;
PLC.ALM1_PERM.%X3 := PLC.ALM.WRN;
PLC.ALM1_PERM.%X4 := PLC.ALM.NWWRN;
PLC.ALM1_PERM.%X5 := PLC.ALM.WRNNACK;
PLC.ALM1_PERM.%X6 := PLC.ALM.BAD;
PLC.ALM1_PERM.%X7 := PLC.ALM.NWBAD;
PLC.ALM1_PERM.%X8 := PLC.ALM.BADNACK;
PLC.ALM1_PERM.%X9 := PLC.ALM.EMCYSTP;
PLC.ALM1_PERM.%X10 := PLC.ALM.STP2RUN;
PLC.ALM1_PERM.%X11 := PLC.ALM.CON2ERR;
PLC.ALM1_PERM.%X12 := PLC.ALM.PLC2STOP;
PLC.ALM1_PERM.%X13 := PLC.ALM.DIOERR;
PLC.ALM1_PERM.%X14 := PLC.ALM.PLCERR;
PLC.ALM1_PERM.%X15 := PLC.ALM.CONHIERR;

(*----- таймерні біти та лічильники*)
BLINK_10Hz(enable:=TRUE, timelow:=T#50MS, timehigh:=T#50MS, out=>Clock_10Hz);
BLINK_5Hz(enable:=TRUE, timelow:=T#100MS, timehigh:=T#100MS, out=>Clock_5Hz);
BLINK_2Hz(enable:=TRUE, timelow:=T#250MS, timehigh:=T#250MS, out=>Clock_2Hz);
BLINK_1Hz(enable:=TRUE, timelow:=T#500MS, timehigh:=T#500MS, out=>Clock_1Hz);
BLINK_0_5Hz(enable:=TRUE, timelow:=T#1000MS, timehigh:=T#1000MS,
out=>Clock_0_5Hz);
(*плинні значення*)
TMBITS_CUR.%X0 := Clock_10Hz; // 100MS
TMBITS_CUR.%X1 := Clock_5Hz; // 200MS
TMBITS_CUR.%X2 := Clock_2Hz; // 500MS
TMBITS_CUR.%X3 := Clock_1Hz; // 1S
TMBITS_CUR.%X7 := PLC.NOW[0] > 16#29; //min

(*відпловування імпульсів необхідних часових періодів*)
PLC.PLS1.P100MS := TMBITS_CUR.%X0 AND NOT TMBITS_PREV.%X0; (*100 мс *)
PLC.PLS1.P200MS := TMBITS_CUR.%X1 AND NOT TMBITS_PREV.%X1; (*200 мс*)
PLC.PLS1.P500MS := TMBITS_CUR.%X2 AND NOT TMBITS_PREV.%X2; (*500 мс*)
PLC.PLS1.P1S := TMBITS_CUR.%X3 AND NOT TMBITS_PREV.%X3; (*1 с*)
PLC.PLS1.P60S := TMBITS_CUR.%X7 AND NOT TMBITS_PREV.%X7; (*1 хв*)
(*підрахунок кількості секунд і хвилин*)
IF PLC.PLS1.P1S THEN
    PLC.TQ := PLC.TQ + 1;
END_IF; (*загальний час з початку 1-го циклу контролера (в секундах)*)
IF PLC.TQ > 16#7FFF_FFFF THEN
    PLC.TQ := 16#7FFF_FFFF;
END_IF;
IF PLC.PLS1.P60S AND NOT PLC.STA1.SCN1 THEN
    PLC.TQM := PLC.TQM + 1;
END_IF; (*загальний час роботи ПЛК з моменту пуску (в хвилинали)*)
IF PLC.TQM > 16#7FFF_FFFF THEN
    PLC.TQM := 16#7FFF_FFFF;
END_IF;
(*розрахунок інших часових періодів*)
PLC.PLS1.P2S := (PLC.TQ MOD 2) = 0 AND PLC.PLS1.P1S; (*2 с*)
PLC.PLS1.P5S := (PLC.TQ MOD 5) = 0 AND PLC.PLS1.P1S; (*5 с*)
PLC.PLS1.P10S := (PLC.TQ MOD 10) = 0 AND PLC.PLS1.P1S; (*10 с*)
(* меандри *)
PLC.PLS1.M1S := Clock_1Hz; (*меандр з періодом 1 с (0.5 с + 0.5 с)*)
PLC.PLS1.M2S := Clock_0_5Hz; (*меандр з періодом 2 с (1 с + 1 с) *)
(* астрономічний час *)
PLC.NOW[0] := TO_BCD_WORD(RTC.SECONDS);
PLC.NOW[1] := SHL(TO_BCD_WORD(RTC.HOURS), 8) OR TO_BCD_WORD(RTC.MINUTES); //16hmm
PLC.NOW[2] := SHL(TO_BCD_WORD(RTC.MONTH), 8) OR TO_BCD_WORD(RTC.DAY); //16mddd

```

```

PLC.NOW[3] := SHL(TO_BCD_WORD(RTC.YEAR / 100), 8) OR TO_BCD_WORD(RTC.YEAR MOD
100);; //16уууу
(* початок години *)
PLC.PLS1.NEWHR := PLC.NOW[1] = 0 (*хвилини*)AND PLC.PLS1.P60S (*один раз за
хвилину*);
(* початок доби*)
PLC.PLS1.NEWDAY := PLC.NOW[1] = 0 (*години_хвилини*)AND PLC.PLS1.P60S (*один раз
за хвилину*);
(*початок зміни*)
IF PLC.SHIFT PARA[0] < 1 OR PLC.SHIFT PARA[0] > 3 THEN
    PLC.SHIFT PARA[0] := 2;
END_IF; (*коректність кількості змін*)

PLC.PLS1.NEWSHIFT := FALSE; (*скидаємо біт початку зміни*)

FOR i := 1 TO TO_INT(PLC.SHIFT PARA[0]) DO
    (*хоча б одна з уставок змін спрацювала*)
    PLC.PLS1.NEWSHIFT := PLC.PLS1.NEWSHIFT OR (PLC.SHIFT PARA[i] = PLC.NOW[1]
(*години хвилини*)AND PLC.PLS1.P60S (*один раз за хвилину*));
END_FOR;
TMBITS_PREV := TMBITS_CUR; (*збереження попередніх значень*)
(*----- визначення зміни*)
IF PLC.SHIFT PARA[0] > 3 OR PLC.SHIFT PARA[0] < 2 THEN
    PLC.SHIFT PARA[0] := 3;
END_IF; (*якщо кількість змін >3 або <1 - робимо 3*)

IF PLC.NOW[1] >= PLC.SHIFT PARA[1] AND (PLC.NOW[1] < PLC.SHIFT PARA[2] OR
PLC.SHIFT PARA[2] = 16#0000) THEN (*1-ша зміна*)
    PLC.SHIFTNMB := 1;
ELSE (*2-га або 3-тя*)
    IF PLC.SHIFT PARA[0] = 2 THEN (*якщо в 2 зміни*)
        PLC.SHIFTNMB := 2;
    ELSE (*якщо 3 зміни*)
        IF PLC.NOW[1] >= PLC.SHIFT PARA[2] AND (PLC.NOW[1] < PLC.SHIFT PARA[3] OR
PLC.SHIFT PARA[3] = 16#0000) THEN
            PLC.SHIFTNMB := 2;
        ELSE
            PLC.SHIFTNMB := 3;
        END_IF;
    END_IF;
END_IF;
(*цикли*)
PLC.TSK_LTIME := LINT_TO_UINT(ESM_DATA.ESM_INFOS[1].TASK_INFOS[1].INTERVAL / 1000);
(*ms*)

IF NOT FirstScan THEN
    //обмеження
    IF PLC.TSK_MAXTIME > 3000 THEN
        PLC.TSK_MAXTIME := 0;
    END_IF;
    IF PLC.TSK_MAXTIME < PLC.TSK_LTIME THEN
        PLC.TSK_MAXTIME := PLC.TSK_LTIME;
    END_IF;
ELSE
    PLC.TSK_MAXTIME := 0;
END_IF;
PLC.TQMS := PLC.TQMS + PLC.TSK_LTIME; //лічильник в мс
(*скидання статусів і тривоги*)
PLC.STA1.BLK := false;
PLC.STA1.ALDIS := false;
PLC.STA1.FRC := false;
PLC.STA1.SMLALL := false;
PLC.STA1.DISP := false;

```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 43 |

```

PLC.STA1.FRC2 := false;
PLC.STA1.FRC1 := false;
PLC.STA1.FRC0 := false;
PLC.STA1.SML := false;

PLC.ALM.ALM := 0;
PLC.ALM.NWALM := 0;
PLC.ALM.ALMNACK := 0;
PLC.ALM.WRN := 0;
PLC.ALM.NWWRN := 0;
PLC.ALM.WRNNACK := 0;
PLC.ALM.BAD := 0;
PLC.ALM.NWBAD := 0;
PLC.ALM.BADNACK := 0;
PLC.ALM.EMCYSTP := 0;
PLC.ALM.STP2RUN := 0;
PLC.ALM.CON2ERR := 0;
PLC.ALM.PLC2STOP := 0;
PLC.ALM.DIOERR := 0;
PLC.ALM.PLCERR := 0;
PLC.ALM.CONHIERR := 0;
(*запам'ятовування лічильників*)
PLC.CNTALM_PERM := PLC.CNTALM;
PLC.CNTWRN_PERM := PLC.CNTWRN;
PLC.CNTBAD_PERM := PLC.CNTBAD;
PLC.CNTFRC_PERM := PLC.CNTFRC;
PLC.CNTMAN_PERM := PLC.CNTMAN;
(*скидання лічильників*)
PLC.CNTALM := 0;
PLC.CNTWRN := 0;
PLC.CNTBAD := 0;
PLC.CNTFRC := 0;
PLC.CNTMAN := 0;

```

3.3. Рівень каналів

Рівень каналів (LVL0) реалізований як 4 функціональні блоки (рис.3.2):

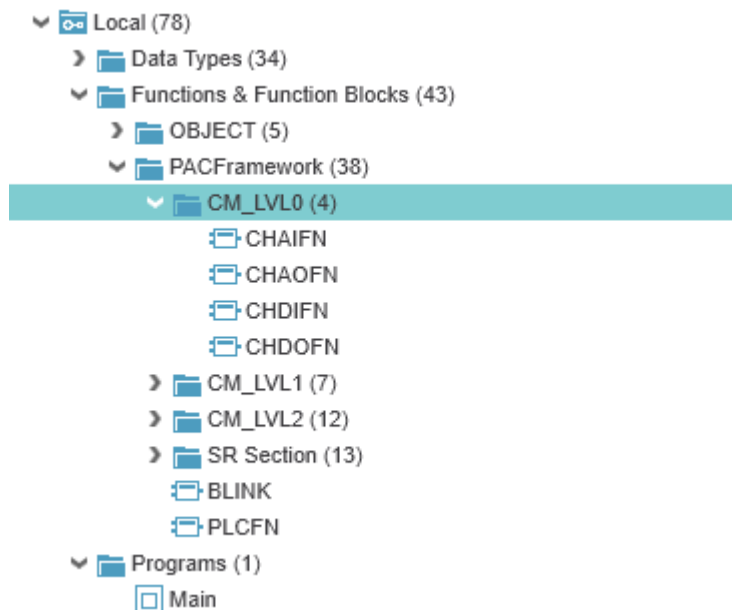


Рисунок 3.2 - Зовнішній вигляд структури функцій LVL0

CHAI FN

Код реалізації функціонального блоку CHAI FN:

```
MIN:=0;
MAX:=10000;
MIDDLE:=5000;

CMD := CHCFG.CMD;
(*розпаковка з STA*)
VRAW := CHCFG.STA1.STA_VRAW;
VAL := CHCFG.STA1.STA_VALB;
BAD := CHCFG.STA1.STA_BAD; (*керується ззовні*)
PNG := CHCFG.STA1.STA_PNG;
ULNK := CHCFG.STA1.STA_ULNK;
MERR := CHCFG.STA1.STA_MERR; (*керується ззовні*)
BRK := CHCFG.STA1.STA_BRK; (*керується ззовні*)
SHRT := CHCFG.STA1.STA_SHRT; (*керується ззовні*)
NBD := CHCFG.STA1.STA_NBD; (*керується ззовні*)
INBUF := CHCFG.STA1.STA_INBUF;
FRC := CHCFG.STA1.STA_FRC;
SML := CHCFG.STA1.STA_SML;
CMDLOAD := CHCFG.STA1.STA_CMDLOAD; (*керується бітом*)

INBUF := (CHCFG.ID = CHBUF.ID) AND (CHCFG.CLSID = CHBUF.CLSID);
CMDLOAD := CHHMI.STA1.STA_CMDLOAD;
CMD := 0;

(* обробник команд*)
(* ширококомвне форсування/дефорсування*)
IF PLCCFG.CMD=16#4301 THEN
    FRC := true; (*форсувати один/усі об'єкти типу*)
END_IF;
IF PLCCFG.CMD=16#4302 THEN
    FRC := false; (*дефорсувати об'єкт типу*)
END_IF;
(*вибір джерела команди згідно пріоритету*)
IF CMDLOAD THEN (*з HMI CMDLOAD*)
    CMD := 16#0100; (*// записати в буфер*)
ELSIF INBUF AND CHBUF.CMD <> 0 THEN (*з буферу*)
    CMD := CHBUF.CMD;
ELSIF CHCFG.CMD <> 0 THEN
    CMD := CHCFG.CMD;
END_IF;

(*commands*)
CASE CMD OF
    16#1: (*записати MAX*)
        IF FRC AND INBUF THEN
            CHBUF.VAL := TO_WORD(MAX);
        END_IF;
    16#2: (*записати MIN*)
        IF FRC AND INBUF THEN
            CHBUF.VAL:=TO_WORD(MIN);
        END_IF;
    16#3: (*TOGGLE*)
        IF FRC AND INBUF THEN
            CHBUF.VAL := TO_WORD(MIDDLE);
        END_IF;
    16#0100: (*прочитати конфігурацію в буфер*)
        CHBUF.ID:= CHCFG.ID;
        CHBUF.CLSID:=CHCFG.CLSID;
```

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Лист | № докум. | Підпис | Дата |

ТО8151.0014.001.АТХ.П

Арк

45

```

CHBUF.STA1.STA_VRAW := CHCFG.STA1.STA_VRAW;
CHBUF.STA1.STA_VALB := CHCFG.STA1.STA_VALB;
CHBUF.STA1.STA_BAD := CHCFG.STA1.STA_BAD;
CHBUF.STA1.STA_b3 := CHCFG.STA1.STA_b3;
CHBUF.STA1.STA_PNG := CHCFG.STA1.STA_PNG;
CHBUF.STA1.STA_ULNK := CHCFG.STA1.STA_ULNK;
CHBUF.STA1.STA_MERR := CHCFG.STA1.STA_MERR;
CHBUF.STA1.STA_BRK := CHCFG.STA1.STA_BRK;
CHBUF.STA1.STA_SHRT := CHCFG.STA1.STA_SHRT;
CHBUF.STA1.STA_NBD := CHCFG.STA1.STA_NBD;
CHBUF.STA1.STA_b10 := CHCFG.STA1.STA_b10;
CHBUF.STA1.STA_INIOTBUF := CHCFG.STA1.STA_INIOTBUF;
CHBUF.STA1.STA_INBUF := CHCFG.STA1.STA_INBUF;
CHBUF.STA1.STA_FRC := CHCFG.STA1.STA_FRC;
CHBUF.STA1.STA_SML := CHCFG.STA1.STA_SML;
CHBUF.STA1.STA_CMDLOAD := CHCFG.STA1.STA_CMDLOAD;

CHBUF.VAL:=CHCFG.VAL;
CHBUF.VARID:=CHCFG.VARID;
16#0300: (*перемкнути форсування*)
    FRC := NOT FRC;
16#301: (*форсувати один/усі об'єкти типу*)
    FRC := true;
16#302: (*дефорсувати один/усі об'єкти типу*)
    FRC := false;
END_CASE;

(*запис значення змінної*)
IF FRC AND INBUF THEN (*режим форсування з занятим буфером*)
    CHCFG.VAL := CHBUF.VAL;
ELSIF FRC AND NOT INBUF THEN (*режим форсування без занятого буферу*)
    ;(*без змін*)
ELSIF NOT FRC THEN (*не режим форсування*)
    CHCFG.VAL := TO_WORD(RAWINT);
END_IF;

(*ping-pong*)
ULNK := PNG; (*прийшов ping - є звязок з верхнім рівнем*)
PNG := false; (*скидання біту PNG звязку з врехнім рівнем PONG*)
IF NOT ULNK THEN
    CHCFG.VARID := 0;
END_IF;

(*скидання оброблених команд*)
CMDLOAD := 0;
CMD := 0;

(*загальносистемні біти та лічильники*)
IF FRC THEN
    PLCCFG.STA1.FRC0 := true;
    PLCCFG.CNTFRC := PLCCFG.CNTFRC + 1;
END_IF;

BAD:= BRK OR SHRT;

(*упковка в STA*)
CHCFG.STA1.STA_VRAW := RAWINT > 0;
CHCFG.STA1.STA_VALB:=CHCFG.VAL > 0;
CHCFG.STA1.STA_BAD:=BAD; (*керується ззовні*)
CHCFG.STA1.STA_PNG:=PNG;
CHCFG.STA1.STA_ULNK:=ULNK;
CHCFG.STA1.STA_MERR:=MERR; (*керується ззовні*)
CHCFG.STA1.STA_BRK:=BRK; (*керується ззовні*)
CHCFG.STA1.STA_SHRT:=SHRT; (*керується ззовні*)
CHCFG.STA1.STA_NBD:=NBD; (*керується ззовні*)

```

```

CHCFG.STA1.STA_INBUF := INBUF;
CHCFG.STA1.STA_FRC := FRC;
CHCFG.STA1.STA_SML := SML;
CHCFG.STA1.STA_CMDLOAD := CMDLOAD; (*керується бітом*)
CHCFG.CMD := CMD;
CHHMI.STA1 := CHCFG.STA1;
CHHMI.VAL := CHCFG.VAL;
(*оновлення буферу*)
IF INBUF THEN
    CHBUF.STA1:= CHCFG.STA1;
    CHBUF.VARID := CHCFG.VARID;
    CHBUF.CMD := 0;
    IF NOT FRC THEN
        CHBUF.VAL := CHCFG.VAL;
    END_IF;
END_IF;

```

СНАОFN

Код реалізації функціонального блоку СНАОFN:

```

MIN:=0;
MAX:=10000;
MIDDLE:=5000;
CMD := CHCFG.CMD;
(*розпаковка з STA*)
VRAW := CHCFG.STA1.STA_VRAW;
VAL := CHCFG.STA1.STA_VALB;
BAD := CHCFG.STA1.STA_BAD; (*керується ззовні*)
PNG := CHCFG.STA1.STA_PNG;
ULNK := CHCFG.STA1.STA_ULNK;
MERR := CHCFG.STA1.STA_MERR; (*керується ззовні*)
BRK := CHCFG.STA1.STA_BRK; (*керується ззовні*)
SHRT := CHCFG.STA1.STA_SHRT; (*керується ззовні*)
NBD := CHCFG.STA1.STA_NBD; (*керується ззовні*)
INBUF := CHCFG.STA1.STA_INBUF;
FRC := CHCFG.STA1.STA_FRC;
SML := CHCFG.STA1.STA_SML;
CMDLOAD := CHCFG.STA1.STA_CMDLOAD; (*керується бітом*)
INBUF := (CHCFG.ID = CHBUF.ID) AND (CHCFG.CLSID = CHBUF.CLSID);
CMDLOAD := CHHMI.STA1.STA_CMDLOAD;
CMD := 0;

(* обробник команд*)
(* широкомовне форсування/дефорсування*)
IF PLCCFG.CMD=16#4301 THEN
    FRC := true; (*форсувати один/усі об'єкти типу*)
END_IF;
IF PLCCFG.CMD=16#4302 THEN
    FRC := false; (*дефорсувати об'єкт типу*)
END_IF;
(*вибір джерела команди згідно пріоритету*)
IF CMDLOAD THEN (*з HMI CMDLOAD*)
    CMD := 16#0100; (*// записати в буфер*)
ELSIF INBUF AND CHBUF.CMD <> 0 THEN (*з буферу*)
    CMD := CHBUF.CMD;
ELSIF CHCFG.CMD <> 0 THEN
    CMD := CHCFG.CMD;
END_IF;
(*commands*)
CASE CMD OF

```

```

16#1: (*записати MAX*)
  IF FRC AND INBUF THEN
    RAWINT := MAX;
  END_IF;
16#2: (*записати MIN*)
  IF FRC AND INBUF THEN
    RAWINT:=MIN;
  END_IF;
16#3: (*TOGGLE*)
  IF FRC AND INBUF THEN
    RAWINT := MIDDLE;
  END_IF;
16#0100: (*прочитати конфігурацію в буфер*)
  CHBUF.ID:= CHCFG.ID;
  CHBUF.CLSID:=CHCFG.CLSID;

  CHBUF.STA1.STA_VRAW := CHCFG.STA1.STA_VRAW;
  CHBUF.STA1.STA_VALB := CHCFG.STA1.STA_VALB;
  CHBUF.STA1.STA_BAD := CHCFG.STA1.STA_BAD;
  CHBUF.STA1.STA_b3 := CHCFG.STA1.STA_b3;
  CHBUF.STA1.STA_PNG := CHCFG.STA1.STA_PNG;
  CHBUF.STA1.STA_ULNK := CHCFG.STA1.STA_ULNK;
  CHBUF.STA1.STA_MERR := CHCFG.STA1.STA_MERR;
  CHBUF.STA1.STA_BRK := CHCFG.STA1.STA_BRK;
  CHBUF.STA1.STA_SHRT := CHCFG.STA1.STA_SHRT;
  CHBUF.STA1.STA_NBD := CHCFG.STA1.STA_NBD;
  CHBUF.STA1.STA_b10 := CHCFG.STA1.STA_b10;
  CHBUF.STA1.STA_INIOTBUF := CHCFG.STA1.STA_INIOTBUF;
  CHBUF.STA1.STA_INBUF := CHCFG.STA1.STA_INBUF;
  CHBUF.STA1.STA_FRC := CHCFG.STA1.STA_FRC;
  CHBUF.STA1.STA_SML := CHCFG.STA1.STA_SML;
  CHBUF.STA1.STA_CMDLOAD := CHCFG.STA1.STA_CMDLOAD;
  CHBUF.VAL:=CHCFG.VAL;
  CHBUF.VARID:=CHCFG.VARID;
16#0300: (*перемкнути форсування*)
  FRC := NOT FRC;
16#301: (*форсувати один/усі об'єкти типу*)
  FRC := true;
16#302: (*дефорсувати один/усі об'єкти типу*)
  FRC := false;
END_CASE;
(*запис значення змінної*)
IF FRC AND INBUF THEN (*режим форсування з занятим буфером*)
  RAWINT := TO_INT(CHBUF.VAL);
ELSIF FRC AND NOT INBUF THEN (*режим форсування без занятого буферу*)
  ;(*без змін*)
ELSIF NOT FRC THEN (*не режим форсування*)
  RAWINT := TO_INT(CHCFG.VAL);
END_IF;
(*ping-pong*)
ULNK := PNG; (*прийшов ping - є звязок з верхнім рівнем*)
PNG := false; (*скидання біту PNG звязку з врехнім рівнем PONG*)
IF NOT ULNK THEN
  CHCFG.VARID := 0;
END_IF;

(*скидання оброблених команд*)
CMDLOAD := 0;
CMD := 0;
(*загальносистемні біти та лічильники*)
IF FRC THEN
  PLCCFG.STA1.FRC0 := true;
  PLCCFG.STA1.FRC := true;

```

```

END_IF;
(*упковка в STA*)
CHCFG.STA1.STA_VRAW := RAWINT > 0;
CHCFG.STA1.STA_VALB:=CHCFG.VAL > 0;
CHCFG.STA1.STA_BAD:=BAD; (*керується ззовні*)
CHCFG.STA1.STA_PNG:=PNG;
CHCFG.STA1.STA_ULNK:=ULNK;
CHCFG.STA1.STA_MERR:=MERR; (*керується ззовні*)
CHCFG.STA1.STA_BRK:=BRK; (*керується ззовні*)
CHCFG.STA1.STA_SHRT:=SHRT; (*керується ззовні*)
CHCFG.STA1.STA_NBD:=NBD; (*керується ззовні*)
CHCFG.STA1.STA_INBUF := INBUF;
CHCFG.STA1.STA_FRC := FRC;
CHCFG.STA1.STA_SML := SML;
CHCFG.STA1.STA_CMDLOAD := CMDLOAD; (*керується бітом*)
CHCFG.CMD := CMD;
CHHMI.STA1 := CHCFG.STA1;
CHHMI.VAL := TO_WORD(RAWINT);
(*оновлення буферу*)
IF INBUF THEN
    CHBUF.STA1:= CHCFG.STA1;
    CHBUF.VARID := CHCFG.VARID;
    CHBUF.CMD := 0;
    IF NOT FRC THEN
        CHBUF.VAL := CHCFG.VAL;
    END_IF;
END_IF;

```

CHDIFN

Код реалізації функціонального блоку CHDIFN:

```

CMD := CHCFG.CMD;
(*розпаковка з STA*)
VRAW := CHCFG.STA1.STA_VRAW;
VAL := CHCFG.STA1.STA_VALB;
BAD := CHCFG.STA1.STA_BAD; (*керується ззовні*)
PNG := CHCFG.STA1.STA_PNG;
ULNK := CHCFG.STA1.STA_ULNK;
MERR := CHCFG.STA1.STA_MERR; (*керується ззовні*)
BRK := CHCFG.STA1.STA_BRK; (*керується ззовні*)
SHRT := CHCFG.STA1.STA_SHRT; (*керується ззовні*)
NBD := CHCFG.STA1.STA_NBD; (*керується ззовні*)
INBUF := CHCFG.STA1.STA_INBUF;
FRC := CHCFG.STA1.STA_FRC;
SML := CHCFG.STA1.STA_SML;
CMDLOAD := CHCFG.STA1.STA_CMDLOAD; (*керується бітом*)
INBUF := (CHCFG.ID = CHBUF.ID) AND (CHCFG.CLSID = CHBUF.CLSID);
CMDLOAD := CHHMI.STA1.STA_CMDLOAD;
CMD := 0;

(* обробник команд*)
(* широкомовне форсування/дефорсування*)
IF PLCCFG.CMD=16#4301 THEN
    FRC := true; (*форсувати один/усі об'єкти типу*)
END_IF;
IF PLCCFG.CMD=16#4302 THEN
    FRC := false; (*дефорсувати об'єкт типу*)
END_IF;
(*вибір джерела команди згідно пріоритету*)
IF CMDLOAD THEN (*з HMI CMDLOAD*)

```

| | | | | | | | |
|-----|------|----------|--------|------|--|-----------------------|-----|
| | | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | | 49 |

```

CMD := 16#0100; (*// записати в буфер*)
ELSIF INBUF AND CHBUF.CMD <> 0 THEN (*з буферу*)
    CMD := CHBUF.CMD;
ELSIF CHCFG.CMD <> 0 THEN
    CMD := CHCFG.CMD;
END_IF;
(*commands*)
CASE CMD OF
    16#1: (*записати 1*)
        IF FRC AND INBUF THEN
            CHBUF.VAL := 1;
        END_IF;
    16#2: (*записати 0*)
        IF FRC AND INBUF THEN
            CHBUF.VAL:=0;
        END_IF;
    16#3: (*TOGGLE*)
        IF FRC AND INBUF THEN
            IF VAL THEN
                CHBUF.VAL:=0;
            ELSE
                CHBUF.VAL:=1;
            END_IF;
        END_IF;
    16#0100: (*прочитати конфігурацію в буфер*)
        CHBUF.ID:= CHCFG.ID;
        CHBUF.CLSID:=CHCFG.CLSID;

        CHBUF.STA1.STA_VRAW := CHCFG.STA1.STA_VRAW;
        CHBUF.STA1.STA_VALB := CHCFG.STA1.STA_VALB;
        CHBUF.STA1.STA_BAD := CHCFG.STA1.STA_BAD;
        CHBUF.STA1.STA_b3 := CHCFG.STA1.STA_b3;
        CHBUF.STA1.STA_PNG := CHCFG.STA1.STA_PNG;
        CHBUF.STA1.STA_ULNK := CHCFG.STA1.STA_ULNK;
        CHBUF.STA1.STA_MERR := CHCFG.STA1.STA_MERR;
        CHBUF.STA1.STA_BRK := CHCFG.STA1.STA_BRK;
        CHBUF.STA1.STA_SHRT := CHCFG.STA1.STA_SHRT;
        CHBUF.STA1.STA_NBD := CHCFG.STA1.STA_NBD;
        CHBUF.STA1.STA_b10 := CHCFG.STA1.STA_b10;
        CHBUF.STA1.STA_INIOTBUF := CHCFG.STA1.STA_INIOTBUF;
        CHBUF.STA1.STA_INBUF := CHCFG.STA1.STA_INBUF;
        CHBUF.STA1.STA_FRC := CHCFG.STA1.STA_FRC;
        CHBUF.STA1.STA_SML := CHCFG.STA1.STA_SML;
        CHBUF.STA1.STA_CMDLOAD := CHCFG.STA1.STA_CMDLOAD;
        CHBUF.VAL:=CHCFG.VAL;
        CHBUF.VARID:=CHCFG.VARID;
    16#0300: (*перемкнути форсування*)
        FRC := NOT FRC;
    16#0301: (*форсувати один/усі об'єкти типу*)
        FRC := true;
    16#0302: (*дефорсувати один/усі об'єкти типу*)
        FRC := false;
END_CASE;
(*запис значення змінної*)
IF FRC AND INBUF THEN (*режим форсування з занятим буфером*)
    CHCFG.VAL := CHBUF.VAL;
    VAL := TO_REAL(TO_INT(CHCFG.VAL)) > 0.5;
ELSIF FRC AND NOT INBUF THEN (*режим форсування без занятого буферу*)
    ;(*без змін*)
ELSIF SML THEN (*режим імітації*)
    ;(*//нічого не робимо, значення змінюється ззовні алгоритму 28.10.2021 Пупена*)
ELSIF NOT FRC THEN (*не режим форсування*)
    VAL := RAW;

```

| | | | | | | | |
|-----|------|----------|--------|------|--|-----------------------|-----|
| | | | | | | ТО8151.0014.001.АТХ.П | Арк |
| | | | | | | | 50 |
| Зм. | Лист | № докум. | Підпис | Дата | | | |

```

IF RAW THEN
    CHCFG.VAL := 1;
ELSE
    CHCFG.VAL := 0;
END_IF;
END_IF;
(*ping-pong*)
ULNK := PNG; (*прийшов ping - є звязок з верхнім рівнем*)
PNG := false; (*скидання біту PNG звязку з врехнім рівнем PONG*)
IF NOT ULNK THEN
    CHCFG.VARID := 0;
END_IF;
(*скидання оброблених команд*)
CMDLOAD := 0;
CMD := 0;
(*загальносистемні біти та лічильники*)
IF FRC THEN
    PLCCFG.STA1.FRC0 := true;
    PLCCFG.CNTFRC := PLCCFG.CNTFRC + 1;
END_IF;
(*упковка в STA*)
CHCFG.STA1.STA_VRAW := RAW;
CHCFG.STA1.STA_VALB:=VAL;
CHCFG.STA1.STA_BAD:=BAD; (*керується ззовні*)
CHCFG.STA1.STA_PNG:=PNG;
CHCFG.STA1.STA_ULNK:=ULNK;
CHCFG.STA1.STA_MERR:=MERR; (*керується ззовні*)
CHCFG.STA1.STA_BRK:=BRK; (*керується ззовні*)
CHCFG.STA1.STA_SHRT:=SHRT; (*керується ззовні*)
CHCFG.STA1.STA_NBD:=NBD; (*керується ззовні*)
CHCFG.STA1.STA_INBUF := INBUF;
CHCFG.STA1.STA_FRC := FRC;
CHCFG.STA1.STA_SML := SML;
CHCFG.STA1.STA_CMDLOAD := CMDLOAD; (*керується бітом*)
CHCFG.CMD := CMD;
CHHMI.STA1 := CHCFG.STA1;
CHHMI.VAL := CHCFG.VAL;
(*оновлення буферу*)
IF INBUF THEN
    CHBUF.STA1:= CHCFG.STA1;
    CHBUF.VARID := CHCFG.VARID;
    CHBUF.CMD := 0;
    IF NOT FRC THEN
        CHBUF.VAL := CHCFG.VAL;
    END_IF;
END_IF;
END_IF;

```

CHDOFN

Код реалізації функціонального блоку CHDOFN:

```

CMD := CHCFG.CMD;
(*розпаковка з STA*)
VRAW := CHCFG.STA1.STA_VRAW;
VAL := CHCFG.STA1.STA_VALB;
BAD := CHCFG.STA1.STA_BAD; (*керується ззовні*)
PNG := CHCFG.STA1.STA_PNG;
ULNK := CHCFG.STA1.STA_ULNK;
MERR := CHCFG.STA1.STA_MERR; (*керується ззовні*)
BRK := CHCFG.STA1.STA_BRK; (*керується ззовні*)
SHRT := CHCFG.STA1.STA_SHRT; (*керується ззовні*)

```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 51 |

```

NBD := CHCFG.STA1.STA_NBD; (*керується ззовні*)
INBUF := CHCFG.STA1.STA_INBUF;
FRC := CHCFG.STA1.STA_FRC;
SML := CHCFG.STA1.STA_SML;
CMDLOAD := CHCFG.STA1.STA_CMDLOAD; (*керується бітом*)
INBUF := (CHCFG.ID = CHBUF.ID) AND (CHCFG.CLSID = CHBUF.CLSID);
CMDLOAD := CHHMI.STA1.STA_CMDLOAD;
CMD := 0;
(* обробник команд*)
(* широкомовне форсування/дефорсування*)
IF PLCCFG.CMD=16#4301 THEN
    FRC := true; (*форсувати один/усі об'єкти типу*)
END_IF;
IF PLCCFG.CMD=16#4302 THEN
    FRC := false; (*дефорсувати об'єкт типу*)
END_IF;
(*вибір джерела команди згідно пріоритету*)
IF CMDLOAD THEN (*з HMI CMDLOAD*)
    CMD := 16#0100; (*// записати в буфер*)
ELSIF INBUF AND CHBUF.CMD <> 0 THEN (*з буферу*)
    CMD := CHBUF.CMD;
ELSIF CHCFG.CMD <> 0 THEN
    CMD := CHCFG.CMD;
END_IF;
(*commands*)
CASE CMD OF
    16#1: (*записати 1*)
        IF FRC AND INBUF THEN
            CHBUF.VAL := 1;
        END_IF;
    16#2: (*записати 0*)
        IF FRC AND INBUF THEN
            CHBUF.VAL:=0;
        END_IF;
    16#3: (*TOGGLE*)
        IF FRC AND INBUF THEN
            IF VRAW THEN
                CHBUF.VAL:=0;
            ELSE
                CHBUF.VAL:=1;
            END_IF;
        END_IF;
    16#0100: (*прочитати конфігурацію в буфер*)
        CHBUF.ID:= CHCFG.ID;
        CHBUF.CLSID:=CHCFG.CLSID;
        CHBUF.STA1.STA_VRAW := CHCFG.STA1.STA_VRAW;
        CHBUF.STA1.STA_VALB := CHCFG.STA1.STA_VALB;
        CHBUF.STA1.STA_BAD := CHCFG.STA1.STA_BAD;
        CHBUF.STA1.STA_b3 := CHCFG.STA1.STA_b3;
        CHBUF.STA1.STA_PNG := CHCFG.STA1.STA_PNG;
        CHBUF.STA1.STA_ULNK := CHCFG.STA1.STA_ULNK;
        CHBUF.STA1.STA_MERR := CHCFG.STA1.STA_MERR;
        CHBUF.STA1.STA_BRK := CHCFG.STA1.STA_BRK;
        CHBUF.STA1.STA_SHRT := CHCFG.STA1.STA_SHRT;
        CHBUF.STA1.STA_NBD := CHCFG.STA1.STA_NBD;
        CHBUF.STA1.STA_b10 := CHCFG.STA1.STA_b10;
        CHBUF.STA1.STA_INIOTBUF := CHCFG.STA1.STA_INIOTBUF;
        CHBUF.STA1.STA_INBUF := CHCFG.STA1.STA_INBUF;
        CHBUF.STA1.STA_FRC := CHCFG.STA1.STA_FRC;
        CHBUF.STA1.STA_SML := CHCFG.STA1.STA_SML;
        CHBUF.STA1.STA_CMDLOAD := CHCFG.STA1.STA_CMDLOAD;
        CHBUF.VAL:=CHCFG.VAL;
        CHBUF.VARID:=CHCFG.VARID;

```

```

16#0300: (*перемкнути форсування*)
    FRC := NOT FRC;
16#0301: (*форсувати один/усі об'єкти типу*)
    FRC := true;
16#0302: (*дефорсувати один/усі об'єкти типу*)
    FRC := false;
END_CASE;

(*запис значення змінної*)
IF FRC AND INBUF THEN (*режим форсування з занятим буфером*)
    CHCFG.VAL := CHBUF.VAL;
    VRAW := TO_REAL(TO_INT(CHCFG.VAL)) > 0.5;
ELSIF FRC AND NOT INBUF THEN (*режим форсування без занятого буферу*)
    ;(*без змін*)
ELSIF NOT FRC THEN (*не режим форсування*)
    VRAW := VAL;
    IF VAL THEN
        CHCFG.VAL := 1;
    ELSE
        CHCFG.VAL := 0;
    END_IF;
END_IF;
RAW := VRAW; (*відправляємо на вихід*)
(*ping-pong*)
ULNK := PNG; (*прийшов ping - є звязок з верхнім рівнем*)
PNG := false; (*скидання біту PNG звязку з врехнім рівнем PONG*)
IF NOT ULNK THEN
    CHCFG.VARID := 0;
END_IF;
(*скидання оброблених команд*)
CMDLOAD := 0;
CMD := 0;

(*загальносистемні біти та лічильники*)
IF FRC THEN
    PLCCFG.STA1.FRC0 := true;
    PLCCFG.CNTFRC := PLCCFG.CNTFRC + 1;
END_IF;
(*упковка в STA*)
CHCFG.STA1.STA_VRAW := VRAW;
CHCFG.STA1.STA_VALB := VAL;
CHCFG.STA1.STA_BAD:=BAD; (*керується ззовні*)
CHCFG.STA1.STA_PNG:=PNG;
CHCFG.STA1.STA_ULNK:=ULNK;
CHCFG.STA1.STA_MERR:=MERR; (*керується ззовні*)
CHCFG.STA1.STA_BRK:=BRK; (*керується ззовні*)
CHCFG.STA1.STA_SHRT:=SHRT; (*керується ззовні*)
CHCFG.STA1.STA_NBD:=NBD; (*керується ззовні*)
CHCFG.STA1.STA_INBUF := INBUF;
CHCFG.STA1.STA_FRC := FRC;
CHCFG.STA1.STA_SML := SML;
CHCFG.STA1.STA_CMDLOAD := CMDLOAD; (*керується бітом*)
CHCFG.CMD := CMD;
CHHMI.STA1 := CHCFG.STA1;
CHHMI.VAL := CHCFG.VAL;
(*оновлення буферу*)
IF INBUF THEN
    CHBUF.STA1:= CHCFG.STA1;
    CHBUF.VARID := CHCFG.VARID;
    CHBUF.CMD := 0;
    IF NOT FRC THEN
        CHBUF.VAL := CHCFG.VAL;
    
```

```
END_IF;  
END_IF;
```

3.4. Рівень технологічних змінних

Рівень технологічних змінних (LVL1) реалізований як 7 функціональних блоків (рис.3.2):

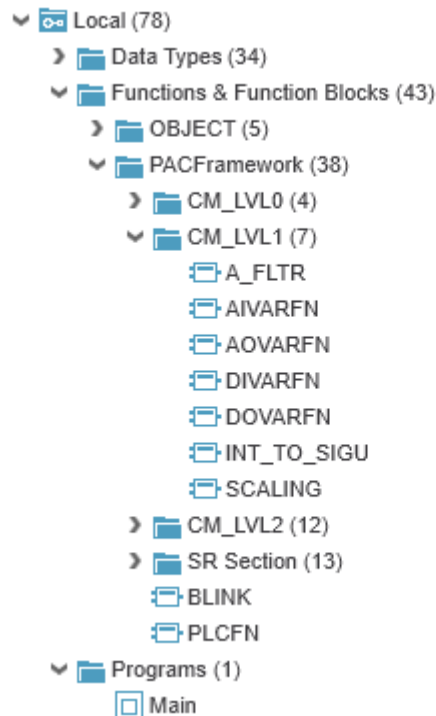


Рисунок 3.3 - Зовнішній вигляд структури функцій LVL1

AIVARFN

Код реалізації функціонального блоку AIVARFN:

```
IF PLCCFG.STA1.SCN1 THEN  
  IF AIVARCFG.CLSID = 0 THEN  
    AIVARCFG.CLSID := 16#1030;  
  END_IF;  
  AIVARCFG.PRM1.PRM_DSBL := FALSE;  
  AIVARCFG.PRM1.PRM_QALENBL := true;  
  AIVARCFG.PRM1.PRM_BRKENBL := TRUE;  
  AIVARCFG.PRM1.PRM_OVRLENBL := TRUE;  
  //AIVARCFG.PRM1.PRM_LOENBL := TRUE;  
  //AIVARCFG.PRM1.PRM_HIENBL := TRUE;  
  //AIVARCFG.PRM1.PRM_LOLOENBL := TRUE;  
  //AIVARCFG.PRM1.PRM_HIHIENBL := TRUE;  
  AIVARCFG.STA1.STA_SML := TRUE;  
  AIVARCFG.T_PREV := PLCCFG.TQMS;  
  IF AIVARCFG.CHID = 0 THEN  
    AIVARCFG.CHID := AIVARCFG.CHIDDF;  
  END_IF;  
  IF CHCFG.ID > 0 THEN  
    VRAW := CHCFG.VAL;  
  ELSE  
    VRAW := 0;
```

```

END_IF;
AIVARCFG.VALPRV := TO_REAL(TO_INT(VRAW));
AIVARCFG.VRAW := AIVARCFG.VALPRV;
AIVARCFG.VAL := AIVARCFG.VRAW;
AIVARCFG.T_STEP1 := 0;
AIVARCFG.STEP1 := 400;
RETURN;
END_IF;

BRK := AIVARCFG.STA1.STA_BRK;
OVRD := AIVARCFG.STA1.STA_OVRD;
BAD := AIVARCFG.STA1.STA_BAD;
ALDIS := AIVARCFG.STA1.STA_ALDIS;
ENBL := AIVARCFG.STA1.STA_ENBL;
ALM := AIVARCFG.STA1.STA_ALM;
LOLO := AIVARCFG.STA1.STA_LOLO;
LO := AIVARCFG.STA1.STA_LO;
HI := AIVARCFG.STA1.STA_HI;
HIHI := AIVARCFG.STA1.STA_HIHI;
WRN := AIVARCFG.STA1.STA_WRN;
FRC := AIVARCFG.STA1.STA_FRC;
SML := AIVARCFG.STA1.STA_SML;
INBUF := (AIVARCFG.ID = VARBUF1.ID) AND (AIVARCFG.CLSID = VARBUF1.CLSID);
CMDLOAD := AIVARHMI.STA1.STA_CMDLOAD;
DLNK := (CHCFG.ID > 0);
VARENBL := NOT AIVARCFG.PRM1.PRM_DSBL AND DLNK;
(*переводимо в реальні одиниці з урахуванням масштабування при необхідності*)
LORAW := AIVARCFG.LORAW;
HIRAW := AIVARCFG.HIRAW;
LOENG := AIVARCFG.LOENG;
HIENG := AIVARCFG.HIENG;
LOSP := AIVARCFG.LOSP;
HISP := AIVARCFG.HISP;
LOLOSP := AIVARCFG.LOLOSP;
HIHISP := AIVARCFG.HIHISP;
IF AIVARCFG.T_FLT < 0 THEN
  AIVARCFG.T_FLT := 1;
END_IF;
T_FLT := UINT TO UDINT(AIVARCFG.T_FLT);
T_DEALL := UINT TO UDINT(AIVARCFG.T_DEALL*100);
T_DEAL := UINT TO UDINT(AIVARCFG.T_DEAL*100);
T_DEAH := UINT TO UDINT(AIVARCFG.T_DEAH*100);
T_DEAHH := UINT TO UDINT(AIVARCFG.T_DEAHH*100);
(*ping-pong*)
IF DLNK THEN
  CHCFG.STA1.STA_PNG := true;
  CHCFG.VARID := AIVARCFG.ID;
END_IF;

dT1 := PLCCFG.TQMS - AIVARCFG.T_PREV;
CMD := 0;
VRAW := CHCFG.VAL;
T_STEPMS := AIVARCFG.T_STEP1;
VAL := AIVARCFG.VAL;
VALPRV := AIVARCFG.VALPRV;
IF VALPRV <= AIVARCFG.LOENG THEN
  VALPRV:=AIVARCFG.LOENG;
ELSIF VALPRV>AIVARCFG.HIENG THEN
  VALPRV:=AIVARCFG.HIENG;
END_IF;
(*перевірка NaN*)
IF NOT (VALPRV<AIVARCFG.LOENG OR VALPRV>=AIVARCFG.LOENG) THEN
  VALPRV:=AIVARCFG.LOENG;

```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 55 |

```

END_IF;
(*1% від шкали*)
VAL1PROC := (AIVARCFG.HIENG - AIVARCFG.LOENG) / 100.0; (* 1 % EU*)
IF VAL1PROC = 0.0 THEN
    VAL1PROC := 1.0;
END_IF;
(*гістерезис в залежності від параметра вказівки гістерезису*)
IF AIVARCFG.PRM1.PRM_PARAISPROC THEN (***)
    HYST := AIVARCFG.HYST * VAL1PROC;
ELSE (*real units*)
    HYST := AIVARCFG.HYST;
END_IF;
(*перевірка на коректність діапазонів*)
IF ABS(AIVARCFG.HIRAW - AIVARCFG.LORAW) < 1 THEN
    AIVARCFG.LORAW := 0;
    AIVARCFG.HIRAW := 10000;
END_IF;
IF ABS(AIVARCFG.HIENG - AIVARCFG.LOENG) < 0.00001 THEN
    AIVARCFG.LOENG := 0.0;
    AIVARCFG.HIENG := 100.0;
END_IF;
(*обнулення некоректних значень + infinity (INF) - infinity (-INF) SNAN QNAN *)
(*коли експонента (23-30 біти) = 255 *)
tmpDWORD := SHR(TO_BCD_DWORD(TO_UINT(AIVARCFG.VAL)), 23) AND TO_DWORD(16#00FF);
IF tmpDWORD = 16#00ff THEN
    AIVARCFG.VAL := 0.0;
END_IF;
(*масштабування*)
PRM_SCL:=0;
STA_SCL:=0;
PRM_SCL.%X0 := AIVARCFG.PRM1.PRM_SQRT; (*квадратнокоренева залежність, X0 *)
PRM_SCL.%X1 := true; (*обмежувати вихідну величину, X1*)

SCALING1(IN := TO_REAL(TO_INT(VRAW)),
    in_min := INT_TO_REAL(LORAW),
    in_max := INT_TO_REAL(HIRAW),
    out_min := LOENG,
    out_max := HIENG,
    STA1 := STA_SCL,
    PRM1 := PRM_SCL,
    SCALING => VALNOFLT);

(* якщо відсічка нуля активна (більше нуля) і значення менше то значення нуля,
інакше маштабуем значення*)
IF AIVARCFG.ZERO_CUT_VAL > 0.0 AND (AIVARCFG.LOENG +
TO_REAL(TO_INT(AIVARCFG.VRAW) - AIVARCFG.LORAW) * (AIVARCFG.HIENG - AIVARCFG.LOENG) /
INT_TO_REAL(AIVARCFG.HIRAW - AIVARCFG.LORAW)) <= AIVARCFG.ZERO_CUT_VAL THEN
    VALNOFLT := 0.0;
END_IF;
(*фільтрація*)
A_FLTR1(IN := VALNOFLT,
    dt1 := dt1,
    T_FLT := T_FLT,
    PRM1 := tmpuint,
    STA1 := tmpuint,
    VALPRV := VALPRV,
    A_FLTR => VAL);
(* ширококомвне дефорсування*)
IF PLCCFG.CMD = 16#4302 THEN
    FRC := false; (*дефорсувати об'єкт типу*)
END_IF;

IF CMDLOAD THEN

```

```

CMD := 16#0100;
ELSIF INBUF AND VARBUF1.CMD <> 0 THEN
  CMD := VARBUF1.CMD;
END_IF;
CASE CMD OF
  16#0001: (*записати максимум діапазону*)
    IF FRC AND INBUF THEN
      AIVARCFG.VALFRC := AIVARCFG.HIENG;
      VAL := AIVARCFG.HIENG;
      AIVARCFG.STEP1 := 400;
      AIVARCFG.T_STEP1 := 0;
    END_IF;
  16#0002: (*записати мінімум діапазону*)
    IF FRC AND INBUF THEN
      AIVARCFG.VALFRC := AIVARCFG.LOENG;
      VAL := AIVARCFG.LOENG;
      AIVARCFG.STEP1 := 400;
      AIVARCFG.T_STEP1 := 0;
    END_IF;
  16#0003: (*записати середину діапазону*)
    IF FRC AND INBUF THEN
      AIVARCFG.VALFRC := (AIVARCFG.HIENG - AIVARCFG.LOENG) / 2.0;
      VAL := (AIVARCFG.HIENG - AIVARCFG.LOENG) / 2.0;
      AIVARCFG.STEP1 := 400;
      AIVARCFG.T_STEP1 := 0;
    END_IF;
  16#0100: (*прочитати конфігурацію 256*)
    VARBUF1.MSG := 201;
    VARBUF1.PRM1.%X0 := AIVARCFG.PRM1.PRM_LOENBL;
    VARBUF1.PRM1.%X1 := AIVARCFG.PRM1.PRM_HIENBL;
    VARBUF1.PRM1.%X2 := AIVARCFG.PRM1.PRM_LOLOENBL;
    VARBUF1.PRM1.%X3 := AIVARCFG.PRM1.PRM_HIHIENBL;
    VARBUF1.PRM1.%X4 := AIVARCFG.PRM1.PRM_BRKENBL;
    VARBUF1.PRM1.%X5 := AIVARCFG.PRM1.PRM_OVRLENBL;
    VARBUF1.PRM1.%X6 := AIVARCFG.PRM1.PRM_QALENBL;
    VARBUF1.PRM1.%X7 := AIVARCFG.PRM1.PRM_DSBL;
    VARBUF1.PRM1.%X8 := AIVARCFG.PRM1.PRM_PWLENBL;
    VARBUF1.PRM1.%X9 := AIVARCFG.PRM1.PRM_TOTALON;
    VARBUF1.PRM1.%X10 := AIVARCFG.PRM1.PRM_SQRT;
    VARBUF1.PRM1.%X11 := AIVARCFG.PRM1.PRM_PARAISPROC;
    VARBUF1.PRM1.%X12 := AIVARCFG.PRM1.PRM_AFRZENBL;
    VARBUF1.PRM1.%X13 := AIVARCFG.PRM1.PRM_ASPDENBL;
    VARBUF1.PRM1.%X14 := AIVARCFG.PRM1.PRM_STATICMAP;
    VARBUF1.PRM1.%X15 := AIVARCFG.PRM1.PRM_NORAW;
    VARBUF1.ID := AIVARCFG.ID;
    VARBUF1.CLSID := AIVARCFG.CLSID;
    VARBUF1.CHID := AIVARCFG.CHID;
    VARBUF1.VALR := AIVARCFG.VALFRC;
    VARBUF1.LORAW := AIVARCFG.LORAW;
    VARBUF1.HIRAW := AIVARCFG.HIRAW;
    VARBUF1.LOENG := AIVARCFG.LOENG;
    VARBUF1.HIENG := AIVARCFG.HIENG;
    VARBUF1.HIHISP := AIVARCFG.HIHISP;
    VARBUF1.HISP := AIVARCFG.HISP;
    VARBUF1.LOSP := AIVARCFG.LOSP;
    VARBUF1.LOLOSP := AIVARCFG.LOLOSP;
    VARBUF1.T_FLTSP := AIVARCFG.T_FLT;
    VARBUF1.HYST := AIVARCFG.HYST;
    VARBUF1.T_DEAHH := AIVARCFG.T_DEAHH;
    VARBUF1.T_DEAH := AIVARCFG.T_DEAH;
    VARBUF1.T_DEAL := AIVARCFG.T_DEAL;
    VARBUF1.T_DEALL := AIVARCFG.T_DEALL;
    VARBUF1.VALPRV_AFRZ := AIVARCFG.VALPRV_AFRZ;

```

```

VARBUF1.VALPRV_ASPD :=AIVARCFG.VALPRV_ASPD;
VARBUF1.DEASP_AFRZ :=AIVARCFG.DEASP_AFRZ;
VARBUF1.DOPSP_ASPD := AIVARCFG.DOPSP_ASPD;
VARBUF1.ZERO_CUT_VAL := AIVARCFG.ZERO_CUT_VAL;

16#0101: (*записати конфігурацію 257*)
(* MSG 200-Ok 400-Error
// 200 - Дані записані
// 201 - Дані прочитані
// 403 - канал вже зайнятий
// 404 - номер каналу не відповідає діапазону *)

AIVARCFG.PRM1.PRM_LOENBL := VARBUF1.PRM1.%X0;
AIVARCFG.PRM1.PRM_HIENBL := VARBUF1.PRM1.%X1;
AIVARCFG.PRM1.PRM_LOLOENBL := VARBUF1.PRM1.%X2;
AIVARCFG.PRM1.PRM_HIHIENBL := VARBUF1.PRM1.%X3;
AIVARCFG.PRM1.PRM_BRKENBL := VARBUF1.PRM1.%X4;
AIVARCFG.PRM1.PRM_OVRLENBL := VARBUF1.PRM1.%X5;
AIVARCFG.PRM1.PRM_QALENBL := VARBUF1.PRM1.%X6;
AIVARCFG.PRM1.PRM_DSBL := VARBUF1.PRM1.%X7;
AIVARCFG.PRM1.PRM_PWLENBL := VARBUF1.PRM1.%X8;
AIVARCFG.PRM1.PRM_TOTALON := VARBUF1.PRM1.%X9;
AIVARCFG.PRM1.PRM_SQRT := VARBUF1.PRM1.%X10;
AIVARCFG.PRM1.PRM_PARAISPROC := VARBUF1.PRM1.%X11;
AIVARCFG.PRM1.PRM_AFRZENBL := VARBUF1.PRM1.%X12;
AIVARCFG.PRM1.PRM_ASPDENBL := VARBUF1.PRM1.%X13;
AIVARCFG.PRM1.PRM_STATICMAP := VARBUF1.PRM1.%X14;
AIVARCFG.PRM1.PRM_NORAW := VARBUF1.PRM1.%X15;
AIVARCFG.LORAW := VARBUF1.LORAW;
AIVARCFG.HIRAW := VARBUF1.HIRAW;
AIVARCFG.LOENG := VARBUF1.LOENG;
AIVARCFG.HIENG := VARBUF1.HIENG;
AIVARCFG.HIHISP := VARBUF1.HIHISP;
AIVARCFG.HISP := VARBUF1.HISP;
AIVARCFG.LOSP := VARBUF1.LOSP;
AIVARCFG.LOLOSP := VARBUF1.LOLOSP;
AIVARCFG.T_FLT := VARBUF1.T_FLTSP;
AIVARCFG.HYST := VARBUF1.HYST;
AIVARCFG.T_DEAHH := VARBUF1.T_DEAHH;
AIVARCFG.T_DEAH := VARBUF1.T_DEAH;
AIVARCFG.T_DEAL := VARBUF1.T_DEAL;
AIVARCFG.T_DEALL := VARBUF1.T_DEALL;
AIVARCFG.VALPRV_AFRZ := VARBUF1.VALPRV_AFRZ ;
AIVARCFG.VALPRV_ASPD := VARBUF1.VALPRV_ASPD ;
AIVARCFG.DEASP_AFRZ := VARBUF1.DEASP_AFRZ ;
AIVARCFG.DOPSP_ASPD := VARBUF1.DOPSP_ASPD ;
AIVARCFG.ZERO_CUT_VAL := VARBUF1.ZERO_CUT_VAL ;
(**)
VARBUF1.MSG:=200;
IF VARBUF1.CHID>=0 AND VARBUF1.CHID <= PLCCFG.AICNT THEN
  IF CHAI[VARBUF1.CHID].VARID = 0 THEN
    AIVARCFG.CHID := VARBUF1.CHID;
  ELSIF VARBUF1.CHID <> AIVARCFG.CHID THEN
    VARBUF1.MSG := 403; (* канал вже зайнятий*)
  END_IF;
ELSE
  VARBUF1.MSG := 404; (*номер каналу не відповідає діапазону*)
END_IF;
IF INBUF THEN
  VARBUF1.CHID := AIVARCFG.CHID;
END_IF;

16#0102: (*записати значення за замовченням*)

```

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Лист | № докум. | Підпис | Дата |

```

    AIVARCFG.CHID := AIVARCFG.CHIDDF;
16#0160: (*інвертувати LOENBL*)
    AIVARCFG.PRM1.PRM_LOENBL := NOT AIVARCFG.PRM1.PRM_LOENBL;
16#0161: (*інвертувати HIENBL*)
    AIVARCFG.PRM1.PRM_HIENBL := NOT AIVARCFG.PRM1.PRM_HIENBL;
16#0162: (*інвертувати LOLOENBL*)
    AIVARCFG.PRM1.PRM_LOLOENBL := NOT AIVARCFG.PRM1.PRM_LOLOENBL;
16#0163: (*інвертувати HIHIENBL*)
    AIVARCFG.PRM1.PRM_HIHIENBL := NOT AIVARCFG.PRM1.PRM_HIHIENBL;
16#0300: (*перемкнути форсування*)
    FRC := NOT FRC;
16#0301: (*форсувати один/усі об'єкти типу*)
    FRC := true;
16#0302: (*дефорсувати один/усі об'єкти типу*)
    FRC := false;
16#0311: (* імітувати*)
    SML := true;
16#0312: (* зняти режим імітації*)
    SML := false;
END_CASE;

IF FRC THEN
    IF INBUF THEN
        AIVARCFG.VAL := VARBUF1.VALR;
        AIVARCFG.VALFRC := VARBUF1.VALR;
    END_IF;
    VAL := AIVARCFG.VALFRC; (**)
ELSIF SML THEN (* режим симуляції - значення змінюється ззовні *)
    VAL := AIVARCFG.VAL;
ELSE
    AIVARCFG.VALFRC := VAL;
END_IF;

    (*обробка тривоги*)
BRK := false;
LO := false;
LOLO := false;
HI := false;
HIHI := false;
OVRLD := false;
T_DEAQALSP := 10;      (* час затримки на виникнення тривоги обривів і кз в 0.1 с*)

    (*не алармувати*)
IF VARENBL THEN (*тільки при активності змінної обробляються тривоги*)
    (*перевірка на спрацювання*)
    tmpOVRLD := CHCFG.VAL >= 11000;
    tmpBRK := TO_INT(CHCFG.VAL) <= -1000;
    tmpHI := (VAL >= AIVARCFG.HISP);
    tmpHIHI := (VAL >= AIVARCFG.HIHISP);
    tmpLO := (VAL <= AIVARCFG.LOSP);
    tmpLOLO := (VAL <= AIVARCFG.LOLOSP);
    (*якщо активована перевірка якості то PRM_BRKENBL та PRM_OVRLENBL ставимо
насильно*)
    IF NOT AIVARCFG.PRM1.PRM_QALENBL THEN
        AIVARCFG.PRM1.PRM_BRKENBL := false;
        AIVARCFG.PRM1.PRM_OVRLENBL := false;
    END_IF;

    CASE AIVARCFG.STEP1 OF
        100: (*BRK - обрив каналу*)
            LO := AIVARCFG.PRM1.PRM_LOENBL;

```

```

      LOLO := AIVARCFG.PRM1.PRM_LOLOENBL;
      IF NOT tmpBRK OR NOT AIVARCFG.PRM1.PRM_BRKENBL THEN (*вихід із зони*)
        AIVARCFG.STEP1 := 200;
        (*AIVARCFG.T_STEP1 := 0;*)
      ELSIF T_STEPMS >= INT_TO_UDINT(T_DEAQALSP) THEN (*тривога*)
        BRK := true;
      END_IF;
200: (*LOLO*)
      LO := AIVARCFG.PRM1.PRM_LOENBL;
      LOLO := AIVARCFG.PRM1.PRM_LOLOENBL AND T_STEPMS >=
UINT_TO_UDINT(AIVARCFG.T_DEALL);
      IF NOT LOLO AND NOT tmpLOLO
        OR NOT AIVARCFG.PRM1.PRM_LOLOENBL
        OR AIVARCFG.VAL > (AIVARCFG.LOLOSP + HYST) THEN (*вихід із зони*)
        AIVARCFG.STEP1 := 300;
        (*AIVARCFG.T_STEP1 := 0;*)
      END_IF;
      IF LOLO AND tmpBRK AND AIVARCFG.PRM1.PRM_BRKENBL THEN
        AIVARCFG.STEP1 := 100;
        AIVARCFG.T_STEP1 := 0;
      END_IF;
300: (*LO*)
      LO := AIVARCFG.PRM1.PRM_LOENBL AND T_STEPMS >=
UINT_TO_UDINT(AIVARCFG.T_DEAL);
      IF (NOT tmpLO AND NOT LO OR NOT AIVARCFG.PRM1.PRM_LOENBL) (*вихід
із зони*)
        OR AIVARCFG.VAL > (AIVARCFG.LOSP + HYST) THEN
        AIVARCFG.STEP1 := 400;
        AIVARCFG.T_STEP1 := 0;
      END_IF;
      IF LO AND tmpLOLO AND AIVARCFG.PRM1.PRM_LOLOENBL THEN
        AIVARCFG.STEP1 := 200;
        AIVARCFG.T_STEP1 := 0;
      END_IF;
      IF LO AND tmpBRK AND AIVARCFG.PRM1.PRM_BRKENBL THEN
        AIVARCFG.STEP1 := 100;
        AIVARCFG.T_STEP1 := 0;
      END_IF;
400: (*norma*)
      IF tmpHI AND AIVARCFG.PRM1.PRM_HIENBL THEN
        AIVARCFG.STEP1 := 500;
        AIVARCFG.T_STEP1 := 0;
      END_IF;
      IF tmpLO AND AIVARCFG.PRM1.PRM_LOENBL THEN
        AIVARCFG.STEP1 := 300;
        AIVARCFG.T_STEP1 := 0;
      END_IF;
      IF tmpHIHI AND AIVARCFG.PRM1.PRM_HIHIENBL THEN
        AIVARCFG.STEP1 := 460;
        AIVARCFG.T_STEP1 := 0;
      END_IF;
      IF tmpLOLO AND AIVARCFG.PRM1.PRM_LOLOENBL THEN
        AIVARCFG.STEP1 := 420;
        AIVARCFG.T_STEP1 := 0;
      END_IF;
      IF tmpBRK AND AIVARCFG.PRM1.PRM_BRKENBL THEN
        AIVARCFG.STEP1 := 410;
        AIVARCFG.T_STEP1 := 0;
      END_IF;
      IF tmpOVRD AND AIVARCFG.PRM1.PRM_OVRLENBL THEN
        AIVARCFG.STEP1 := 470;
        AIVARCFG.T_STEP1 := 0;
      END_IF;

```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 60 |

```

410: (*norma -> break*)
    IF NOT tmpBRK OR NOT AIVARCFG.PRM1.PRM_BRKENBL THEN (*вихід із зони*)
        AIVARCFG.STEP1 := 400;
        AIVARCFG.T_STEP1 := 0;
    ELSIF T_STEPMS >= INT_TO_UDINT(T_DEAQALSP) THEN (*тривога*)
        AIVARCFG.STEP1 := 100;
    END_IF;
420: (*norma -> LOLO*)
    IF NOT tmpLOLO OR NOT AIVARCFG.PRM1.PRM_LOLOENBL THEN (*вихід із
зони*)
        AIVARCFG.STEP1 := 400;
        AIVARCFG.T_STEP1 := 0;
    ELSIF T_STEPMS >= UINT_TO_UDINT(AIVARCFG.T_DEALL) THEN (*тривога*)
        AIVARCFG.STEP1 := 200;
    END_IF;
460: (*norma -> HIHI*)
    IF NOT tmpHIHI OR NOT AIVARCFG.PRM1.PRM_HIHIENBL THEN (*вихід із
зони*)
        AIVARCFG.STEP1 := 400;
        AIVARCFG.T_STEP1 := 0;
    ELSIF T_STEPMS >= UINT_TO_UDINT(AIVARCFG.T_DEAHH) THEN (*тривога*)
        AIVARCFG.STEP1 := 600;
    END_IF;
470: (*norma -> OVRLD*)
    IF NOT tmpOVRLD OR NOT AIVARCFG.PRM1.PRM_OVRLENBL THEN (*вихід із
зони*)
        AIVARCFG.STEP1 := 400;
        AIVARCFG.T_STEP1 := 0;
    ELSIF T_STEPMS >= INT_TO_UDINT(T_DEAQALSP) THEN (*тривога*)
        AIVARCFG.STEP1 := 700;
    END_IF;
500: (*HI*)
    HI := AIVARCFG.PRM1.PRM_HIENBL AND T_STEPMS >=
UINT_TO_UDINT(AIVARCFG.T_DEAH);
    IF NOT tmpHI AND NOT HI
        OR NOT AIVARCFG.PRM1.PRM_HIENBL (*вихід із зони*)
        OR AIVARCFG.VAL < (AIVARCFG.HISP - HYST) THEN
        AIVARCFG.STEP1 := 400;
        AIVARCFG.T_STEP1 := 0;
    END_IF;
    IF HI AND tmpHIHI AND AIVARCFG.PRM1.PRM_HIHIENBL THEN
        AIVARCFG.STEP1 := 600;
        AIVARCFG.T_STEP1 := 0;
    END_IF;
    IF HI AND tmpOVRLD AND AIVARCFG.PRM1.PRM_OVRLENBL THEN
        AIVARCFG.STEP1 := 700;
        AIVARCFG.T_STEP1 := 0;
    END_IF;
600: (*HIHI*)
    HI := AIVARCFG.PRM1.PRM_HIENBL;
    HIHI := AIVARCFG.PRM1.PRM_HIHIENBL AND T_STEPMS >=
UINT_TO_UDINT(AIVARCFG.T_DEAHH);
    IF NOT tmpHIHI AND NOT HIHI OR
        NOT AIVARCFG.PRM1.PRM_HIHIENBL OR
        AIVARCFG.VAL < (AIVARCFG.HIHISP - HYST) THEN (*вихід із зони*)
        AIVARCFG.STEP1 := 500;
        (*AIVARCFG.T_STEP1 := 0;*)
    END_IF;
    IF HI AND tmpOVRLD AND AIVARCFG.PRM1.PRM_OVRLENBL THEN
        AIVARCFG.STEP1 := 700;
        AIVARCFG.T_STEP1 := 0;
    END_IF;
700: (*OVRLD - перегрузка/КЗ*)

```

```

HI := AIVARCFG.PRM1.PRM_HIENBL;
HIHI := AIVARCFG.PRM1.PRM_HIHIENBL;
IF (NOT tmpOVRLD OR NOT AIVARCFG.PRM1.PRM_OVRLENBL) THEN (*вихід із
зони*)
    AIVARCFG.STEP1 := 600;
    (*AIVARCFG.T_STEP1 := 0;*)
    ELSIF T_STEPMS >= INT_TO_UDINT(T_DEAQALSP) THEN (*тривога*)
        OVRLD := true;
    END_IF;
ELSE
    AIVARCFG.STEP1 := 400;
END_CASE;
ELSE
    AIVARCFG.VRAW := TO_REAL(TO_INT(VRAW));
    AIVARCFG.VAL := VALNOFLT;
    AIVARCFG.T_STEP1 := 0;
    AIVARCFG.STEP1 := 400;
    BAD := false;
    BRK := false;
    LO := false;
    LOLO := false;
    HI := false;
    HIHI := false;
    OVRLD := false;
    ENBL := false;
END_IF;

(*----- не є частиною каркасу, кастом для Енікон*)
IF PLCCFG.STA1.SCN1 THEN
    AIVARCFG.VALPRV_AFRZ := VAL;
    AIVARCFG.VALPRV_ASPD := VAL;
END_IF;
(* Перевірка на зависання *)
(*протягом 20 сек значення повинно змінитися більш ніж на задану уставку у
од.вимір якщо уставка 0 - по дефолту*)
IF AIVARCFG.PRM1.PRM_AFRZENBL THEN
    tmpAFRZ:=AIVARCFG.STA2.STA2_AFRZ;
    (*перевіряти на кожній 20 сек на один цикл*)
    IF (PLCCFG.TQ MOD 20) = 0 AND PLCCFG.PLS1.P1S THEN
        IF ABS(AIVARCFG.VALPRV_AFRZ - VAL) < AIVARCFG.DEASP_AFRZ THEN
            tmpAFRZ:=true;
        ELSE
            tmpAFRZ:=false; (*самоскидання*)
        END_IF;
        AIVARCFG.VALPRV_AFRZ := VAL;
    END_IF;
    (*якщо значення зміниться у будь який час оновити його (відкидання помилки
випадкової рівності через вказаний час) *)
    IF ABS(VAL - TO_INT(AIVARCFG.VALPRV)) > AIVARCFG.DEASP_AFRZ THEN
        AIVARCFG.VALPRV_AFRZ := VAL;
    END_IF;
    AIVARCFG.STA2.STA2_AFRZ := tmpAFRZ;
END_IF;
(* Перевірка на скачок *)
IF AIVARCFG.PRM1.PRM_ASPDENBL THEN
    (* аларм швидкості зміни, якщо через заданий час значення скочило більше ніж
допуск - помилка WRN*)
    tmpASPD:=false;
    AIVARCFG.STA2.STA2_ASPD := tmpASPD;
    (*перевіряти на кожній 5 сек на один цикл*)
    IF (PLCCFG.TQ MOD 5) = 0 AND PLCCFG.PLS1.P1S THEN
        IF ABS(AIVARCFG.VALPRV_ASPD - VAL) > AIVARCFG.DOPSP_ASPD THEN
            tmpASPD:=true;

```

```

ELSE
    tmpASPD:=false; (*самоскидання*)
END_IF;
AIVARCFG.VALPRV_ASPD := VAL;
END_IF;

AIVARCFG.STA2.STA2_AOVRFL:=VRAW>10100 AND VRAW<10999;
AIVARCFG.STA2.STA2_AUNDRFL := TO_INT(VRAW) <- 100 AND TO_INT(VRAW) >- 999;
tmpAOVRFL:=AIVARCFG.STA2.STA2_AOVRFL;
tmpAUNDRFL:=AIVARCFG.STA2.STA2_AUNDRFL;
END_IF;
(*-----*)
BAD := (CHCFG.STA1.STA_BAD OR BRK OR OVRLD OR tmpAOVRFL OR tmpAUNDRFL OR tmpAFRZ)
AND AIVARCFG.PRM1.PRM_QALENBL AND VARENBL AND NOT SML AND NOT FRC;
ALM := (LOLO OR HIHI) AND NOT BAD;
WRN := (LO OR HI OR tmpASPD) AND NOT ALM AND NOT BAD;

IF BAD THEN
    PLCCFG.ALM.BAD := true;
    PLCCFG.CNTBAD := PLCCFG.CNTBAD + 1;
    IF NOT AIVARCFG.STA1.STA_BAD THEN
        PLCCFG.ALM.NWBAD := true;
    END_IF;
END_IF;
IF ALM THEN
    PLCCFG.ALM.ALM := true;
    PLCCFG.CNTALM := PLCCFG.CNTALM + 1;
    IF NOT AIVARCFG.STA1.STA_ALM THEN
        PLCCFG.ALM.NWALM := true;
    END_IF;
END_IF;
IF WRN THEN
    PLCCFG.ALM.WRN := true;
    PLCCFG.CNTWRN := PLCCFG.CNTWRN + 1;
    IF NOT AIVARCFG.STA1.STA_WRN THEN
        PLCCFG.ALM.NWWRN := true;
    END_IF;
END_IF;
IF FRC THEN
    PLCCFG.STA1.FRC1 := true;
    PLCCFG.CNTFRC := PLCCFG.CNTFRC + 1;
END_IF;
IF SML THEN
    PLCCFG.STA1.SML := true;
END_IF;

AIVARCFG.STA1.STA_BRK := BRK;
AIVARCFG.STA1.STA_OVRLD := OVRLD;
AIVARCFG.STA1.STA_BAD := BAD;
AIVARCFG.STA1.STA_ALDIS := ALDIS;
AIVARCFG.STA1.STA_DLNK := DLNK;
AIVARCFG.STA1.STA_ENBL := VARENBL;
AIVARCFG.STA1.STA_ALM := ALM;
AIVARCFG.STA1.STA_LOLO := LOLO;
AIVARCFG.STA1.STA_LO := LO;
AIVARCFG.STA1.STA_HI := HI;
AIVARCFG.STA1.STA_HIHI := HIHI;
AIVARCFG.STA1.STA_WRN := WRN;
AIVARCFG.STA1.STA_INBUF := INBUF;
AIVARCFG.STA1.STA_FRC := FRC;
AIVARCFG.STA1.STA_SML := SML;
AIVARCFG.STA1.STA_CMDLOAD := FALSE;
VALPROC := (VAL - AIVARCFG.LOENG) / VAL1PROC;

```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 63 |

```

IF VALPROC < 0.0 THEN
    VALPROC := 0.0;
END_IF;
IF VALPROC > 100.0 THEN
    VALPROC := 100.0;
END_IF;

AIVARCFG.VAL := VAL;
AIVARCFG.VRAW := TO_REAL(TO_INT(VRAW));
AIVARCFG.VALPRV := VALPRV;
(* може бути частиною каркасу при необхідності статусних бітів і ларамів >15*)
AIVARCFG.VALPRCSTA2 := TO_WORD(TO_INT(VALPROC)*256) AND TO_WORD(16#FF00);
AIVARCFG.VALPRCSTA2.%X0 := AIVARCFG.STA2.STA2_ASPD;
AIVARCFG.VALPRCSTA2.%X1 := AIVARCFG.STA2.STA2_AFRZ;
AIVARCFG.VALPRCSTA2.%X2 := AIVARCFG.STA2.STA2_AOVRFL;
AIVARCFG.VALPRCSTA2.%X3 := AIVARCFG.STA2.STA2_AUNDRFL;
AIVARCFG.T_PREV := PLCCFG.TQMS;
AIVARCFG.T_STEP1 := AIVARCFG.T_STEP1 + dt1;
AIVARHMI.STA1 := AIVARCFG.STA1;
AIVARHMI.VAL := VAL;
AIVARHMI.VALPRCSTA2 := AIVARCFG.VALPRCSTA2;

IF AIVARCFG.T_STEP1 > 16#7FFF_FFFF THEN
    AIVARCFG.T_STEP1 := 16#7FFF_FFFF;
END_IF;

IF INBUF THEN
    VARBUF1.CMD := 0;
    VARBUF1.VALR := AIVARCFG.VAL;
    VARBUF1.STA1 := AIVARCFG.STA1;
    VARBUF1.VRAWR := TO_REAL(TO_INT(VRAW));
    IF NOT FRC THEN
        VARBUF1.VALR := VAL;
    END_IF;
    VARBUF1.STEP1 := AIVARCFG.STEP1;
    VARBUF1.T_STEP1 := AIVARCFG.T_STEP1;
    VARBUF1.VALPROC := AIVARCFG.VALPRCSTA2;
    VARBUF1.LOLOSP_PRC := REAL_TO_INT(AIVARCFG.LOLOSP / VAL1PROC * 100.0);
    VARBUF1.LOSP_PRC := REAL_TO_INT(AIVARCFG.LOSP / VAL1PROC * 100.0);
    VARBUF1.HISP_PRC := REAL_TO_INT(AIVARCFG.HISP / VAL1PROC * 100.0);
    VARBUF1.HIHISP_PRC := REAL_TO_INT(AIVARCFG.HIHISP / VAL1PROC * 100.0);
    VARBUF1.CH_CLSID := CHCFG.CLSID;
    VARBUF1.CH_STA.%X0 := CHCFG.STA1.STA_VRAW;
    VARBUF1.CH_STA.%X1 := CHCFG.STA1.STA_VALB;
    VARBUF1.CH_STA.%X2 := CHCFG.STA1.STA_BAD;
    VARBUF1.CH_STA.%X3 := CHCFG.STA1.STA_b3;
    VARBUF1.CH_STA.%X4 := CHCFG.STA1.STA_PNG;
    VARBUF1.CH_STA.%X5 := CHCFG.STA1.STA_ULNK;
    VARBUF1.CH_STA.%X6 := CHCFG.STA1.STA_MERR;
    VARBUF1.CH_STA.%X7 := CHCFG.STA1.STA_BRK;
    VARBUF1.CH_STA.%X8 := CHCFG.STA1.STA_SHRT;
    VARBUF1.CH_STA.%X9 := CHCFG.STA1.STA_NBD;
    VARBUF1.CH_STA.%X10 := CHCFG.STA1.STA_b10;
    VARBUF1.CH_STA.%X11 := CHCFG.STA1.STA_INIOTBUF;
    VARBUF1.CH_STA.%X12 := CHCFG.STA1.STA_INBUF;
    VARBUF1.CH_STA.%X13 := CHCFG.STA1.STA_FRC;
    VARBUF1.CH_STA.%X14 := CHCFG.STA1.STA_SML;
    VARBUF1.CH_STA.%X15 := CHCFG.STA1.STA_CMDLOAD;

    INT_TO_SIGU1(CLSID := CHCFG.CLSID,
                VALINT := VRAW,
                INT_TO_SIGU => VARBUF1.CH_VALSIG);

```

END_IF;

AOVARFN

Код реалізації функціонального блоку AOVARFN:

```
IF PLCCFG.STA1.SCN1 THEN
  AOVARCFG.CLSID := 16#1040;
  AOVARCFG.PRM1.PRM_DSBL := FALSE;
  AOVARCFG.PRM1.PRM_QALENBL := true;
  AOVARCFG.T_PREV := PLCCFG.TQMS;
  IF AOVARCFG.CHID = 0 THEN
    AOVARCFG.CHID := AOVARCFG.CHIDDF;
  END_IF;
  IF CHCFG.ID > 0 THEN
    CHCFG.VAL := AOVARCFG.VRAW;
  ELSE
    CHCFG.VAL := 0;
  END_IF;
  AOVARCFG.T_STEP1 := 0;
  AOVARCFG.STEP1 := 100;
  RETURN;
END_IF;

BAD := AOVARCFG.STA1.STA_BAD;
ALDIS := AOVARCFG.STA1.STA_ALDIS;
DLNK := AOVARCFG.STA1.STA_DLNK;
ENBL := AOVARCFG.STA1.STA_ENBL;
INBUF := AOVARCFG.STA1.STA_INBUF;
FRC := AOVARCFG.STA1.STA_FRC;
SML := AOVARCFG.STA1.STA_SML;
CMDLOAD := AOVARCFG.STA1.STA_CMDLOAD;
PRM_QALENBL := AOVARCFG.PRM1.PRM_QALENBL;
PRM_DSBL := AOVARCFG.PRM1.PRM_DSBL;
PRM_PWLENBL := AOVARCFG.PRM1.PRM_PWLENBL;
LORAW := AOVARCFG.LORAW;
HIRAW := AOVARCFG.HIRAW;
LOENG := AOVARCFG.LOENG;
HIENG := AOVARCFG.HIENG;
MVAL := AOVARCFG.MVAL;
IF AOVARCFG.T_FLT <= 0 THEN
  AOVARCFG.T_FLT := 1;
END_IF;
T_FLT := AOVARCFG.T_FLT;

INBUF := (AOVARCFG.ID = VARBUF1.ID) AND (AOVARCFG.CLSID = VARBUF1.CLSID);
CMDLOAD := AOVARHMI.STA1.STA_CMDLOAD;
CMD := 0;
DLNK := (CHCFG.ID > 0);
VARENBL := NOT PRM_DSBL AND DLNK;
T_STEPMS := AOVARCFG.T_STEP1;

VAL := AOVARCFG.VAL;
VRAW := AOVARCFG.VRAW;

(*ping-pong*)
IF DLNK THEN
  CHCFG.STA1.STA_PNG := true;
  CHCFG.VARID := AOVARCFG.ID;
END_IF;

IF NOT VARENBL THEN
  (* DOVARCFG.STA.VRAW := VRAW; *)
```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 65 |

```

AOVARCFG.T_STEP1 := 0;
AOVARCFG.STEP1 := 400;
END_IF;

dT1 := PLCCFG.TQMS - AOVARCFG.T_PREV;

(* ширококомвне дефорсування*)
IF PLCCFG.CMD = 16#4302 THEN
    FRC := false; (*дефорсувати об'єкт типу*)
END_IF;

IF CMDLOAD THEN
    CMD := 16#0100;
ELSIF INBUF AND VARBUF1.CMD <> 0 THEN
    CMD := VARBUF1.CMD;
END_IF;

(*commands*)
CASE CMD OF
    16#0001: (*записати максимум діапазону*)
        IF FRC AND INBUF THEN
            AOVARCFG.VALFRC := AOVARCFG.HIENG;
            VAL := AOVARCFG.HIENG;
            AOVARCFG.STEP1 := 100;
            AOVARCFG.T_STEP1 := 0;
        END_IF;
    16#0002: (*записати мінімум діапазону*)
        IF FRC AND INBUF THEN
            AOVARCFG.VALFRC := AOVARCFG.LOENG;
            VAL := AOVARCFG.LOENG;
            AOVARCFG.STEP1 := 100;
            AOVARCFG.T_STEP1 := 0;
        END_IF;
    16#0003: (*записати середину діапазону*)
        IF FRC AND INBUF THEN
            AOVARCFG.VALFRC := (AOVARCFG.HIENG - AOVARCFG.LOENG) / 2.0;
            VAL := (AOVARCFG.HIENG - AOVARCFG.LOENG) / 2.0;
            AOVARCFG.STEP1 := 100;
            AOVARCFG.T_STEP1 := 0;
        END_IF;
    16#0100: (*прочитати конфігурацію*)
        VARBUF1.MSG := 201;
        VARBUF1.PRM1.%X6 := PRM_QALENBL;
        VARBUF1.PRM1.%X7 := PRM_DSBL;
        VARBUF1.PRM1.%X8 := PRM_PWLENBL;
        VARBUF1.ID := AOVARCFG.ID;
        VARBUF1.CLSID := AOVARCFG.CLSID;
        VARBUF1.CHID := AOVARCFG.CHID;
        VARBUF1.VALR := AOVARCFG.VALFRC;
        VARBUF1.LORAW := LORAW;
        VARBUF1.HIRAW := HIRAW;
        VARBUF1.LOENG := LOENG;
        VARBUF1.HIENG := HIENG;
        VARBUF1.MVAL := MVAL;
        VARBUF1.T_FLTSP := T_FLTSP;

    16#0101: (*записати конфігурацію*)
        (* MSG 200-Ok 400-Error
        // 200 - Дані записані
        // 201 - Дані прочитані
        // 403 - канал вже зайнятий
        // 404 - номер каналу не відповідає діапазону *)
        PRM_QALENBL := VARBUF1.PRM1.%X6;

```

```

PRM_DSBL := VARBUF1.PRM1.%X7;
PRM_PWLENBL := VARBUF1.PRM1.%X8;
LORAW := VARBUF1.LORAW;
HIRAW := VARBUF1.HIRAW;
LOENG := VARBUF1.LOENG;
HIENG := VARBUF1.HIENG;
MVAL := VARBUF1.MVAL;
T_FLT := VARBUF1.T_FLTSP;
  (*перевіряти незайнятість каналу треба в функції контролю буферу *)
  VARBUF1.MSG:=200;
  IF VARBUF1.CHID>=0 AND VARBUF1.CHID <= PLCCFG.AOCNT THEN
    IF CHAO[VARBUF1.CHID].VARID = 0 THEN
      AOVARCFG.CHID := VARBUF1.CHID;
    ELSIF VARBUF1.CHID <> AOVARCFG.CHID THEN
      VARBUF1.MSG := 403; (* канал вже зайнятий*)
    END_IF;
  ELSE
    VARBUF1.MSG := 404; (*номер каналу не відповідає діапазону*)
  END_IF;

IF INBUF THEN
  VARBUF1.CHID := AOVARCFG.CHID;
END_IF;

16#0102: (*записати значення за замовченням*)
  AOVARCFG.CHID := AOVARCFG.CHIDDF;
;
16#0300: (*перемкнути форсування*)
  FRC := NOT FRC;
16#0301: (*форсувати один/усі об'єкти типу*)
  FRC := true;
16#0302: (*дефорсувати один/усі об'єкти типу*)
  FRC := false;
16#0311: (* імітувати*)
  SML := true;
16#0312: (* зняти режим імітації*)
  SML := false;
END_CASE;

IF ABS (HIRAW - LORAW) < 1 THEN
  LORAW := 0;
  HIRAW := 10000;
END_IF;
IF ABS(HIENG - LOENG) < 0.00001 THEN
  LOENG := 0.0;
  HIENG := 100.0;
END_IF;
VAL1PROC := INT_TO_REAL(HIRAW - LORAW) / 100.0; (* значення у % від шкали вимірювання *)

(*val*)
IF FRC THEN (*режим форсування*)
  IF INBUF THEN
    AOVARCFG.VALFRC := VARBUF1.VALR;
  END_IF;
  VAL := AOVARCFG.VALFRC;
ELSE
  AOVARCFG.VALFRC := VAL;
END_IF;

IF VARENBL THEN
  IF AOVARCFG.VALPRV<LOENG THEN AOVARCFG.VALPRV:=LOENG; END_IF;
  IF AOVARCFG.VALPRV>HIENG THEN AOVARCFG.VALPRV:=HIENG; END_IF;

```

```

(*фільтрація*)
A_FLTR1(IN := VAL,
        dt1 := dt1,
        T_FLT := UINT_TO_UDINT(T_FLT),
        PRM1 := tmpuint,
        STA1 := tmpuint,
        VALPRV := AOVARCFG.VALPRV,
        A_FLTR => VALFLT);
(*масштабування*)
PRM_SCL:=0;STA_SCL:=0;
PRM_SCL.%X0 := false;(*//квадратнокоренева залежність, X0 *)
PRM_SCL.%X1 := FALSE;(*//обмежувати вихідну величину, X1 *)
SCALING1(IN := VALFLT,
         in_min := LOENG,
         in_max := HIENG,
         out_min := INT_TO_REAL(LORAW),
         out_max := INT_TO_REAL(HIRAW),
         STA1 := STA_SCL,
         PRM1 := PRM_SCL);
VRAW:=TO_WORD(TO_INT(SCALING1.SCALING));

IF NOT SML THEN
    CHCFG.VAL := VRAW;
END_IF;(* режим симуляції - вихідне значення не змінюється *)
END_IF;
(*недостовірність каналу*)
tempBAD := CHCFG.STA1.STA_BAD AND PRM_QALENBL AND VARENBL AND NOT SML;
TDEAQUALSP := 10;      (* час затримки на виникнення тривог бед в 0.1 с*)

CASE AOVARCFG.STEP1 OF
    0:(*init*)
        AOVARCFG.STEP1 := 100;
        AOVARCFG.T_STEP1 := 0;
    100:(*norma*)
        BAD := false;
        IF tempBAD THEN
            AOVARCFG.STEP1 := 150;
            AOVARCFG.T_STEP1 := 0;
        END_IF;
    150:(*norma to BAD*)
        IF AOVARCFG.T_STEP1 > INT_TO_UDINT(TDEAQUALSP) THEN
            AOVARCFG.STEP1 := 200;
            AOVARCFG.T_STEP1 := 0;
        ELSIF NOT tempBAD THEN
            AOVARCFG.STEP1 := 100;
            AOVARCFG.T_STEP1 := 0;
        END_IF;
    200:(*BAD*)
        BAD := true;
        IF NOT tempBAD AND AOVARCFG.T_STEP1 > INT_TO_UDINT(TDEAQUALSP) THEN
            AOVARCFG.STEP1 := 100;
            AOVARCFG.T_STEP1 := 0;
        END_IF;
ELSE
    AOVARCFG.STEP1 := 0;
END_CASE;

IF BAD THEN
    PLCCFG.ALM.BAD := true;
    PLCCFG.CNTBAD := PLCCFG.CNTBAD + 1;
    IF NOT AOVARCFG.STA1.STA_BAD THEN
        PLCCFG.ALM.NWBAD := true;
    END_IF;

```

```

END_IF;
IF FRC THEN
    PLCCFG.STA1.FRC1 := true;
    PLCCFG.CNTFRC := PLCCFG.CNTFRC + 1;
END_IF;
IF SML THEN
    PLCCFG.STA1.SML := true;
END_IF;

CMDLOAD := FALSE;
AOVARCFG.STA1.STA_BAD := BAD;
AOVARCFG.STA1.STA_ALDIS := ALDIS;
AOVARCFG.STA1.STA_DLNK := DLNK;
AOVARCFG.STA1.STA_ENBL := VARENBL;
AOVARCFG.STA1.STA_INBUF := INBUF;
AOVARCFG.STA1.STA_FRC := FRC;
AOVARCFG.STA1.STA_SML := SML;
AOVARCFG.STA1.STA_CMDLOAD := FALSE;
AOVARCFG.PRM1.PRM_QALENBL := PRM_QALENBL;
AOVARCFG.PRM1.PRM_DSBL := PRM_DSBL;
AOVARCFG.PRM1.PRM_PWLENBL := PRM_PWLENBL;
AOVARHMI.STA1 := AOVARCFG.STA1;

IF VAL1PROC = 0.0 THEN
    VAL1PROC := 1.0;
END_IF;
VALPROC := INT_TO_REAL(TO_INT(CHCFG.VAL) - LORAW) / VAL1PROC;
IF VALPROC < 0.0 THEN
    VALPROC := 0.0;
END_IF;
IF VALPROC > 100.0 THEN
    VALPROC := 100.0;
END_IF;

AOVARCFG.VAL := VAL;
AOVARHMI.VAL := VAL;
AOVARCFG.VRAW := VRAW;
AOVARCFG.VALPROC := TO_WORD(TO_UINT(VALPROC*256.0)) AND TO_WORD(16#FF00); (*по
аналогії з AIVAR*)
AOVARHMI.VALPROC := AOVARCFG.VALPROC;
AOVARCFG.LORAW := LORAW;
AOVARCFG.HIRAW := HIRAW;
AOVARCFG.LOENG := LOENG;
AOVARCFG.HIENG := HIENG;
AOVARCFG.MVAL := MVAL;
AOVARCFG.T_FLT := T_FLT;
AOVARCFG.T_PREV := PLCCFG.TQMS;
AOVARCFG.T_STEP1 := AOVARCFG.T_STEP1 + dt1;
IF AOVARCFG.T_STEP1 > 16#7FFF_FFFF THEN
    AOVARCFG.T_STEP1 := 16#7FFF_FFFF;
END_IF;

IF INBUF THEN
    VARBUF1.CMD := 0;
    VARBUF1.STA_AO := AOVARHMI.STA1;
    VARBUF1.VRAWR := TO_REAL(TO_INT(VRAW));
    VARBUF1.VALR := VAL;
    VARBUF1.VALPROC := AOVARCFG.VALPROC;
    VARBUF1.STEP1 := AOVARCFG.STEP1;
    VARBUF1.T_STEP1 := AOVARCFG.T_STEP1;
    VARBUF1.CH_CLSID := CHCFG.CLSID;
    VARBUF1.CH_STA.%X0 := CHCFG.STA1.STA_VRAW;
    VARBUF1.CH_STA.%X1 := CHCFG.STA1.STA_VALB;

```

```

VARBUF1.CH_STA.%X2 := CHCFG.STA1.STA_BAD;
VARBUF1.CH_STA.%X3 := CHCFG.STA1.STA_b3;
VARBUF1.CH_STA.%X4 := CHCFG.STA1.STA_PNG;
VARBUF1.CH_STA.%X5 := CHCFG.STA1.STA_ULNK;
VARBUF1.CH_STA.%X6 := CHCFG.STA1.STA_MERR;
VARBUF1.CH_STA.%X7 := CHCFG.STA1.STA_BRK;
VARBUF1.CH_STA.%X8 := CHCFG.STA1.STA_SHRT;
VARBUF1.CH_STA.%X9 := CHCFG.STA1.STA_NBD;
VARBUF1.CH_STA.%X10 := CHCFG.STA1.STA_b10;
VARBUF1.CH_STA.%X11 := CHCFG.STA1.STA_INIOTBUF;
VARBUF1.CH_STA.%X12 := CHCFG.STA1.STA_INBUF;
VARBUF1.CH_STA.%X13 := CHCFG.STA1.STA_FRC;
VARBUF1.CH_STA.%X14 := CHCFG.STA1.STA_SML;
VARBUF1.CH_STA.%X15 := CHCFG.STA1.STA_CMDLOAD;

INT_TO_SIGU1 (CLSID := CHCFG.CLSID,
              VALINT := VRAW,
              INT_TO_SIGU => VARBUF1.CH_VALSIG);

END_IF;

```

DIVARFN

Код реалізації функціонального блоку DIVARFN:

```

(*DIVARCFG.CLSID
16#1010 - класичний
16#1011 - з лічильником в полі VALI, не змінюється в цій функції в нормальному
режимі*)
IF PLCCFG.STA1.SCN1 THEN
  DIVARCFG.CLSID := 16#1010;
  DIVARCFG.PRM1.PRM_DSBL := FALSE;
  DIVARCFG.T_PREV := PLCCFG.TQMS;
  DIVARCFG.STA1.STA_SML := TRUE;
  IF DIVARCFG.CHID = 0 THEN
    DIVARCFG.CHID := DIVARCFG.CHIDDF;
  END_IF;
  IF CHCFG.ID > 0 THEN
    VRAW := CHCFG.STA1.STA_VALB;
  ELSE
    VRAW := 0;
  END_IF;
  DIVARCFG.STA1.STA_VALPRV := VRAW;
  DIVARCFG.STA1.STA_VRAW := DIVARCFG.STA1.STA_VALPRV;
  DIVARCFG.STA1.STA_VALB := DIVARCFG.STA1.STA_VRAW;
  DIVARCFG.T_STEP1 := 0;
  DIVARCFG.STEP1 := 400;
  RETURN;
END_IF;

VRAW := DIVARCFG.STA1.STA_VRAW;
VAL := DIVARCFG.STA1.STA_VALB;
BAD := DIVARCFG.STA1.STA_BAD;
ALDIS := DIVARCFG.STA1.STA_ALDIS;
DLNK := DIVARCFG.STA1.STA_DLNK;
ENBL := DIVARCFG.STA1.STA_ENBL;
ALM := DIVARCFG.STA1.STA_ALM;
VALPRV := DIVARCFG.STA1.STA_VALPRV;
WRN := DIVARCFG.STA1.STA_WRN;
INBUF := DIVARCFG.STA1.STA_INBUF;
FRC := DIVARCFG.STA1.STA_FRC;
SML := DIVARCFG.STA1.STA_SML;
CMDLOAD := DIVARCFG.STA1.STA_CMDLOAD;

```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| | | | | | | 70 |
| Зм. | Лист | № докум. | Підпис | Дата | | |

```

PRM_ISALM := DIVARCFG.PRM1.PRM_ISALM;
PRM_ISWRN := DIVARCFG.PRM1.PRM_ISWRN;
PRM_INVERSE := DIVARCFG.PRM1.PRM_INVERSE;
PRM_NRMVAL := DIVARCFG.PRM1.PRM_NRMVAL;
PRM_QALENBL := DIVARCFG.PRM1.PRM_QALENBL;
PRM_DSBL := DIVARCFG.PRM1.PRM_DSBL;
INBUF := (DIVARCFG.ID = VARBUF1.ID) AND (DIVARCFG.CLSID = VARBUF1.CLSID);
CMDLOAD := DIVARHMI.STA1.STA_CMDLOAD;
CMD := 0;
DLNK := (CHCFG.ID > 0);
VARENBL := NOT PRM_DSBL AND DLNK;
VRAW := CHCFG.STA1.STA_VALB;
T_STEPMS := DIVARCFG.T_STEP1;

(*ping-pong*)
IF DLNK THEN
    CHCFG.STA1.STA_PNG := true;
    CHCFG.VARID := DIVARCFG.ID;
END_IF;

IF NOT VARENBL THEN
    DIVARCFG.STA1.STA_VRAW := VRAW;
    DIVARCFG.T_STEP1 := 0;
    DIVARCFG.STEP1 := 400;
END_IF;

dT1 := PLCCFG.TQMS - DIVARCFG.T_PREV;

(* ширококомовне дефорсування*)
IF PLCCFG.CMD = 16#4302 THEN
    FRC := false; (*дефорсувати об'єкт типу*)
END_IF;

IF CMDLOAD THEN
    CMD := 16#0100;
ELSIF INBUF AND VARBUF1.CMD <> 0 THEN
    CMD := VARBUF1.CMD;
END_IF;

(*commands*)
CASE CMD OF
    16#0001: (*записати 1 - тільки при форсуванні*)
        IF FRC AND INBUF THEN
            DIVARCFG.VALI := 1;
            VAL := true;
            DIVARCFG.STEP1 := 401;
            DIVARCFG.T_STEP1 := 0;
        END_IF;
    16#0002: (*записати 0 - тільки при форсуванні*)
        IF FRC AND INBUF THEN
            DIVARCFG.VALI := 0;
            VAL := false;
            DIVARCFG.STEP1 := 400;
            DIVARCFG.T_STEP1 := 0;
        END_IF;
    16#0003: (*TOGGLE - тільки при форсуванні*)
        IF FRC AND INBUF THEN
            IF DIVARCFG.VALI > 0 THEN
                DIVARCFG.VALI := 0;
                VAL := false;
                DIVARCFG.STEP1 := 400;
                DIVARCFG.T_STEP1 := 0;
            ELSE

```

```

        DIVARCFG.VALI := 1;
        VAL := true;
        DIVARCFG.STEP1 := 401;
        DIVARCFG.T_STEP1 := 0;
    END_IF;
END_IF;
16#0100: (*прочитати конфігурацію*)
    VARBUF1.MSG := 201;
    VARBUF1.PRM1.%X0 := PRM_ISALM;
    VARBUF1.PRM1.%X1 := PRM_ISWRN;
    VARBUF1.PRM1.%X2 := PRM_INVERSE;
    VARBUF1.PRM1.%X5 := PRM_NRMVAL;
    VARBUF1.PRM1.%X6 := PRM_QALENBL;
    VARBUF1.PRM1.%X7 := PRM_DSBL;
    VARBUF1.ID := DIVARCFG.ID;
    VARBUF1.CLSID := DIVARCFG.CLSID;
    VARBUF1.CHID := DIVARCFG.CHID;
    VARBUF1.VALR := INT_TO_REAL(DIVARCFG.VALI);
    VARBUF1.T_FLTSP := DIVARCFG.T_FLTSP;
    VARBUF1.T_DEALL := DIVARCFG.T_DEASP;
    (*//VARBUF.STEP1 := DIVARCFG.STEP1;*)
    (*VARBUF.T_STEP1 := DIVARCFG.T_STEP1;*)

16#0101: (*записати конфігурацію*)
    (* MSG 200-Ok 400-Error
    // 200 - Дані записані
    // 201 - Дані прочитані
    // 403 - канал вже зайнятий
    // 404 - номер каналу не відповідає діапазону *)
    PRM_ISALM := VARBUF1.PRM1.%X0;
    PRM_ISWRN := VARBUF1.PRM1.%X1;
    PRM_INVERSE := VARBUF1.PRM1.%X2;
    PRM_NRMVAL := VARBUF1.PRM1.%X5;
    PRM_QALENBL := VARBUF1.PRM1.%X6;
    PRM_DSBL := VARBUF1.PRM1.%X7;

    VARBUF1.MSG:=200;
    IF VARBUF1.CHID>=0 AND VARBUF1.CHID <= PLCCFG.DICNT THEN
        IF CHDI[VARBUF1.CHID].VARID = 0 THEN
            DIVARCFG.CHID := VARBUF1.CHID;
        ELSIF VARBUF1.CHID <> DIVARCFG.CHID THEN
            VARBUF1.MSG := 403; (* канал вже зайнятий*)
        END_IF;
    ELSE
        VARBUF1.MSG := 404; (*номер каналу не відповідає діапазону*)
    END_IF;

    IF INBUF THEN
        VARBUF1.CHID := DIVARCFG.CHID;
    END_IF;
    DIVARCFG.T_FLTSP := VARBUF1.T_FLTSP;
    DIVARCFG.T_DEASP := VARBUF1.T_DEALL;
16#0102: (*записати значення за замовченням*)
    DIVARCFG.CHID := DIVARCFG.CHIDDF;
;
16#0300: (*перемкнути форсування*)
    FRC := NOT FRC;
16#0301: (*форсувати один/усі об'єкти типу*)
    FRC := true;
16#0302: (*дефорсувати один/усі об'єкти типу*)
    FRC := false;
16#0311: (* імітувати*)
    SML := true;

```

```

16#0312: (* зняти режим імітації*)
    SML := false;
16#0400: (* запуск калібрування*)
    DIVARCFG.STA1.STA_ISALM := true;
END_CASE;
(*обробка значень*)
IF NOT FRC AND NOT SML THEN
    IF PRM_INVERSE THEN
        DI := NOT VRAW;
    ELSE
        DI := VRAW;
    END_IF;
CASE DIVARCFG.STEP1 OF
    400: (*DI =0*)
        IF DI THEN
            DIVARCFG.STEP1 := 401;
            DIVARCFG.T_STEP1 := 0;
        END_IF;
        IF T_STEPMS >= UINT_TO_UDINT(DIVARCFG.T_FLTSP) THEN
            VAL := FALSE;
        END_IF;
    401: (*DI =1*)
        IF NOT DI THEN
            DIVARCFG.STEP1 := 400;
            DIVARCFG.T_STEP1 := 0;
        END_IF;
        IF T_STEPMS >= UINT_TO_UDINT(DIVARCFG.T_FLTSP) THEN
            VAL := true;
        END_IF;
    ELSE
        DIVARCFG.STEP1 := 400;
        DIVARCFG.T_STEP1 := 0;
END_CASE;
(*якщо це не DI з лічильником*)
IF DIVARCFG.CLSID <> 16#1011 THEN
    DIVARCFG.VALI := TO_INT(VAL);
END_IF;
ELSE (* обробка без фільтрації та інверсії*)
    IF FRC THEN (*при форсування значення береться з VALI*)
        (*якщо це не DI з лічильником*)
        IF DIVARCFG.CLSID <> 16#1011 THEN
            VAL := INT_TO_BOOL (DIVARCFG.VALI);
        END_IF;
    ELSIF SML THEN (*в режимі імітації VAL змінюється ззовні а VALI береться з
нього*)
        (*якщо це не DI з лічильником*)
        IF DIVARCFG.CLSID <> 16#1011 THEN
            DIVARCFG.VALI := TO_INT(VAL);
        END_IF;
    END_IF;
CASE DIVARCFG.STEP1 OF
    400: (*DI =0*)
        IF VAL THEN
            DIVARCFG.STEP1 := 401;
            DIVARCFG.T_STEP1 := 0;
        END_IF;
    401: (*DI =1*)
        IF NOT VAL THEN
            DIVARCFG.STEP1 := 400;
            DIVARCFG.T_STEP1 := 0;
        END_IF;
    ELSE
        DIVARCFG.STEP1 := 400;

```

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Лист | № докум. | Підпис | Дата |

ТО8151.0014.001.АТХ.П

Арк

73

```

        DIVARCFG.T_STEP1 := 0;
    END_CASE;
END_IF;
(*обработка тривог*)
IF VARENBL AND DIVARCFG.T_STEP1 >= UINT_TO_UDINT(DIVARCFG.T_DEASP) * 100 THEN
    ALM := NOT (PRM_NRMVAL = VAL) AND PRM_ISALM;
    WRN := NOT (PRM_NRMVAL = VAL) AND PRM_ISWRN;
ELSE
    ALM := false;
    WRN := false;
END_IF;
BAD := CHCFG.STA1.STA_BAD AND PRM_QALENBL AND NOT SML;

IF BAD THEN
    PLCCFG.ALM.BAD := true;
    PLCCFG.CNTBAD := PLCCFG.CNTBAD + 1;
    IF NOT DIVARCFG.STA1.STA_BAD THEN
        PLCCFG.ALM.NWBAD := true;
    END_IF;
END_IF;

IF ALM THEN
    PLCCFG.ALM.ALM := true;
    PLCCFG.CNTALM := PLCCFG.CNTALM + 1;
    IF NOT DIVARCFG.STA1.STA_ALM THEN
        PLCCFG.ALM.NWALM := true;
    END_IF;
END_IF;

IF WRN THEN
    PLCCFG.ALM.WRN := true;
    PLCCFG.CNTWRN := PLCCFG.CNTWRN + 1;
    IF NOT DIVARCFG.STA1.STA_WRN THEN
        PLCCFG.ALM.NWWRN := true;
    END_IF;
END_IF;

IF FRC THEN
    PLCCFG.STA1.FRC1 := true;
    PLCCFG.CNTFRC := PLCCFG.CNTFRC + 1;
END_IF;

IF SML THEN
    PLCCFG.STA1.SML := true;
END_IF;

ISALM := PRM_ISALM;
ISWRN := PRM_ISWRN;
DIVARCFG.STA1.STA_VRAW := VRAW;
DIVARCFG.STA1.STA_VALB := VAL;
DIVARCFG.STA1.STA_BAD := BAD;
DIVARCFG.STA1.STA_ALDIS := ALDIS;
DIVARCFG.STA1.STA_DLNK := DLNK;
DIVARCFG.STA1.STA_ENBL := VARENBL;
DIVARCFG.STA1.STA_ALM := ALM;
DIVARCFG.STA1.STA_VALPRV := VALPRV;
DIVARCFG.STA1.STA_ISALM := ISALM;
DIVARCFG.STA1.STA_ISWRN := ISWRN;
DIVARCFG.STA1.STA_WRN := WRN;
DIVARCFG.STA1.STA_INBUF := INBUF;
DIVARCFG.STA1.STA_FRC := FRC;
DIVARCFG.STA1.STA_SML := SML;
DIVARCFG.STA1.STA_CMDLOAD := FALSE;
DIVARCFG.PRMI.PRMI_ISALM := PRM_ISALM;
DIVARCFG.PRMI.PRMI_ISWRN := PRM_ISWRN;
DIVARCFG.PRMI.PRMI_INVERSE := PRM_INVERSE;

```

```

DIVARCFG.PRM1.PRM_NRMVAL := PRM_NRMVAL;
DIVARCFG.PRM1.PRM_QALENBL := PRM_QALENBL;
DIVARCFG.PRM1.PRM_DSBL := PRM_DSBL;
DIVARHMI.STA1 := DIVARCFG.STA1;
DIVARCFG.T_PREV := PLCCFG.TQMS;
DIVARCFG.T_STEP1 := DIVARCFG.T_STEP1 + dt1;
IF DIVARCFG.T_STEP1 > 16#7FFF_FFFF THEN
    DIVARCFG.T_STEP1 := 16#7FFF_FFFF;
END_IF;

IF INBUF THEN
    VARBUF1.CMD := 0;
    VARBUF1.VALR := INT_TO_REAL(DIVARCFG.VALI);
    VARBUF1.STA_DI := DIVARCFG.STA1;
    VARBUF1.STEP1 := DIVARCFG.STEP1;
    VARBUF1.T_STEP1 := DIVARCFG.T_STEP1;
    VARBUF1.CH_CLSID := CHCFG.CLSID;
    VARBUF1.CH_STA.%X0 := CHCFG.STA1.STA_VRAW;
    VARBUF1.CH_STA.%X1 := CHCFG.STA1.STA_VALB;
    VARBUF1.CH_STA.%X2 := CHCFG.STA1.STA_BAD;
    VARBUF1.CH_STA.%X3 := CHCFG.STA1.STA_b3;
    VARBUF1.CH_STA.%X4 := CHCFG.STA1.STA_PNG;
    VARBUF1.CH_STA.%X5 := CHCFG.STA1.STA_ULNK;
    VARBUF1.CH_STA.%X6 := CHCFG.STA1.STA_MERR;
    VARBUF1.CH_STA.%X7 := CHCFG.STA1.STA_BRK;
    VARBUF1.CH_STA.%X8 := CHCFG.STA1.STA_SHRT;
    VARBUF1.CH_STA.%X9 := CHCFG.STA1.STA_NBD;
    VARBUF1.CH_STA.%X10 := CHCFG.STA1.STA_b10;
    VARBUF1.CH_STA.%X11 := CHCFG.STA1.STA_INIOTBUF;
    VARBUF1.CH_STA.%X12 := CHCFG.STA1.STA_INBUF;
    VARBUF1.CH_STA.%X13 := CHCFG.STA1.STA_FRC;
    VARBUF1.CH_STA.%X14 := CHCFG.STA1.STA_SML;
    VARBUF1.CH_STA.%X15 := CHCFG.STA1.STA_CMDLOAD;
END_IF;

```

DOVARFN

Код реалізації функціонального блоку DOVARFN:

```

IF PLCCFG.STA1.SCN1 THEN
    DOVARCFG.CLSID := 16#1020;
    DOVARCFG.PRM1.PRM_DSBL := FALSE;
    DOVARCFG.T_PREV := PLCCFG.TQMS;
    IF DOVARCFG.CHID = 0 THEN
        DOVARCFG.CHID := DOVARCFG.CHIDDF;
    END_IF;
    IF CHCFG.ID > 0 THEN
        CHCFG.STA1.STA_VALB := DOVARCFG.STA1.STA_VALB;
    ELSE
        CHCFG.STA1.STA_VALB := 0;
    END_IF;
    DOVARCFG.T_STEP1 := 0;
    DOVARCFG.STEP1 := 400;
    RETURN;
END_IF;

VRAW := DOVARCFG.STA1.STA_VRAW;
VAL := DOVARCFG.STA1.STA_VALB;
BAD := DOVARCFG.STA1.STA_BAD;
DLNK := DOVARCFG.STA1.STA_DLNK;
ENBL := DOVARCFG.STA1.STA_ENBL;
VALPRV := DOVARCFG.STA1.STA_VALPRV;
MVAL := DOVARCFG.STA1.STA_MVAL;

```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 75 |

```

INBUF := DOVARCFG.STA1.STA_INBUF;
FRC := DOVARCFG.STA1.STA_FRC;
SML := DOVARCFG.STA1.STA_SML;
CMDLOAD := DOVARCFG.STA1.STA_CMDLOAD;

PRM_INVERSE := DOVARCFG.PRM1.PRM_INVERSE;
PRM_QALENBL := DOVARCFG.PRM1.PRM_QALENBL;
PRM_DSBL := DOVARCFG.PRM1.PRM_DSBL;
INBUF := (DOVARCFG.ID = VARBUF1.ID) AND (DOVARCFG.CLSID = VARBUF1.CLSID);
CMDLOAD := DOVARHMI.STA1.STA_CMDLOAD;
CMD := 0;
DLNK := (CHCFG.ID > 0);
VARENBL := NOT PRM_DSBL AND DLNK;
VRAW := CHCFG.STA1.STA_VALB;
T_STEPMS := DOVARCFG.T_STEP1;
(*ping-pong*)
IF DLNK THEN
    CHCFG.STA1.STA_PNG := true;
    CHCFG.VARID := DOVARCFG.ID;
END_IF;

IF NOT VARENBL THEN
    (* DOVARCFG.STA.VRAW := VRAW; *)
    DOVARCFG.T_STEP1 := 0;
    DOVARCFG.STEP1 := 400;
END_IF;

dt1 := PLCCFG.TQMS - DOVARCFG.T_PREV;

(* ширококомвне дефорсування*)
IF PLCCFG.CMD = 16#4302 THEN
    FRC := false; (*дефорсувати об'єкт типу*)
END_IF;

IF CMDLOAD THEN
    CMD := 16#0100;
ELSIF INBUF AND VARBUF1.CMD <> 0 THEN
    CMD := VARBUF1.CMD;
END_IF;

(*commands*)
CASE CMD OF
    16#0001: (*записати 1*)
        IF FRC AND INBUF THEN
            DOVARCFG.VALI := 1;
            VARBUF1.VALR:=1.0;
            VAL := true;
            DOVARCFG.STEP1 := 401;
            DOVARCFG.T_STEP1 := 0;
        END_IF;
    16#0002: (*записати 0*)
        IF FRC AND INBUF THEN
            DOVARCFG.VALI := 0;
            VARBUF1.VALR:=0.0;
            VAL := false;
            DOVARCFG.STEP1 := 400;
            DOVARCFG.T_STEP1 := 0;
        END_IF;
    16#0003: (*TOGGLE*)
        IF FRC AND INBUF THEN
            IF DOVARCFG.VALI > 0 THEN
                DOVARCFG.VALI := 0;
                VARBUF1.VALR:=0.0;
            END_IF;
        END_IF;

```

```

        VAL := false;
        DOVARCFG.STEP1 := 400;
        DOVARCFG.T_STEP1 := 0;
    ELSE
        DOVARCFG.VALI := 1;
        VARBUF1.VALR:=1.0;
        VAL := true;
        DOVARCFG.STEP1 := 401;
        DOVARCFG.T_STEP1 := 0;
    END_IF;
END_IF;
16#0100: (*прочитати конфігурацію*)
    VARBUF1.MSG:=201;
    VARBUF1.PRM1.%X2 := PRM_INVERSE;
    VARBUF1.PRM1.%X6 := PRM_QALENBL;
    VARBUF1.PRM1.%X7 := PRM_DSBL;

    VARBUF1.ID := DOVARCFG.ID;
    VARBUF1.CLSID := DOVARCFG.CLSID;
    VARBUF1.CHID := DOVARCFG.CHID;
    VARBUF1.VALR := INT_TO_REAL(DOVARCFG.VALI);
16#0101: (*записати конфігурацію*)
    (* MSG 200-Ok 400-Error
    // 200 - Дані записані
    // 201 - Дані прочитані
    // 403 - канал вже зайнятий
    // 404 - номер каналу не відповідає діапазону *)
    PRM_INVERSE := VARBUF1.PRM1.%X2;
    PRM_QALENBL := VARBUF1.PRM1.%X6;
    PRM_DSBL := VARBUF1.PRM1.%X7;
    (*перевіряти незайнятість каналу треба в функції контролю буферу *)

    VARBUF1.MSG:=200;
    IF VARBUF1.CHID>=0 AND VARBUF1.CHID <= PLCCFG.DOCNT THEN
        IF CHDO[VARBUF1.CHID].VARID = 0 THEN
            DOVARCFG.CHID := VARBUF1.CHID;
        ELSIF VARBUF1.CHID <> DOVARCFG.CHID THEN
            VARBUF1.MSG := 403; (* канал вже зайнятий*)
        END_IF;
    ELSE
        VARBUF1.MSG := 404; (*номер каналу не відповідає діапазону*)
    END_IF;

    IF INBUF THEN
        VARBUF1.CHID := DOVARCFG.CHID;
    END_IF;

16#0102: (*записати значення за замовченням*)
    DOVARCFG.CHID := DOVARCFG.CHIDDF;
;
16#0300: (*перемкнути форсування*)
    FRC := NOT FRC;
16#0301: (*форсувати один/усі об'єкти типу*)
    FRC := true;
16#0302: (*дефорсувати один/усі об'єкти типу*)
    FRC := false;
16#0311: (* імітувати*)
    SML := true;
16#0312: (* зняти режим імітації*)
    SML := false;
END_CASE;
(*val*)
IF FRC THEN (*режим форсування*)

```

```

IF INBUF THEN
    DOVARCFG.VALI := REAL_TO_INT (VARBUF1.VALR);
END_IF;
VAL := DOVARCFG.VALI > 0;
IF VAL THEN
    DOVARCFG.STEP1 := 401;
ELSE
    DOVARCFG.STEP1 := 400;
END_IF;
ELSE
    IF VAL THEN
        DOVARCFG.VALI := 1;
        DOVARCFG.STEP1 := 401;
    ELSE
        DOVARCFG.VALI := 0;
        DOVARCFG.STEP1 := 400;
    END_IF;
END_IF;
(*фронт*)
IF VAL <> VALPRV THEN
    DOVARCFG.T_STEP1 := 0;
END_IF;

IF VARENBL THEN
    IF PRM_INVERSE THEN
        VRAW := NOT VAL;
    ELSE
        VRAW := VAL;
    END_IF;
    IF NOT SML THEN
        CHCFG.STA1.STA_VALB := VRAW;
    END_IF; (* режим симуляції - вихідне значення не змінюється *)
END_IF;
(*недостовірність каналу*)
BAD := CHCFG.STA1.STA_BAD AND PRM_QALENBL AND VARENBL AND NOT SML;
IF BAD THEN
    PLCCFG.ALM.BAD := true;
    PLCCFG.CNTBAD := PLCCFG.CNTBAD + 1;
    IF NOT DOVARCFG.STA1.STA_BAD THEN
        PLCCFG.ALM.NWBAD := true;
    END_IF;
END_IF;
IF FRC THEN
    PLCCFG.STA1.FRC1 := true;
    PLCCFG.CNTFRC := PLCCFG.CNTFRC + 1;
END_IF;
IF SML THEN
    PLCCFG.STA1.SML := true;
END_IF;

DOVARCFG.STA1.STA_VRAW := VRAW;
DOVARCFG.STA1.STA_VALB := VAL;
DOVARCFG.STA1.STA_BAD := BAD;
DOVARCFG.STA1.STA_DLNK := DLNK;
DOVARCFG.STA1.STA_ENBL := VARENBL;
DOVARCFG.STA1.STA_VALPRV:=VAL;
DOVARCFG.STA1.STA_MVAL := DOVARCFG.VALI > 0;
DOVARCFG.STA1.STA_INBUF := INBUF;
DOVARCFG.STA1.STA_FRC := FRC;
DOVARCFG.STA1.STA_SML := SML;
DOVARCFG.STA1.STA_CMDLOAD := FALSE;
DOVARCFG.PRM1.PRM_INVERSE := PRM_INVERSE;
DOVARCFG.PRM1.PRM_QALENBL := PRM_QALENBL;

```

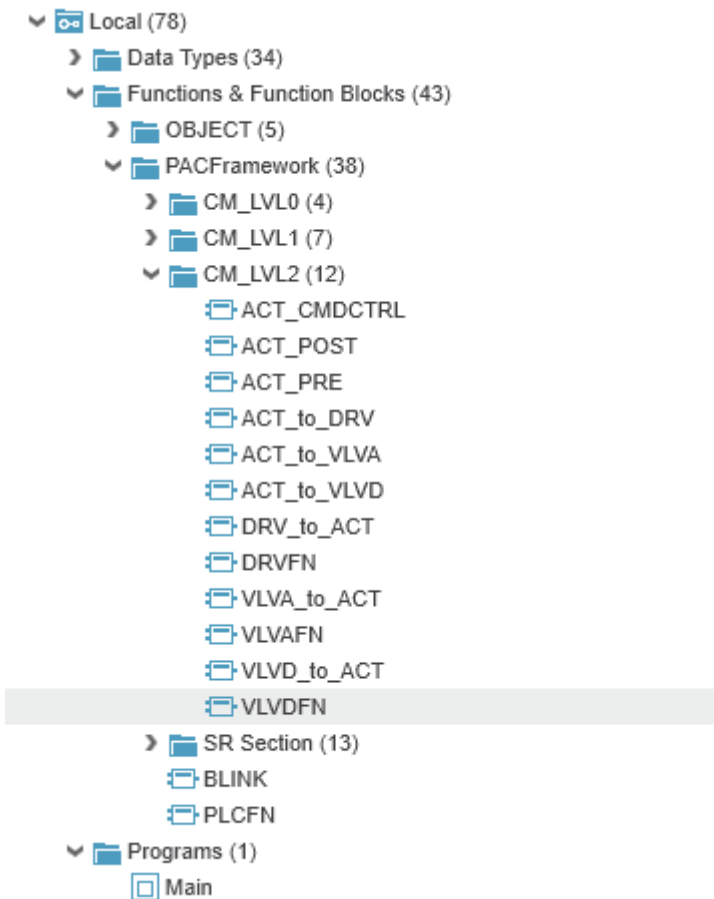



Рисунок 3.4 - Зовнішній вигляд структури функцій LVL2

VLVDNFN

Код функціонального блоку VLVDNFN:

```

VLVD_to_ACT1(VLVD CFG := ACTCFG, VLVDHMI := ACTHMI, ACTCFG := ACTCFGu);
(*попередня обробка: ініт STA, ALM, CMD, INBUF, SML, dt *)
ACT_PRE1(ACTCFG := ACTCFGu, STA1 := STA1, ALMs := ALMS, CMD := CMD, dt1 := dt1,
PLCCFG := PLCCFG, ACTBUF := ACTBUF);
(*значення за замовченням*)
IF PLCCFG.STA1.SCN1 THEN (*first scan*)
  IF ACTCFGu.T_OPNSP <= 0 THEN (*якщо уставка часу відкриття не виставлена*)
    ACTCFGu.T_OPNSP := 500; (*5 секунд*)
  END_IF;
  IF ACTCFGu.T_DEASP <= 0 THEN (*якщо уставка затримки часу тривоги не виставлена*)
    ACTCFGu.T_DEASP := 200; (*2 секунди*)
  END_IF;
  (*технологічні тривоги для датчиків не використовуються*)
  IF SOPN.ID <> 0 THEN
    SOPN.PRM1.PRM_ISALM := false; (*ISALM*)
    SOPN.PRM1.PRM_ISWRN := false; (*ISWRN*)
  END_IF;
  IF SCLS.ID <> 0 THEN
    SCLS.PRM1.PRM_ISALM := false; (*ISALM*)
    SCLS.PRM1.PRM_ISWRN := false; (*ISWRN*)
  END_IF;
END_IF;
(* ----- блок параметрів*)
(*параметри перевірка наявності/використання датчиків на вході*)
ACTCFGu.PRM1.PRM1.PRM_MANCFG:=false;(*у цьому проекті не буде ручного конфігурування
параметрів IO*)

```

| | | | | | | | |
|-----|------|----------|--------|------|--|-----------------------|-----|
| | | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | | 80 |

```

ACTCFGU.PRM1.PRM1.PRM_ZCLSENBL := NOT SCLS.PRM1.PRM_DSBL AND SCLS.ID <> 0;
ACTCFGU.PRM1.PRM1.PRM_ZOPENBL := NOT SOPN.PRM1.PRM_DSBL AND SOPN.ID <> 0;
ACTCFGU.PRM1.PRM1.PRM_ZPOSENBL := FALSE;
(*параметри імпульсного керування*)
ACTCFGU.PRM1.PRM1.PRM_PULSCTRLLENBL := ACTCFGU.CLSID=16#2014;
(*----- блок для режиму імітації*)
(*режим імітації підлеглих від хозяїна*)
SOPN.STA1.STA_SML := STA1.STA1.STA_SML;
SCLS.STA1.STA_SML := STA1.STA1.STA_SML;
COPN.STA1.STA_SML := STA1.STA1.STA_SML;
CCLS.STA1.STA_SML := STA1.STA1.STA_SML;

(*логіка для режиму імітації *)
IF STA1.STA1.STA_SML THEN
  dpos:=
  UDINT_TO_REAL(dt1*100)/UDINT_TO_REAL((UINT_TO_UDINT(ACTCFGU.T_OPNSP)*100));(*%*)
  IF ACTCFGU.STEP1=2 THEN
    ACTCFGU.POS := ACTCFGU.POS + dpos; (*0-10000*)
  END_IF;
  IF ACTCFGU.STEP1=3 THEN
    ACTCFGU.POS := ACTCFGU.POS - dpos; (*0-10000*)
  END_IF;
  IF ACTCFGU.POS<0.0 THEN ACTCFGU.POS:=0.0; END_IF;
  IF ACTCFGU.POS>100.0 THEN ACTCFGU.POS:=100.0; END_IF;
  (*імітація датчиків *)
  IF NOT SOPN.STA1.STA_FRC THEN
    SOPN.STA1.STA_VALB:= STA1.STA1.STA_OPNING AND ACTCFGU.T_STEP1 >=
    UINT_TO_UDINT(ACTCFGU.T_OPNSP)*100 OR STA1.STA1.STA_OPND;
  END_IF;
  IF NOT SCLS.STA1.STA_FRC THEN
    SCLS.STA1.STA_VALB:= STA1.STA1.STA_CLSING AND ACTCFGU.T_STEP1 >=
    UINT_TO_UDINT(ACTCFGU.T_OPNSP)*100 OR STA1.STA1.STA_CLSD;
  END_IF;
END_IF;
(*----- блок обробки команд *)
(*стандартний обробник команд*)
ACT_CMDCTRL1(ACTCFG := ACTCFGU, STA1 := STA1, CMD := CMD, ACTBUF := ACTBUF);
(*----- блок обробки станів датчиків відкриття/закриття, або їх
заміна на логіку*)
IF NOT ACTCFGU.PRM1.PRM1.PRM_ZOPENBL THEN
  SOPN1:= STA1.STA1.STA_OPNING AND ACTCFGU.T_STEP1 >=
  UINT_TO_UDINT(ACTCFGU.T_OPNSP)*100
  OR STA1.STA1.STA_OPND;
ELSE
  SOPN1:=SOPN.STA1.STA_VALB;
END_IF;
IF NOT ACTCFGU.PRM1.PRM1.PRM_ZCLSENBL THEN
  SCLS1:=STA1.STA1.STA_CLSING AND ACTCFGU.T_STEP1 >=
  UINT_TO_UDINT(ACTCFGU.T_OPNSP)*100
  OR STA1.STA1.STA_CLSD;
ELSE
  SCLS1:=SCLS.STA1.STA_VALB;
END_IF;

IF NOT ACTCFGU.PRM1.PRM1.PRM_ZPOSENBL THEN
  SPOS1:=-10000;
ELSE
  ;
END_IF;
(*//----- автомат станів позиції та тривог позиції та основне керування
//0 - ініціалізація, 1- зупинено в проміжному стані, 4 - зупинений у відкритому
стані, 5 - зупинений в закритому стані

```

```

//7 - рухається в довільному напрямку (ручний зі щита), 2 - відкривається , 3 -
закривається
//керування автоматом станів див в наступному блоці
//керування виходами COPN, CCLS згідно логіки*)
CASE ACTCFGU.STEP1 OF
  0: (*ініціалізація*)
    ACTCFGU.STEP1 := 1;
    ACTCFGU.T_STEP1 := 0;
  1, 4, 5: (*зупинений, 1 - зупинено в проміжному стані 4 - зупинений у відкритому
стані, 5 - зупинений в закритому стані*)
    IF SOPN1 AND NOT SCLS1 THEN
      IF ACTCFGU.STEP1<>4 THEN ACTCFGU.T_STEP1 := 0;END_IF;
      ACTCFGU.STEP1 := 4;
    ALMS.ALM_1.ALM_ALMSHFT:=FALSE;
    ALMS.ALM_1.ALM_ALMCLS := FALSE;
    ALMS.ALM_1.ALM_ALMOPN := FALSE;
    END_IF;
    IF SCLS1 AND NOT SOPN1 THEN
      IF ACTCFGU.STEP1<>5 THEN ACTCFGU.T_STEP1 := 0;END_IF;
      ACTCFGU.STEP1 := 5;
    ALMS.ALM_1.ALM_ALMSHFT:=FALSE;
    ALMS.ALM_1.ALM_ALMCLS := FALSE;
    ALMS.ALM_1.ALM_ALMOPN := FALSE;
    END_IF;
    (*здвиг (показують датчики)*)
    IF (ACTCFGU.STEP1 = 4 AND NOT SOPN1 OR (ACTCFGU.STEP1 = 5 AND NOT SCLS1))
THEN
      ACTCFGU.T_STEP1 := 0;
      ACTCFGU.STEP1 := 7; (*рухається в невизначеному напрямку (ручний зі
щита)*)
    END_IF;
    IF SOPN1 AND SCLS1 THEN
      ALMS.ALM_1.ALM_ALMSHFT:=true;
      ACTCFGU.STEP1 := 1;
    END_IF;
  2: (*Відкривається*)
    ALMS.ALM_1.ALM_ALMSTP:=false;
    IF SOPN1 AND NOT SCLS1 THEN
      ACTCFGU.STEP1 := 4; (*у стан відкрито*)
      ACTCFGU.T_STEP1 := 0;
    END_IF;
    ALMS.ALM_1.ALM_ALMOPN := FALSE; ALMS.ALM_1.ALM_ALMCLS :=
FALSE;ALMS.ALM_1.ALM_ALMSTP:= false;
    IF ACTCFGU.T_STEP1 >= (UINT_TO_UDINT(ACTCFGU.T_OPNSP)*100 +
UINT_TO_UDINT(ACTCFGU.T_DEASP)*100) THEN
      ALMS.ALM_1.ALM_ALMOPN := TRUE;
      ALMS.ALM_1.ALM_ALMCLS := FALSE;
      IF SCLS1 AND NOT SOPN1 THEN
        ACTCFGU.STEP1 := 5; (*у стан закрито*)
        ACTCFGU.T_STEP1 := 0;
      END_IF;
    END_IF;
  3: (*закривається*)
    ALMS.ALM_1.ALM_ALMSTP:=false;
    IF SCLS1 AND NOT SOPN1 THEN
      ACTCFGU.STEP1 := 5;
      ACTCFGU.T_STEP1 := 0;
    END_IF;
    IF ACTCFGU.T_STEP1 >= UINT_TO_UDINT(ACTCFGU.T_OPNSP)*100 +
UINT_TO_UDINT(ACTCFGU.T_DEASP)*100 THEN
      ALMS.ALM_1.ALM_ALMCLS := TRUE;
      ALMS.ALM_1.ALM_ALMOPN := FALSE;
      IF SOPN1 AND NOT SCLS1 THEN

```

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Лист | № докум. | Підпис | Дата |

ТО8151.0014.001.АТХ.П

Арк

82

```

        ACTCFGU.STEP1 := 2;
        ACTCFGU.T_STEP1 := 0;
        END_IF;
    END_IF;
7: (*рухається в довільному напрямку (ручний зі щита) чи помилка довільного
зсуву*)
    ALMS.ALM_1.ALM_ALMSHFT:=true;
    ACTCFGU.STEP1 := 1;
    ACTCFGU.T_STEP1 := 0;
ELSE (*невизначеність*)
    ACTCFGU.STEP1 := 0;
END_CASE;
(*автомат станів побітово*)
STA1.STA1.STA_IMSTPD:= ACTCFGU.STEP1 = 1;
STA1.STA1.STA_MANRUNING:= ACTCFGU.STEP1 = 7;
STA1.STA1.STA_OPNING:= ACTCFGU.STEP1 = 2;
STA1.STA1.STA_CLSING := ACTCFGU.STEP1 = 3;
STA1.STA1.STA_OPND:= ACTCFGU.STEP1 = 4;
STA1.STA1.STA_CLSD:= ACTCFGU.STEP1 = 5;

IF ACTCFGU.PRM1.PRM1.PRM_ZPOSENBL THEN
    ;(*ACTCFG.POS := REAL_TO_INT(POS.VAL * 0.01);*)
ELSIF NOT STA1.STA1.STA_SML THEN
    ACTCFGU.POS := -32768.0; (*відключено для показу*)
END_IF;
(*----- блокування приводу*)
(*умова блокування *)
IF CMD.CMD1.CMD_BLCK THEN
    STA1.STA1.STA_BLCK:=TRUE;
    ALMS.ALM_1.ALM_ALMSTP:=false;
END_IF;
(*умова розблокування *)
IF CMD.CMD1.CMD_DBLCK THEN
    STA1.STA1.STA_BLCK:=FALSE;
    ALMS.ALM_1.ALM_ALMSTP:=false;
END_IF;
(*умова блокування , умова спрацьовує тільки при початку блокування *)
IF (ALMS.ALM_1.ALM_ALMPWR OR CMD.CMD1.CMD_PROTECT) AND NOT STA1.STA1.STA_BLCK THEN
    STA1.STA1.STA_BLCK:=true;
END_IF;
(*стан блокування*)
IF STA1.STA1.STA_BLCK THEN
    ;(*//**)
END_IF;
(*-----виходи керування *)
(*керування OPN/CLS тільки при дозволі керування або тимчасовому розблокуванні та при
відсутності блокування*)
IF (CMD.CMD2.CMD_RESOLUTION OR PLCCFG.STA_PERM.%X6) AND NOT STA1.STA1.STA_BLCK THEN
    IF CMD.CMD1.CMD_CLS THEN CMD.CMD1.CMD_OPN:=FALSE; END_IF; (*CLS має пріоритет над
OPN*)
    IF CMD.CMD1.CMD_CLS AND ACTCFGU.STEP1 <> 5 AND ACTCFGU.STEP1 <> 3 THEN
        ACTCFGU.STEP1 := 3;
        ACTCFGU.T_STEP1 := 0;
        ACTCFGU.CNTPER := ACTCFGU.CNTPER + 1;
        COPN.STA1.STA_VALB:= FALSE;
        CCLS.STA1.STA_VALB:= TRUE;
        END_IF;
    IF CMD.CMD1.CMD_OPN AND ACTCFGU.STEP1 <> 4 AND ACTCFGU.STEP1 <> 2 THEN
        ACTCFGU.STEP1 := 2;
        ACTCFGU.T_STEP1 := 0;
        ACTCFGU.CNTPER := ACTCFGU.CNTPER + 1;
        COPN.STA1.STA_VALB:= TRUE;
        CCLS.STA1.STA_VALB:= FALSE;

```

| | | | | | | | |
|-----|------|----------|--------|------|--|-----------------------|-----|
| | | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | | 83 |

```

    END_IF;
END_IF;
(*----- режими*)
(*місцевий режим керування*)
STA1.STA1.STA_MANBX := CMD.CMD2.CMD_CLCL;
(*у місцевому режимі керування відключати усі виходи і перевести в ручний режим*)
IF STA1.STA1.STA_MANBX THEN
    COPN.STA1.STA_VALB:=FALSE;
    CCLS.STA1.STA_VALB:= FALSE;
    STA1.STA1.STA_DISP:=true;
END_IF;

STA1.STA1.STA_FRC := COPN.STA1.STA_FRC AND COPN.ID<>0
    OR CCLS.STA1.STA_FRC AND CCLS.ID<>0
    OR SOPN.STA1.STA_FRC AND SOPN.ID<>0
    OR SCLS.STA1.STA_FRC AND SCLS.ID<>0;
(*----- зведення кастомних тривог, режимів, бітів*)
IF ALMS.ALM_1.ALM_ALMOPN THEN
    PLCCFG.ALM.ALM := TRUE;
    IF NOT ACTCFGu.ALM.ALM_1.ALM_ALMOPN THEN
        PLCCFG.ALM.NWALM := TRUE;
        ACTCFGu.CNTALM := ACTCFGu.CNTALM + 1;
    END_IF;
END_IF;
IF ALMS.ALM_1.ALM_ALMCLS THEN
    PLCCFG.ALM.ALM := TRUE;
    IF NOT ACTCFGu.ALM.ALM_1.ALM_ALMCLS THEN
        PLCCFG.ALM.NWALM := TRUE;
        ACTCFGu.CNTALM := ACTCFGu.CNTALM + 1;
    END_IF;
END_IF;
IF ALMS.ALM_1.ALM_ALMOPN2 THEN
    PLCCFG.ALM.ALM := TRUE;
    IF NOT ACTCFGu.ALM.ALM_1.ALM_ALMOPN2 THEN
        PLCCFG.ALM.NWALM := TRUE;
        ACTCFGu.CNTALM := ACTCFGu.CNTALM + 1;
    END_IF;
END_IF;
IF ALMS.ALM_1.ALM_ALMCLS2 THEN
    PLCCFG.ALM.ALM := TRUE;
    IF NOT ACTCFGu.ALM.ALM_1.ALM_ALMCLS2 THEN
        PLCCFG.ALM.NWALM := TRUE;
        ACTCFGu.CNTALM := ACTCFGu.CNTALM + 1;
    END_IF;
END_IF;
IF ALMS.ALM_1.ALM_ALMSTP THEN
    PLCCFG.ALM.ALM := TRUE;
    IF NOT ACTCFGu.ALM.ALM_1.ALM_ALMSTP THEN
        PLCCFG.ALM.NWALM := TRUE;
        ACTCFGu.CNTALM := ACTCFGu.CNTALM + 1;
    END_IF;
END_IF;

ALMS.ALM_1.ALM_ALM:=ALMS.ALM_1.ALM_ALMOPN OR ALMS.ALM_1.ALM_ALMCLS OR
ALMS.ALM_1.ALM_ALMOPN2 OR ALMS.ALM_1.ALM_ALMCLS2 OR ALMS.ALM_1.ALM_ALMSTP
    OR ALMS.ALM_1.ALM_ALMSTPBTN OR ALMS.ALM_1.ALM_ALMPWR OR
STA1.STA1.STA_BLK;
(*заклучна обробка: зведення в PLC.CFG, STA, ALM, CMD, INBUF, SML, dt *)
ACT_POST1(ACTCFG := ACTCFGu, STA1 := STA1, ALMS := ALMS, CMD := CMD, dt1 := dt1,
ACTBUF := ACTBUF, PLCCFG := PLCCFG);

ACT_to_VLVD1(VLVDCFG := ACTCFG, VLVDHMI := ACTHMI, ACTCFG := ACTCFGu);

```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 84 |

VLVAFN

Код функціонального блоку VLVAFN:

```
VLVA_to_ACT1(VLVACFG:=ACTCFG, VLVAHMI:=ACTHMI, ACTCFG:=ACTCFGu);
(*попередня обробка: ініт STA, ALM, CMD, INBUF, SML, dt *)
ACT_PRE1(ACTCFG := ACTCFGu, STA1 := STA1, ALMs := ALMs, CMD := CMD, dt1 := dt1,
PLCCFG := PLCCFG, ACTBUF := ACTBUF);
(*значення за замовченням*)
IF PLCCFG.STA1.SCN1 THEN (*first scan*)
  IF ACTCFGu.T_OPNSP <= 0 THEN (*якщо уставка часу відкриття не виставлена*)
    ACTCFGu.T_OPNSP := 500; (*5 секунд*)
  END_IF;
  IF ACTCFGu.T_DEASP <= 0 THEN (*якщо уставка затримки часу тривоги не виставлена*)
    ACTCFGu.T_DEASP := 200; (*2 секунди*)
  END_IF;
  (*технологічні тривоги для датчиків не використовуються*)
  IF SOPN.ID <> 0 THEN
    SOPN.PRM1.PRM_ISALM := false; (*ISALM*)
    SOPN.PRM1.PRM_ISWRN := false; (*ISWRN*)
  END_IF;
  IF SCLS.ID <> 0 THEN
    SCLS.PRM1.PRM_ISALM := false; (*ISALM*)
    SCLS.PRM1.PRM_ISWRN := false; (*ISWRN*)
  END_IF;
END_IF;
(* ----- блок параметрів
//параметри перевірка наявності/використання датчиків на вході*)
ACTCFGu.PRM1.PRM1.PRM_MANCFG:=false;(*у цьому проекті не буде ручного конфігурування
параметрів IO*)
ACTCFGu.PRM1.PRM1.PRM_ZCLSENBL := NOT SCLS.PRM1.PRM_DSBL AND SCLS.ID <> 0;
ACTCFGu.PRM1.PRM1.PRM_ZOPNENBL := NOT SOPN.PRM1.PRM_DSBL AND SOPN.ID <> 0;
ACTCFGu.PRM1.PRM1.PRM_ALMENBL := false;
ACTCFGu.PRM1.PRM1.PRM_ZWRKENBL := false;
ACTCFGu.PRM1.PRM1.PRM_ZPOSENBL := NOT POS.PRM1.PRM_DSBL AND POS.ID <> 0;
ACTCFGu.PRM1.PRM1.PRM_PWRENBL := false;
ACTCFGu.PRM1.PRM1.PRM_BTNSTPENBL := false;
ACTCFGu.PRM1.PRM1.PRM_ALMENBL := false;
ACTCFGu.PRM1.PRM1.PRM_SELLCLENBL := false;
(*параметри імпульсного керування*)
ACTCFGu.PRM1.PRM1.PRM_PULSCTRLENBL := FALSE;
(*----- блок для режиму імітації
//режим імітації підлеглих від хазяїна*)
SOPN.STA1.STA_SML := STA1.STA1.STA_SML;
SCLS.STA1.STA_SML := STA1.STA1.STA_SML;
(*логіка для режиму імітації *)
IF STA1.STA1.STA_SML THEN
  ACTCFGu.POS:=ACTCFGu.CPOS;
  IF ACTCFGu.POS<0.0 THEN ACTCFGu.POS:=0.0; END_IF;
  IF ACTCFGu.POS>100.0 THEN ACTCFGu.POS:=100.0; END_IF;
  (*імітація датчиків *)
  IF NOT SOPN.STA1.STA_FRC THEN
    SOPN.STA1.STA_VALB:= ACTCFGu.POS>99.0;
  END_IF;
  IF NOT SCLS.STA1.STA_FRC THEN
    SCLS.STA1.STA_VALB:= ACTCFGu.POS<3.0;
  END_IF;
END_IF;
END_IF;
(*----- блок обробки команд
//стандартний обробник команд*)
```

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Лист | № докум. | Підпис | Дата |

ТО8151.0014.001.АТХ.П

Арк

85

```

ACT_CMDCTRL1 (ACTCFG:=ACTCFGu, CMD:=CMD, STA1:=STA1, ACTBUF := ACTBUF);
(* ----- блок обробки станів датчиків відкриття/закриття, або їх
заміна на логіку*)
IF NOT ACTCFGu.PRM1.PRM1.PRM_ZPOSENBL THEN      (*// ЯКЩО НЕМАЄ СИГНАЛ ЗВОРОТНЬОГО
ЗВ'ЯЗКУ ПОЛОЖЕННЯ*)
    SPOS1:=CPOS.VAL;                            (* // ТО ПОЛОЖЕННЯ ПРИРІВНЮЄМО ЗАДАНОМУ
ЗНАЧЕННЮ*)
ELSE
    SPOS1:=POS.VAL;                             (* // ІНАКШЕ БЕРЕЗ ЗНАЧЕННЯ З ДАТЧИКА*)
END_IF;
IF NOT ACTCFGu.PRM1.PRM1.PRM_ZOPNENBL THEN      (*// ЯКЩО НЕМАЄ КІНЦЕВИКА ВІДКРИТО*)
    SOPN1:= SPOS1>=3.0;                         (* // ТО ЗНАЧЕННЯ ВІДКРИТОСТІ БЕРЕМ ЯКЩО
ПОЗИЦІЯ ВІЛЬШЕ 3*)
ELSE
    SOPN1:=SOPN.STA1.STA_VALB;                 (* // ІНАКШЕ БЕРЕЗ ЗНАЧЕННЯ З
ДАТЧИКА*)
END_IF;
IF NOT ACTCFGu.PRM1.PRM1.PRM_ZCLSENBL THEN      (* // ЯКЩО НЕМАЄ КІНЦЕВИКА ЗАКРИТО*)
    SCLS1:=SPOS1<3.0;                          (* // ТО ЗНАЧЕННЯ ВІДКРИТОСТІ БЕРЕМ ЯКЩО
ПОЗИЦІЯ МЕНШЕ 3*)
ELSE
    SCLS1:=SCLS.STA1.STA_VALB;                 (* // ІНАКШЕ БЕРЕЗ ЗНАЧЕННЯ З
ДАТЧИКА*)
END_IF;
(*----- автомат станів позиції та тривоги позиції *)
CASE ACTCFGu.STEP1 OF
    0: (*ініціалізація*)
        ACTCFGu.STEP1 := 1;
        ACTCFGu.T_STEP1 := 0;
    1, 4, 5: (*зупинений, 1 - зупинено в проміжному стані 4 - зупинений у відкритому
стані, 5 - зупинений в закритому стані*)
        IF SOPN1 OR NOT SCLS1 THEN
            ACTCFGu.STEP1 := 4;
            ACTCFGu.T_STEP1 := 0;
        END_IF;
        IF SCLS1 AND NOT SOPN1 THEN
            ACTCFGu.STEP1 := 5;
            ACTCFGu.T_STEP1 := 0;
        END_IF;
    ELSE (*невизначеність*)
        ACTCFGu.STEP1 := 0;
END_CASE;
(*автомат станів побітово*)
STA1.STA1.STA_IMSPD:= ACTCFGu.STEP1 = 0 OR ACTCFGu.STEP1 = 1;
STA1.STA1.STA_OPND:= ACTCFGu.STEP1 = 4;
STA1.STA1.STA_CLSD:= ACTCFGu.STEP1 = 5;
(*----- керування ВМ*)
(*керування OPN/CLS тільки при дозволі керування або тимчасовому розблокуванні*)
IF (CMD.CMD2.CMD_RESOLUTION OR PLCCFG.STA_PERM.%X6) AND NOT STA1.STA1.STA_BLK THEN
;
ELSE
    ACTCFG.CPOS:=0.0;
    ACTHMI.CPOS:=0.0;
END_IF;
(*----- режими*)
STA1.STA1.STA_FRC := CPOS.STA1.STA_FRC AND CPOS.ID<>0
OR POS.STA1.STA_FRC AND POS.ID<>0
OR SOPN.STA1.STA_FRC AND SOPN.ID<>0
OR SCLS.STA1.STA_FRC AND SCLS.ID<>0;
(*----- зведення ксатомних тривоги, режимів, бітів*)
ALMs.ALM_1.ALM_ALM := false;
(*----- вибір джерела завдання*)
IF ACTCFGu.STA1.STA1.STA_INBUF AND ACTCFGu.STA1.STA1.STA_DISP THEN

```

```

ACTCFGU.CPOS := ACTBUF.CPOS;
ELSIF NOT ACTCFGU.STA1.STA1.STA_INBUF AND ACTCFGU.STA1.STA1.STA_DISP THEN
  ACTCFGU.CPOS := ACTHMI.CPOS;
ELSE
  ACTHMI.CPOS := ACTCFGU.CPOS;
END_IF;

ACTHMI.CPOS := ACTCFGU.CPOS;
ACTCFG.CPOS := ACTCFGU.CPOS;
ACTCFGU.POS := SPOS1;
CPOS.VAL:=ACTCFGU.CPOS;

(*заключна обробка: зведення в PLC.CFG, STA, ALM, CMD, INBUF, SML, dt *)
ACT_POST1(ACTCFG:=ACTCFGU, STA1:=STA1, ALMs:=ALMs, CMD:=CMD, dt1 := dt1, ACTBUF :=
ACTBUF, PLCCFG := PLCCFG);
ACT_to_VLVA1(VLVACFG:=ACTCFG, VLVAHMI:=ACTHMI, ACTCFG:=ACTCFGU);

```

DRVFN

Реалізація DRVFN:

```

DRV_to_ACT1(DRVCFG:=ACTCFG, DRVHMI:=ACTHMI, ACTCFG:=ACTCFGU);
(*попередня обробка: ініт STA, ALM, CMD, INBUF, SML, dt *)
ACT_PRE1(ACTCFG := ACTCFGU, STA1 := STA1, ALMs := ALMs, CMD := CMD, dt1 := dt1,
PLCCFG := PLCCFG, ACTBUF := ACTBUF);
(*значення за замовченням*)
IF PLCCFG.STA1.SCN1 THEN (*first scan*)
  IF ACTCFGU.T_OPNSP <= 0 THEN (*якщо уставка часу відкриття не виставлена*)
    ACTCFGU.T_OPNSP := 500; (*5 секунд*)
  END_IF;
  IF ACTCFGU.T_DEASP <= 0 THEN (*якщо уставка затримки часу тривоги не виставлена*)
    ACTCFGU.T_DEASP := 200; (*2 секунди*)
  END_IF;
  (*технологічні тривоги для датчиків не використовуються*)
  IF RUN.ID <> 0 THEN
    RUN.PRM1.PRM_ISALM := false; (*ISALM*)
    RUN.PRM1.PRM_ISWRN := false; (*ISWRN*)
  END_IF;
END_IF;
(* ----- блок параметрів*)
(*параметри перевірка наявності/використання датчиків на вході*)
ACTCFGU.PRM1.PRM1.PRM_MANCFG := false; (*у цьому проекті не буде ручного
конфігурування параметрів IO*)
ACTCFGU.PRM1.PRM1.PRM_ALMENBL := NOT ALM.PRM1.PRM_DSBL AND ALM.ID <> 0;
ACTCFGU.PRM1.PRM1.PRM_ZWRKENBL := NOT RUN.PRM1.PRM_DSBL AND RUN.ID <> 0;
ACTCFGU.PRM1.PRM1.PRM_ZPOSENBL := NOT SPD.PRM1.PRM_DSBL AND SPD.ID <> 0;
ACTCFGU.PRM1.PRM1.PRM_PWRENBL := NOT PWR.PRM1.PRM_DSBL AND PWR.ID <> 0;
ACTCFGU.PRM1.PRM1.PRM_BTNSTPENBL := NOT LSTP.PRM1.PRM_DSBL AND LSTP.ID <> 0;
ACTCFGU.PRM1.PRM1.PRM_ALMENBL := NOT ALM.PRM1.PRM_DSBL AND ALM.ID <> 0;
ACTCFGU.PRM1.PRM1.PRM_SELLCLENBL := NOT RMT.PRM1.PRM_DSBL AND RMT.ID <> 0;
(* ----- блок для режиму імітації*)
(*режим імітації підлеглих від хозяїна*)
LSTP.STA1.STA_SML := STA1.STA1.STA_SML;
RUN.STA1.STA_SML := STA1.STA1.STA_SML;
PWR.STA1.STA_SML := STA1.STA1.STA_SML;
ALM.STA1.STA_SML := STA1.STA1.STA_SML;
RMT.STA1.STA_SML := STA1.STA1.STA_SML;
CSTRT.STA1.STA_SML := STA1.STA1.STA_SML;
SPD.STA1.STA_SML := STA1.STA1.STA_SML;
CSPD.STA1.STA_SML := STA1.STA1.STA_SML;
(*логіка для режиму імітації *)
IF STA1.STA1.STA_SML THEN

```

```

(*імітація датчиків *)
IF NOT SPD.STA1.STA_FRC THEN SPD.VAL := CSPD.VAL; END_IF;
IF NOT RUN.STA1.STA_FRC THEN RUN.STA1.STA_VALB:=CSTRT.STA1.STA_VALB AND
CSTRT.T_STEP1>=500; END_IF;
IF NOT PWR.STA1.STA_FRC THEN PWR.STA1.STA_VALB:=true; END_IF; (*живлення е*)
IF NOT ALM.STA1.STA_FRC THEN ALM.STA1.STA_VALB := false; END_IF; (*тривоги немає*)
END_IF;
(*----- блок обробки команд
//стандартний обробник команд*)
ACT_CMDCTRL1(ACTCFG := ACTCFGu, STA1 := STA1, CMD := CMD, ACTBUF := ACTBUF);
(*переключення в ручний режим при вклученні локального ручного і блокування усіх
команд*)
IF RMT.STA1.STA_VALB AND NOT RMT.PRM1.PRM_DSBL AND ACTCFGu.PRM1.PRM1.PRM_SELLCLENBL
THEN
    STA1.STA1.STA_DISP := TRUE;
    (*CMD_RESET*)
    ACTCFGu.CMD.CMD1.CMD_OPN:=FALSE;
    ACTCFGu.CMD.CMD1.CMD_CLS:=FALSE;
    ACTCFGu.CMD.CMD1.CMD_TOGGLE:=FALSE;
    ACTCFGu.CMD.CMD1.CMD_ALMACK:=FALSE;
    ACTCFGu.CMD.CMD1.CMD_ALMRESET:=FALSE;
    ACTCFGu.CMD.CMD1.CMD_BLCK:=FALSE;
    ACTCFGu.CMD.CMD1.CMD_DBLCK:=FALSE;
    ACTCFGu.CMD.CMD1.CMD_STOPTUN:=FALSE;
    ACTCFGu.CMD.CMD1.CMD_TUNING:=FALSE;
    ACTCFGu.CMD.CMD1.CMD_MAN:=FALSE;
    ACTCFGu.CMD.CMD1.CMD_AUTO:=FALSE;
    ACTCFGu.CMD.CMD1.CMD_PROTECT:=FALSE;
    ACTCFGu.CMD.CMD1.CMD_START:=FALSE;
    ACTCFGu.CMD.CMD1.CMD_STOP:=FALSE;
    ACTCFGu.CMD.CMD1.CMD_UP:=FALSE;
    ACTCFGu.CMD.CMD1.CMD_DWN:=FALSE;
    ACTCFGu.CMD.CMD2.CMD_CRMT:=FALSE;
    ACTCFGu.CMD.CMD2.CMD_RESOLUTION:=FALSE;
    ACTCFGu.CMD.CMD2.CMD_REVERS:=FALSE;
    ACTCFGu.CMD.CMD2.CMD_CLCL:=FALSE;
    ACTCFGu.CMD.CMD2.CMD_DBLCKACTTOGGLE:=FALSE;
    ACTCFGu.CMD.CMD2.CMD_STARTDELAY:=FALSE;
    ACTCFGu.CMD.CMD2.CMD_STOPDELAY:=FALSE;
    ACTCFGu.CMD.CMD2.CMD_P_RESOLUTION:=FALSE;
    ACTCFGu.CMD.CMD2.CMD_BUFLOAD:=FALSE;
    ACTCFGu.CMD.CMD2.CMD_b25:=FALSE;
    ACTCFGu.CMD.CMD2.CMD_b26:=FALSE;
    ACTCFGu.CMD.CMD2.CMD_b27:=FALSE;
    ACTCFGu.CMD.CMD2.CMD_b28:=FALSE;
    ACTCFGu.CMD.CMD2.CMD_b29:=FALSE;
    ACTCFGu.CMD.CMD2.CMD_b30:=FALSE;
    ACTCFGu.CMD.CMD2.CMD_b31:=FALSE;
END_IF;
(* ----- блок обробки станів датчиків відкриття/закриття, або їх
заміна на логіку*)
IF NOT ACTCFGu.PRM1.PRM1.PRM_ZWRKENBL THEN
    SRUN1:=CSTRT.STA1.STA_VALB;
ELSE
    SRUN1:=RUN.STA1.STA_VALB;
END_IF;
IF NOT ACTCFGu.PRM1.PRM1.PRM_PWRENBL THEN
    SPWR1:=true;
ELSE
    SPWR1:=PWR.STA1.STA_VALB;
END_IF;
IF NOT ACTCFGu.PRM1.PRM1.PRM_ALMENBL THEN
    SALM1:=FALSE;

```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 88 |

```

ELSE
    SALM1:=ALM.STA1.STA_VALB;
END_IF;
IF NOT АСТСFGu.PRM1.PRM1.PRM_ZPOSENBL THEN      (* ЯКЩО НЕМА СИГНАЛ ЗВОРОТНЬОГО
ЗВ'ЯЗКУ ПОЛОЖЕННЯ*)
    SPOS1:=CSPD.VAL;                            (* ТО ПОЛОЖЕННЯ ПРИРІВНЮЄМО ЗАДАНОМУ
ЗНАЧЕННЮ*)
ELSE
    SPOS1:=SPD.VAL;                             (* ІНАКШЕ БЕРЕЗ ЗНАЧЕННЯ З ДАТЧИКА*)
END_IF;
(*----- автомат станів позиції та тривог позиції *)
CASE АСТСFGu.STEP1 OF
    0: (*ініціалізація*)
        АСТСFGu.STEP1 := 1;
        АСТСFGu.T_STEP1 := 0;
    1, 4, 5: (*кінцеві стани, 1 - невизначений 4 - запущений, 5 - зупинений*)
        IF ((АСТСFGu.STEP1 = 4 AND NOT SRUN1) OR (АСТСFGu.STEP1 = 5 AND SRUN1)) AND
NOT STA1.STA1.STA_MANBX THEN
            ALMS.ALM_1.ALM_ALMSHFT:=true;
        ELSE
            ALMS.ALM_1.ALM_ALMSHFT:=FALSE;
        END_IF;

        IF SRUN1 AND NOT ALMS.ALM_1.ALM_ALMSHFT THEN
            АСТСFGu.STEP1 := 4;
            CSTRT.STA1.STA_VALB:=true;
        END_IF;
        IF NOT SRUN1 AND NOT ALMS.ALM_1.ALM_ALMSHFT THEN
            АСТСFGu.STEP1 := 5;
            CSTRT.STA1.STA_VALB:=FALSE;
        END_IF;
    2: (*запускається*)
        CSTRT.STA1.STA_VALB:=true;
        IF SRUN1 THEN
            АСТСFGu.STEP1 := 4; (*у стан запущено*)
            АСТСFGu.T_STEP1 := 0;
        END_IF;

        ALMS.ALM_1.ALM_ALMSTRT := FALSE;
        ALMS.ALM_1.ALM_ALMSTP := FALSE;

        IF АСТСFGu.T_STEP1 >= (UINT_TO_UDINT(АСТСFGu.T_DEASP)*100) THEN
            ALMS.ALM_1.ALM_ALMSTRT := TRUE;
            ALMS.ALM_1.ALM_ALMSTP := FALSE;
            АСТСFGu.STEP1 := 6; (*у стан заблоковано*)
            АСТСFGu.T_STEP1 := 0;
        END_IF;
    3: (*зупиняється*)
        CSTRT.STA1.STA_VALB:=FALSE;
        IF NOT SRUN1 THEN
            АСТСFGu.STEP1 := 5;
            АСТСFGu.T_STEP1 := 0;
        END_IF;

        ALMS.ALM_1.ALM_ALMCLS := false;
        ALMS.ALM_1.ALM_ALMOPN := FALSE;

        IF АСТСFGu.T_STEP1 >= (UINT_TO_UDINT(АСТСFGu.T_DEASP)*100) THEN
            ALMS.ALM_1.ALM_ALMSTP := TRUE;
            ALMS.ALM_1.ALM_ALMSTRT := FALSE;
            АСТСFGu.STEP1 := 6; (*у стан заблоковано*)
            АСТСFGu.T_STEP1 := 0;
        END_IF;

```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 89 |

```

6: (*заблоковано*)
  CSTRT.STA1.STA_VALB:=FALSE;
  IF CMD.CMD1.CMD_DBLCK THEN
    ACTCFGU.STEP1 := 5; (*у стан зупинено*)
    ACTCFGU.T_STEP1 := 0;
    ALMS.ALM_1.ALM_ALMSTRT:=FALSE;
    ALMS.ALM_1.ALM_ALMSTP:=FALSE;
    ALMS.ALM_2.ALM_ALMINVRTR:=FALSE;
    ALMS.ALM_1.ALM_ALMPWR:=FALSE;
    ALMS.ALM_1.ALM_ALMSHFT:=FALSE;
  END_IF;
ELSE (*невизначеність*)
  ACTCFGU.STEP1 := 0;
END_CASE;

STA1.STA2.STA_ISANALOG:=CSPD.ID<>0;
STA1.STA1.STA_STOPING :=ACTCFGU.STEP1 = 3;
STA1.STA1.STA_STRTING :=ACTCFGU.STEP1 = 2;
STA1.STA2.STA_STOPEd :=ACTCFGU.STEP1 = 5 OR ACTCFGU.STEP1 = 6;
STA1.STA1.STA_WRKED :=ACTCFGU.STEP1 = 4;
STA1.STA1.STA_BLCK :=ACTCFGU.STEP1 = 6;
(*----- керування ВМ*)
IF CMD.CMD1.CMD_STOP THEN
  CSTRT.STA1.STA_VALB:=FALSE;
END_IF;
(*керування OPN/CLS тільки при дозволі керування або тимчасовому розблокуванні*)
IF CMD.CMD2.CMD_RESOLUTION OR PLCCFG.STA_PERM.%X6 THEN
  IF CMD.CMD1.CMD_START AND ACTCFGU.STEP1 <> 4 THEN
    ACTCFGU.STEP1 := 2;
    ACTCFGU.T_STEP1 := 0;
    ACTCFGU.CNTPER := ACTCFGU.CNTPER + 1;
  END_IF;
  IF CMD.CMD1.CMD_STOP AND ACTCFGU.STEP1 <> 5 THEN
    ACTCFGU.STEP1 := 3;
    ACTCFGU.T_STEP1 := 0;
  END_IF;
ELSE
  CSTRT.STA1.STA_VALB:=FALSE;
END_IF;
(*----- режими*)
STA1.STA1.STA_FRC :=LSTP.STA1.STA_FRC AND LSTP.ID<>0
OR RUN.STA1.STA_FRC AND RUN.ID<>0
OR PWR.STA1.STA_FRC AND PWR.ID<>0
OR ALM.STA1.STA_FRC AND ALM.ID<>0
OR RMT.STA1.STA_FRC AND RMT.ID<>0
OR CSTRT.STA1.STA_FRC AND CSTRT.ID<>0
OR SPD.STA1.STA_FRC AND SPD.ID<>0
OR CSPD.STA1.STA_FRC AND CSPD.ID<>0;

ALMS.ALM_1.ALM_ALMPWR:=NOT SPWR1;
ALMS.ALM_1.ALM_ALM := ALMS.ALM_1.ALM_ALMSTRT OR ALMS.ALM_1.ALM_ALMSTP OR
ALMS.ALM_2.ALM_ALMINVRTR OR ALMS.ALM_1.ALM_ALMPWR OR ALMS.ALM_1.ALM_ALMSHFT;
IF ALMS.ALM_1.ALM_ALM AND NOT STA1.STA1.STA_MANBXOUT AND ACTCFGU.STEP1<>6 THEN
  ACTCFGU.STEP1 := 6;
  ACTCFGU.T_STEP1 := 0;
END_IF;
(*----- зведення ксатомних тривог, режимів, бітів*)
IF ALMS.ALM_1.ALM_ALMSTRT THEN
  PLCCFG.ALM.ALM := TRUE;
  IF NOT ACTCFGU.ALM.ALM_1.ALM_ALMSTRT THEN
    PLCCFG.ALM.NWALM := TRUE;
    ACTCFGU.CNTALM := ACTCFGU.CNTALM + 1;
  END_IF;

```

```

END_IF;

IF ALMS.ALM_1.ALM_ALMSTP THEN
  PLCCFG.ALM.ALM := TRUE;
  IF NOT ACTCFGU.ALM.ALM_1.ALM_ALMSTP THEN
    PLCCFG.ALM.NWALM := TRUE;
    ACTCFGU.CNTALM := ACTCFGU.CNTALM + 1;
  END_IF;
END_IF;

IF ALMS.ALM_1.ALM_ALMSHFT THEN
  PLCCFG.ALM.ALM := TRUE;
  IF NOT ACTCFGU.ALM.ALM_1.ALM_ALMSHFT THEN
    PLCCFG.ALM.NWALM := TRUE;
    ACTCFGU.CNTALM := ACTCFGU.CNTALM + 1;
  END_IF;
END_IF;

IF ALMS.ALM_2.ALM_ALMINVRTR THEN
  PLCCFG.ALM.ALM := TRUE;
  IF NOT ACTCFGU.ALM.ALM_2.ALM_ALMINVRTR THEN
    PLCCFG.ALM.NWALM := TRUE;
    ACTCFGU.CNTALM := ACTCFGU.CNTALM + 1;
  END_IF;
END_IF;

IF ALMS.ALM_1.ALM_ALMPWR THEN
  PLCCFG.ALM.ALM := TRUE;
  IF NOT ACTCFGU.ALM.ALM_1.ALM_ALMPWR THEN
    PLCCFG.ALM.NWALM := TRUE;
    ACTCFGU.CNTALM := ACTCFGU.CNTALM + 1;
  END_IF;
END_IF;

IF ACTCFGU.STA1.STA1.STA_WRKED THEN
  ACTCFGU.TQ_LAST:=ACTCFGU.T_STEP1/60000;
END_IF;

IF ACTCFGU.TQ_LAST>=16#7FFFFFFF THEN
  ACTCFGU.TQ_LAST:=16#7FFFFFFF;
END_IF;

IF ACTCFGU.STA1.STA1.STA_WRKED THEN
  IF PLCCFG.PLS1.P60S THEN
    ACTCFGU.TQ_TOTAL:=ACTCFGU.TQ_TOTAL+1;
  END_IF;
END_IF;

IF ACTCFGU.TQ_TOTAL>=16#7FFFFFFF THEN
  ACTCFGU.TQ_TOTAL:=16#7FFFFFFF;
END_IF;

IF ACTCFGU.STA1.STA1.STA_INBUF AND ACTCFGU.STA1.STA1.STA_DISP THEN
  ACTCFGU.CPOS := ACTBUF.CPOS;
ELSEIF NOT ACTCFGU.STA1.STA1.STA_INBUF AND ACTCFGU.STA1.STA1.STA_DISP THEN
  ACTCFGU.CPOS := ACTHMI.CSPD;
ELSE
  ACTHMI.CSPD := ACTCFGU.CPOS;
END_IF;

ACTHMI.CSPD := ACTCFGU.CPOS;
ACTCFG.CSPD := ACTCFGU.CPOS;
ACTCFGU.POS := SPOS1;
CSPD.VAL:=ACTCFGU.CPOS;
(*заклучна обробка: зведення в PLC.CFG, STA, ALM, CMD, INBUF, SML, dt *)

```


The screenshot shows the UaExpert interface. On the left, the 'Project' tree shows a server named 'eUAServer@axc-f-1152-1'. Below it, the 'Address Space' tree shows a hierarchy: NetworkSet > PLCnext > Arp.Plc.Eclr > Main1 > AIH1 > T1.LT1. The 'T1.LT1' node is selected. On the right, the 'Data Access View' table displays the following data:

| # | Server | Node Id | Display Name | Value | Datatype | Source Timestamp | Server Timestamp | Statuscode |
|----|------------------|--|--------------|-------|----------|------------------|------------------|------------|
| 1 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.VAL | VAL | 0 | Float | 12:14:27.817 | 12:14:27.817 | Good |
| 2 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.VALPRCSTA2 | VALPRCSTA2 | 0 | UInt16 | 12:14:27.818 | 12:14:27.818 | Good |
| 3 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.STA_ALDIS | STA_ALDIS | false | Boolean | 12:14:27.818 | 12:14:27.818 | Good |
| 4 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.STA1.STA_ALM | STA_ALM | false | Boolean | 12:14:27.818 | 12:14:27.818 | Good |
| 5 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.STA1.STA_BAD | STA_BAD | false | Boolean | 12:14:27.818 | 12:14:27.818 | Good |
| 6 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.STA1.STA_BRK | STA_BRK | false | Boolean | 12:14:27.819 | 12:14:27.819 | Good |
| 7 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.STA1.STA_CMDLOAD | STA_CMDLOAD | false | Boolean | 12:14:27.819 | 12:14:27.819 | Good |
| 8 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.STA1.STA_DLNK | STA_DLNK | true | Boolean | 12:14:27.819 | 12:14:27.819 | Good |
| 9 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.STA1.STA_ENBL | STA_ENBL | true | Boolean | 12:14:27.819 | 12:14:27.819 | Good |
| 10 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.STA1.STA_FRC | STA_FRC | false | Boolean | 12:14:27.820 | 12:14:27.820 | Good |
| 11 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.STA1.STA_HI | STA_HI | false | Boolean | 12:14:27.820 | 12:14:27.820 | Good |
| 12 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.STA1.STA_HIHI | STA_HIHI | false | Boolean | 12:14:27.820 | 12:14:27.820 | Good |
| 13 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.STA1.STA_INBUF | STA_INBUF | false | Boolean | 12:14:27.820 | 12:14:27.820 | Good |
| 14 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.STA1.STA_LO | STA_LO | false | Boolean | 12:14:27.821 | 12:14:27.821 | Good |
| 15 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.STA1.STA_LOLO | STA_LOLO | false | Boolean | 12:14:27.821 | 12:14:27.821 | Good |
| 16 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.STA1.STA_OVRLD | STA_OVRLD | false | Boolean | 12:14:27.821 | 12:14:27.821 | Good |
| 17 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.STA1.STA_SML | STA_SML | true | Boolean | 12:14:27.821 | 12:14:27.821 | Good |
| 18 | eUAServer@axc... | NSS[String]Arp.Plc.Eclr/Main1.AI11.T1.LT1.STA1.STA_WRN | STA_WRN | false | Boolean | 12:14:27.822 | 12:14:27.822 | Good |

Рисунок 3.7 - Перевірка відправлених даних через UaExpert

3) Бібліотекою для Node-Red - node-red-contrib-opcsua, передбачено пересилання даних через OPC UA. Створимо ноду OpсUa-Client та в її налаштуваннях виставимо дію BROWSE (Рис. 3.8). Дія BROWSE видає перелік усіх вузлів даних починаючи з кореневого, що заданий в Topic вхідного повідомлення. Для кожного із вузлів було створено ноду Change (Рисунки 3.9, 3.10) для витягування даних з конкретного вузла, вказавши в Topic номер елемента (значенн NodeId з утиліти UaExpert).

Изменить узел OpсUa-Client

Удалить Отмена **Готово**

Свойства

Endpoint

Action

Certificate

Local certificate file with absolute path

Local private key file with absolute path

PKI certificate folder

Name

Рисунок 3.8 - Налаштування ноди OpсUa-Client

Изменить узел change

Удалить Отмена **Готово**

Свойства

Имя

Правила

Установить to the value

Установить to the value

Установить to the value

Deep copy value

Рисунок 3.9 - Налаштування ноди Change для даних PLC

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Лист | № докум. | Підпис | Дата |

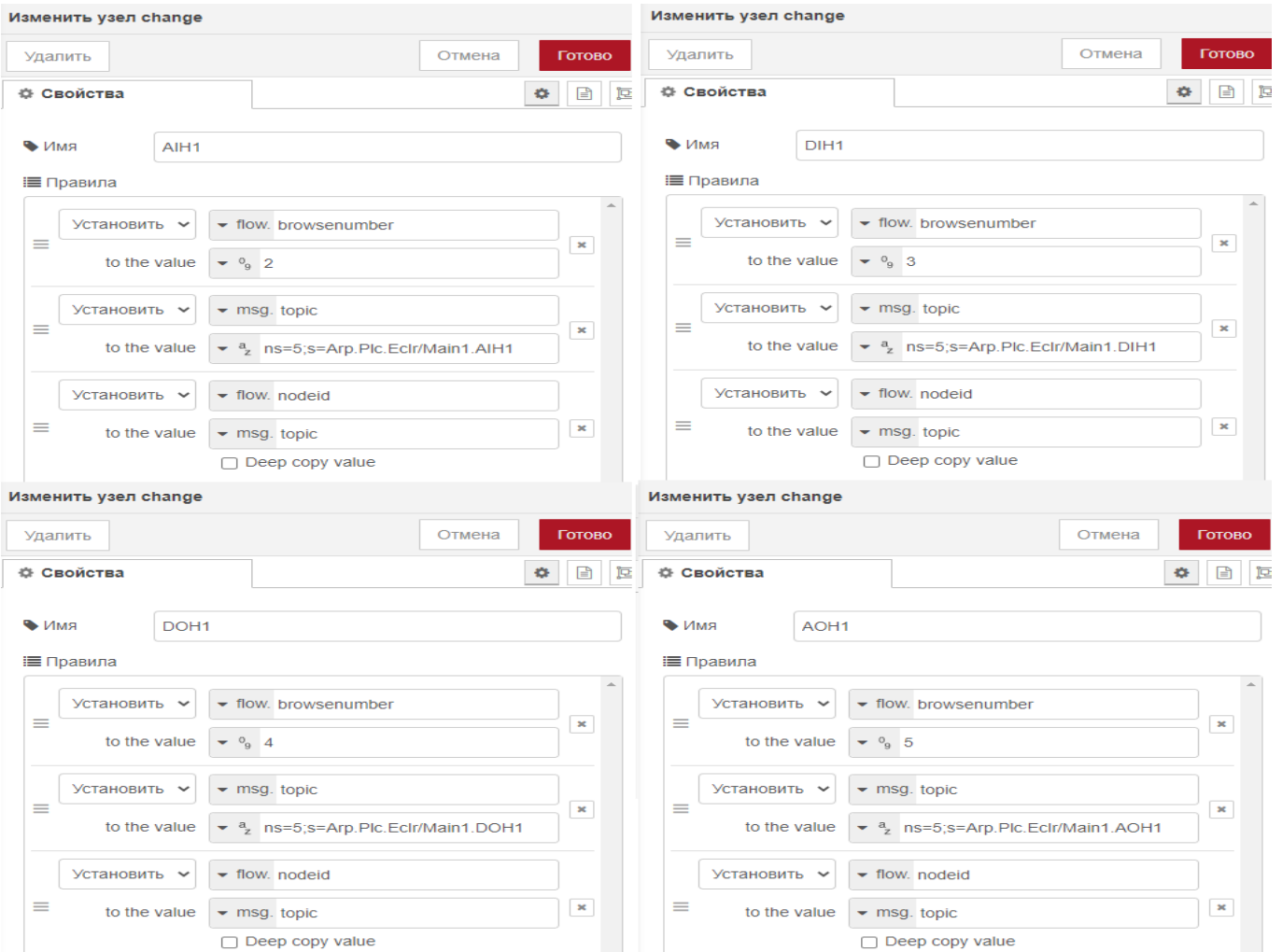


Рисунок 3.10 - Налаштування ноди Change для даних AIH1, AOH1, DIH1, DOH1

4. ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ І АНАЛІЗ ФУНКЦІОНУВАННЯ ПЛК

У даному розділі описані програмні частини реалізації демонстраційної установки для перевірки роботи розробленої бібліотеки PACFramework для PLCnext.

Для перевірки було зроблено наступні дії:

- розроблено імітаційну модель об'єкту з використанням ресурсів ПЛК
- розроблено людино-машинний інтерфейс для імітаційного об'єкту
- розгорнуто каркас для даного об'єкту керування
- перевірено роботу каркасу з використанням людино-машинного інтерфейсу налагодження

4.1. Розроблення імітаційної моделі

Для перевірки роботи каркасу на програмній платформі PLCnext Engineer було розроблено імітаційну модель рецептурно-змішувального комплексу для приготування продукції з дозуванням за рівнем компонентів у дозаторі та періодичним змішуванням компонентів (Рис. 4.1) відповідно до вимог наведених в п. 1.4.

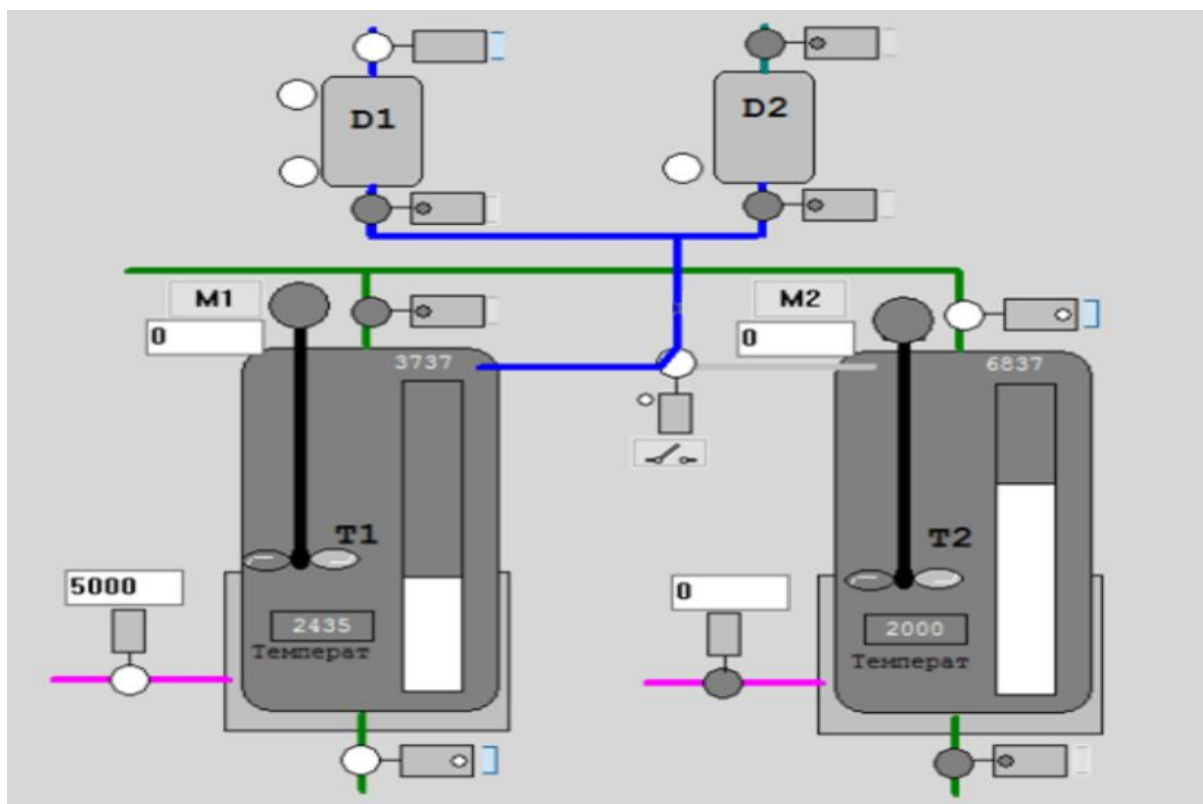


Рисунок 4.1 - Зображення імітованої установки

Для кожного з об'єктів було створено імітаційну модель безпосередньо в програмі ПЛК.

Для імітаційної моделі танку був створений функціональний блок smTankT (рис.4.2)

Танки

| Name | Type | Usage | Translate | Comment | Init |
|----------------|------|--------|--------------------------|---|-------------|
| <i>default</i> | | | | | |
| INIT | BOOL | Input | <input type="checkbox"/> | команда ініціалізації (задання початкових умов) | FALSE |
| Fin | REAL | Input | <input type="checkbox"/> | вхідна витрата рідини, м3/с | REAL#0.0 |
| Fout | REAL | Input | <input type="checkbox"/> | вихідна витрата рідини, м3/с | REAL#0.0 |
| Fa | REAL | Input | <input type="checkbox"/> | витрата теплоагента, м3/с | REAL#0.0 |
| Tin | REAL | Input | <input type="checkbox"/> | вхідна температура, град С | REAL#0.0 |
| Tain | REAL | Input | <input type="checkbox"/> | температура теплоагента, град С | REAL#0.0 |
| L | REAL | Output | <input type="checkbox"/> | рівень рідини в ємності, м | REAL#0.0 |
| V | REAL | Output | <input type="checkbox"/> | об'єм рідини в ємності, м3 | REAL#0.0 |
| T | REAL | Output | <input type="checkbox"/> | температура рідини в ємності, град С | REAL#0.0 |
| Ta | REAL | Output | <input type="checkbox"/> | температура теплоагента в кожусі, град С | REAL#0.0 |
| K1 | REAL | Local | <input type="checkbox"/> | коефіцієнт | REAL#0.0 |
| K2 | REAL | Local | <input type="checkbox"/> | коефіцієнт | REAL#0.0 |
| Vold | REAL | Local | <input type="checkbox"/> | об'єм на попередньому кроці, м3 | REAL#0.0 |
| S | REAL | Local | <input type="checkbox"/> | ефективна поверхня теплообміну кожуха з ємністю, м2/с | REAL#0.0 |
| d_t | REAL | Input | <input type="checkbox"/> | періодичність виклику, с | REAL#0.1 |
| Vmax | REAL | Local | <input type="checkbox"/> | об'єм ємності, м3 | REAL#10.0 |
| Sv | REAL | Local | <input type="checkbox"/> | поперечний переріз ємності, м2 | REAL#1.0 |
| V0 | REAL | Local | <input type="checkbox"/> | початк. об'єм рідини в ємності, м3 | REAL#0.0 |
| T0 | REAL | Local | <input type="checkbox"/> | початк. температура рідини в ємності, м3 | REAL#20.0 |
| Ta0 | REAL | Local | <input type="checkbox"/> | початк. температура теплоносія в кожусі, град С | REAL#20.0 |
| Sa | REAL | Local | <input type="checkbox"/> | повна поверхня теплообміну кожуха, м2 | REAL#10.0 |
| C | REAL | Local | <input type="checkbox"/> | теплоємність рідини, кДж/(кг*К) | REAL#4.19 |
| Ca | REAL | Local | <input type="checkbox"/> | теплоємність теплоносія, кДж/(кг*К) | REAL#0.74 |
| k | REAL | Local | <input type="checkbox"/> | коефіцієнт теплопередачі, кВт/(м2*градС) | REAL#2.0 |
| ro | REAL | Local | <input type="checkbox"/> | густина рідини, кг*м3 | REAL#1000.0 |
| roa | REAL | Local | <input type="checkbox"/> | густина теплоносія, кг*м3 | REAL#1000.0 |
| Va | REAL | Local | <input type="checkbox"/> | об'єм кожуха, м3 | REAL#1.0 |
| La | REAL | Local | <input type="checkbox"/> | висота кожуха, м | REAL#1.0 |

Рисунок 4.2 - Змінні функціонального блоку smTankT

Код реалізації танку:

```

IF INIT THEN
  V := V0;
  T := T0;
  Ta := Ta0;
END_IF;

(*-----розрахунок об'єму*)
Vold := V;
V := V + d_t * (Fin - Fout);

IF V < 0.0 THEN
  V := 0.0;
END_IF;

IF V > Vmax THEN
  V := Vmax;
END_IF;

L := V / Sv;

(*-----розрахунок температур*)
IF L > La THEN
  S := (L / La) * Sa;
ELSE
  S := Sa;
END_IF;

K1 := k * S / (ro * C);

```

```

K2 := k * S / (roa * Ca);

IF V > Vmax / 1000.0 THEN (*об'єм відмінний від нуля*)
  IF V < Vmax THEN (*ємність не повна*)
    T := T * (Vold / V) + (d_t / V) * (Fin * Tin + K1 * (Ta - T) - Fout * T);
  ELSE (*ємність повна*)
    T := T + (d_t / Vmax) * K1 * (Ta - T);
  END_IF;
  Ta := Ta + (d_t / Va) * (Fa * (Tain - Ta) - K2 * (Ta - T));
ELSE (*порожня ємність*)
  T := Tin;
  Ta := Ta + (d_t / Va) * (Fa * (Tain - Ta));
END_IF;

```

Дозатори

Імітаційна модель дозатору реалізована через функціональний блок (рис.4.3)

| Name | Type | Usage | Translate | Comment | Init |
|----------------|------|--------|--------------------------|--|-----------|
| <i>default</i> | | | | | |
| INIT | BOOL | Input | <input type="checkbox"/> | команда ініціалізації (задання початкових умов) | FALSE |
| Fin | REAL | Input | <input type="checkbox"/> | вхідна витрата задається в м3/с або в л/с | REAL#0.0 |
| Fout | REAL | Input | <input type="checkbox"/> | вихідна витрата задається в м3/с або в л/с | REAL#0.0 |
| d_t | REAL | Input | <input type="checkbox"/> | періодичність виклику в сек | REAL#0.1 |
| L | REAL | Output | <input type="checkbox"/> | рівень в метрах або в сантиметрах | REAL#0.0 |
| LSH | BOOL | Output | <input type="checkbox"/> | сигналізатор верхнього рівня | FALSE |
| LSL | BOOL | Output | <input type="checkbox"/> | сигналізатор нижнього рівня | FALSE |
| V0 | REAL | Local | <input type="checkbox"/> | значення об'єму при ініціалізації моделі | REAL#0.0 |
| V | REAL | Local | <input type="checkbox"/> | об'єм рідини в ємності | REAL#0.0 |
| Vmax | REAL | Local | <input type="checkbox"/> | об'єм ємності | REAL#50.0 |
| S | REAL | Local | <input type="checkbox"/> | задається в м2 при витратах м3/с або см2 при ви... | REAL#20.0 |

Рисунок 4.3 - Змінні функціонального блоку smLevelCyl

Код реалізації дозатору:

```

IF INIT THEN
  V := V0; (*мінімальний об'єм*)
ELSE
  V := V + d_t * (Fin - Fout); (*приріст об'єму*)
  IF V < 0.0 THEN V := 0.0; END_IF; (*обмеження по мінімуму*)
  IF V > Vmax THEN V := Vmax; END_IF; (*обмеження по максимуму*)
  L := V / S;
  LSH := (V / Vmax) > 0.999; (*сигналізатор верхнього рівня*)
  LSL := (V / Vmax) > 0.001; (*сигналізатор нижнього рівня*)
END_IF;

```

Трьохходовий клапан

Для імітування роботи клапану був розроблений функціональний блок smValve.

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 98 |

| Name | Type | Usage | Translate | Comment | Init |
|--------------|------|--------|--------------------------|--|-----------|
| <i>fault</i> | | | | | |
| INIT | BOOL | Input | <input type="checkbox"/> | ініціалізація | FALSE |
| cmdOPN | BOOL | Input | <input type="checkbox"/> | команда відкрити/більше | FALSE |
| cmdCLS | BOOL | Input | <input type="checkbox"/> | команда закрити/менше | FALSE |
| cmdPOS | INT | Input | <input type="checkbox"/> | задане значення позиціонера 0-10000 | INT#0 |
| stOPN | BOOL | Output | <input type="checkbox"/> | кран відкритий | FALSE |
| stCLS | BOOL | Output | <input type="checkbox"/> | кран закритий | FALSE |
| stPOS | INT | Output | <input type="checkbox"/> | положення штока/заслінки 0-10000 | INT#0 |
| Kf | REAL | Output | <input type="checkbox"/> | коефіцієнт витрати від 0 до 1 | REAL#0.0 |
| POS | REAL | Local | <input type="checkbox"/> | положення штока/заслінки 0-100% | REAL#0.0 |
| d_t | REAL | Input | <input type="checkbox"/> | періодичність виклику | REAL#0.1 |
| t_valve | REAL | Local | <input type="checkbox"/> | час повного відкриття крану/заслінки (с) | REAL#10.0 |
| APOS | BOOL | Input | <input type="checkbox"/> | TRUE - кран з позиціонером, управління аналогове | FALSE |
| v_type | INT | Local | <input type="checkbox"/> | хар-ка, 0- лінійна, 1-швидкого відкриття, 2-рівнопроцентна | INT#0 |
| cmdPOSr | REAL | Local | <input type="checkbox"/> | | REAL#0.0 |
| dposr | REAL | Local | <input type="checkbox"/> | | REAL#0.0 |

Рисунок 4.4 - Змінні функціонального блоку smValve

Код реалізації smValve:

```

IF INIT THEN
  POS := 0;
END_IF;

cmdPOSr := INT_TO_REAL(cmdPOS) * 0.01;
dposr := d_t * 100.0 / t_valve;

IF APOS THEN //аналогове керування
  IF cmdPOSr > POS + 0.5 THEN
    POS := POS + dposr;
  ELSIF cmdPOSr < POS - 0.5 THEN
    POS := POS - dposr;
  END_IF;
ELSE
  IF cmdOPN THEN
    POS := POS + dposr;
  ELSIF cmdCLS THEN
    POS := POS - dposr;
  END_IF;
END_IF;

IF POS > 100.0 THEN
  POS := 100.0;
ELSIF POS < 0.0 THEN
  POS := 0.0;
END_IF;

stOPN := POS > 99.999;
stCLS := POS < 0.001;
stPOS := REAL_TO_INT(POS * 100.0);

CASE v_type OF
  0:
    Kf := POS * 0.01;
  1:
    Kf := SQRT(POS * 0.01);
  ELSE
    Kf := POS * 0.01;
END_CASE;

```

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Лист | № докум. | Підпис | Дата |

| Name | Type | Usage | Translate | Comment | Init |
|---------------|-------------|--------|--------------------------|--|----------|
| stLSH_D2 | BOOL | Output | <input type="checkbox"/> | сигналізатор верхнього рівня дозатору D2 | FALSE |
| stLSL_D2 | BOOL | Output | <input type="checkbox"/> | сигналізатор нижнього рівня дозатору D2 | FALSE |
| stLE_T1 | INT | Output | <input type="checkbox"/> | датчик рівня танка T1 | INT#0 |
| stLE_T2 | INT | Output | <input type="checkbox"/> | датчик рівня T2 | INT#0 |
| stTE_T1 | INT | Output | <input type="checkbox"/> | датчик температури T1 | INT#0 |
| stTE_T2 | INT | Output | <input type="checkbox"/> | датчик температури T2 | INT#0 |
| stOPN_Vdoz_T1 | BOOL | Output | <input type="checkbox"/> | клапан дозування переключений на T1 | FALSE |
| stOPN_Vdoz_T2 | BOOL | Output | <input type="checkbox"/> | клапан дозування переключений на T2 | FALSE |
| stON_MixerT1 | BOOL | Output | <input type="checkbox"/> | | FALSE |
| stON_MixerT2 | BOOL | Output | <input type="checkbox"/> | | FALSE |
| smTank2 | smTankT | Local | <input type="checkbox"/> | | |
| smD1 | smLevelCyl1 | Local | <input type="checkbox"/> | | |
| smVnaborT2 | smValve | Local | <input type="checkbox"/> | | |
| smVslivT2 | smValve | Local | <input type="checkbox"/> | | |
| smVnaborD1 | smValve | Local | <input type="checkbox"/> | | |
| smVslivD1 | smValve | Local | <input type="checkbox"/> | | |
| smTank1 | smTankT | Local | <input type="checkbox"/> | | |
| smVNaborT1 | smValve | Local | <input type="checkbox"/> | | |
| smVSlivT1 | smValve | Local | <input type="checkbox"/> | | |
| smVnagrevT1 | smValve | Local | <input type="checkbox"/> | | |
| smVnagrevT2 | smValve | Local | <input type="checkbox"/> | | |
| smD2 | smLevelCyl1 | Local | <input type="checkbox"/> | | |
| smVslivD2 | smValve | Local | <input type="checkbox"/> | | |
| smVnaborD2 | smValve | Local | <input type="checkbox"/> | | |
| smVdoz_T1_T2 | smValve | Local | <input type="checkbox"/> | | |
| Ta | REAL | Local | <input type="checkbox"/> | | REAL#0.0 |
| L_D1 | REAL | Local | <input type="checkbox"/> | | REAL#0.0 |
| L_D2 | REAL | Local | <input type="checkbox"/> | | REAL#0.0 |
| d_t | REAL | Local | <input type="checkbox"/> | | REAL#0.1 |
| meaprev | BOOL | Local | <input type="checkbox"/> | | FALSE |

Рисунок 4.6 - Змінні функціонального блоку smObject1

```

IF INIT THEN
    smTank1.d_t := d_t;
    smVNaborT1.d_t := d_t;
    smVSlivT1.d_t := d_t;
    smVnagrevT1.d_t := d_t;
    smTank2.d_t := d_t;
    smVnaborT2.d_t := d_t;
    smVslivT2.d_t := d_t;
    smVnagrevT2.d_t := d_t;
    smD1.d_t := d_t;
    smVnaborD1.d_t := d_t;
    smVslivD1.d_t := d_t;
    smD2.d_t := d_t;
    smVnaborD2.d_t := d_t;
    smVslivD2.d_t := d_t;
    smVdoz_T1_T2.d_t := d_t;

    smVNaborT1(INIT := INIT,
               cmdOPN := cmdOPN_Vnabor_T1,
               cmdCLS := NOT cmdOPN_Vnabor_T1);
    smVSlivT1(INIT := INIT,
              cmdOPN := cmdOPN_Vsliv_T1,
              cmdCLS := NOT cmdOPN_Vsliv_T1);
    smVnagrevT1(INIT := INIT,
                cmdPOS := cmdPOS_Vnagrev_T1);
    smTank1(INIT := INIT,
            Fin := smVNaborT1.Kf * 0.025,
            Fout := smVSlivT1.Kf * 0.025,

```

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Лист | № докум. | Підпис | Дата |

ТО8151.0014.001.АТХ.П

Арк

101

```

Fa := smVnagrevT1.Kf * 0.035,
Tin := 20.0,
Tain := 90.0);

smVnaborT2 (INIT := INIT,
            cmdOPN := cmdOPN_Vnabor_T2,
            cmdCLS := NOT cmdOPN_Vnabor_T2);
smVslivT2 (INIT := INIT,
           cmdOPN := cmdOPN_Vsliv_T2,
           cmdCLS := NOT cmdOPN_Vsliv_T2);
smVnagrevT2 (INIT := INIT,
            cmdPOS := cmdPOS_Vnagrev_T2);
smTank2 (INIT := INIT,
         Fin := smVnaborT2.Kf * 0.025,
         Fout := smVslivT2.Kf * 0.025,
         Fa := smVnagrevT2.Kf * 0.035,
         Tin := 20.0,
         Tain := 90.0);

smVNaborT1 (INIT := INIT,
            cmdOPN := cmdOPN_Vnabor_D1,
            cmdCLS := NOT cmdOPN_Vnabor_D1);
smVslivD1 (INIT := INIT,
           cmdOPN := cmdOPN_Vsliv_D1,
           cmdCLS := NOT cmdOPN_Vsliv_D1);
smD1 (INIT := INIT,
     Fin := smVNaborT1.Kf * 1.0,
     Fout := smVslivD1.Kf * 1.0,
     L => L_D1);

smVnaborT2 (INIT := INIT,
            cmdOPN := cmdOPN_Vnabor_D2,
            cmdCLS := NOT cmdOPN_Vnabor_D2);
smVslivD2 (INIT := INIT,
           cmdOPN := cmdOPN_Vsliv_D2,
           cmdCLS := NOT cmdOPN_Vsliv_D2);
smD2 (INIT := INIT,
     Fin := smVnaborT2.Kf * 1.0,
     Fout := smVslivD2.Kf * 1.0,
     L => L_D2);

smVdoz_T1_T2 (INIT := INIT,
             cmdOPN := cmdSEL_Vdoz_T1_T2,
             cmdCLS := NOT cmdSEL_Vdoz_T1_T2);

END_IF;

stOPN_Vnabor_T1 := smVNaborT1.stOPN;
stCLS_Vnabor_T1 := smVNaborT1.stCLS;
stOPN_Vnabor_T2 := smVnaborT2.stOPN;
stCLS_Vnabor_T2 := smVnaborT2.stCLS;
stOPN_Vsliv_T1 := smVslivT1.stOPN;
stCLS_Vsliv_T1 := smVslivT1.stCLS;
stOPN_Vsliv_T2 := smVslivT2.stOPN;
stCLS_Vsliv_T2 := smVslivT2.stCLS;
stLE_T1 := REAL_TO_INT(smTank1.L * 10000.0);
stTE_T1 := REAL_TO_INT(smTank1.T * 100.0);
stLE_T2 := REAL_TO_INT(smTank2.L * 10000.0);
stTE_T2 := REAL_TO_INT(smTank2.T * 100.0);
Ta := smTank1.T;
Ta := smTank2.T;
stON_MixerT1 := cmdON_MixerT1;
stON_MixerT2 := cmdON_MixerT2;
stLSH_D1 := smD1.LSH;

```



```

    END_IF;
END_FOR;
(*тут обробка =каналів AI*)
CHAI FN1 (RAWINT := 1, CHCFG := CHCFG[0], CHHMI := CHHMI[0], PLCCFG := PLCCFG, CHBUF :=
CH_BUF1);
CHAI FN2 (RAWINT := 2, CHCFG := CHCFG[1], CHHMI := CHHMI[1], PLCCFG := PLCCFG, CHBUF :=
CH_BUF1);
CHAI FN3 (RAWINT := 3, CHCFG := CHCFG[2], CHHMI := CHHMI[2], PLCCFG := PLCCFG, CHBUF :=
CH_BUF1);
CHAI FN4 (RAWINT := 4, CHCFG := CHCFG[3], CHHMI := CHHMI[3], PLCCFG := PLCCFG, CHBUF :=
CH_BUF1);
CHAI FN5 (RAWINT := 5, CHCFG := CHCFG[4], CHHMI := CHHMI[4], PLCCFG := PLCCFG, CHBUF :=
CH_BUF1);
CHAI FN6 (RAWINT := 6, CHCFG := CHCFG[5], CHHMI := CHHMI[5], PLCCFG := PLCCFG, CHBUF :=
CH_BUF1);
CHAI FN7 (RAWINT := 7, CHCFG := CHCFG[6], CHHMI := CHHMI[6], PLCCFG := PLCCFG, CHBUF :=
CH_BUF1);
CHAI FN8 (RAWINT := 8, CHCFG := CHCFG[7], CHHMI := CHHMI[7], PLCCFG := PLCCFG, CHBUF :=
CH_BUF1);

```

dichs

Код функціонального блоку dichs:

```

FOR i := 1 TO PLCCFG.DICNT DO
    (*на першому циклі ініціалізуємо змінні ID + CLSID*)
    IF PLCCFG.STA1.SCN1 THEN
        CHCFG[i].ID := INT_TO_UINT(i);
        IF CHCFG[i].CLSID = 0 THEN CHCFG[i].CLSID := 16#0010;END_IF;
    END_IF;
END_FOR;
(*тут обробка =каналів DI*)
CHDIFN1 (RAW := TRUE, CHCFG := CHCFG[0], CHHMI := CHHMI[0], PLCCFG := PLCCFG, CHBUF
:= CH_BUF1);
CHDIFN2 (RAW := TRUE, CHCFG := CHCFG[1], CHHMI := CHHMI[1], PLCCFG := PLCCFG, CHBUF
:= CH_BUF1);
CHDIFN3 (RAW := TRUE, CHCFG := CHCFG[2], CHHMI := CHHMI[2], PLCCFG := PLCCFG, CHBUF
:= CH_BUF1);
CHDIFN4 (RAW := TRUE, CHCFG := CHCFG[3], CHHMI := CHHMI[3], PLCCFG := PLCCFG, CHBUF
:= CH_BUF1);
CHDIFN5 (RAW := TRUE, CHCFG := CHCFG[4], CHHMI := CHHMI[4], PLCCFG := PLCCFG, CHBUF
:= CH_BUF1);
CHDIFN6 (RAW := TRUE, CHCFG := CHCFG[5], CHHMI := CHHMI[5], PLCCFG := PLCCFG, CHBUF
:= CH_BUF1);
CHDIFN7 (RAW := TRUE, CHCFG := CHCFG[6], CHHMI := CHHMI[6], PLCCFG := PLCCFG, CHBUF
:= CH_BUF1);
CHDIFN8 (RAW := TRUE, CHCFG := CHCFG[7], CHHMI := CHHMI[7], PLCCFG := PLCCFG, CHBUF
:= CH_BUF1);
CHDIFN9 (RAW := TRUE, CHCFG := CHCFG[8], CHHMI := CHHMI[8], PLCCFG := PLCCFG, CHBUF
:= CH_BUF1);
CHDIFN10 (RAW := TRUE, CHCFG := CHCFG[9], CHHMI := CHHMI[9], PLCCFG := PLCCFG,
CHBUF := CH_BUF1);
CHDIFN11 (RAW := TRUE, CHCFG := CHCFG[10], CHHMI := CHHMI[10], PLCCFG := PLCCFG,
CHBUF := CH_BUF1);
CHDIFN12 (RAW := TRUE, CHCFG := CHCFG[11], CHHMI := CHHMI[11], PLCCFG := PLCCFG,
CHBUF := CH_BUF1);
CHDIFN13 (RAW := TRUE, CHCFG := CHCFG[12], CHHMI := CHHMI[12], PLCCFG := PLCCFG,
CHBUF := CH_BUF1);
CHDIFN14 (RAW := TRUE, CHCFG := CHCFG[13], CHHMI := CHHMI[13], PLCCFG := PLCCFG,
CHBUF := CH_BUF1);
CHDIFN15 (RAW := TRUE, CHCFG := CHCFG[14], CHHMI := CHHMI[14], PLCCFG := PLCCFG,
CHBUF := CH_BUF1);

```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 104 |


```

IF PLCCFG.STA1.SCN1 THEN
    CHCFG[i].ID := INT_TO_UINT(i);
    IF CHCFG[i].CLSID = 0 THEN CHCFG[i].CLSID := 16#0020;END_IF;
END_IF;
END_FOR;
(*тут обробка =каналів DO*)
CHDOFN1(CHCFG := CHCFG[0],  CHHMI := CHHMI[0],  PLCCFG := PLCCFG,  CHBUF := CH_BUF1,
RAW => RAWs[0]);
CHDOFN2(CHCFG := CHCFG[1],  CHHMI := CHHMI[1],  PLCCFG := PLCCFG,  CHBUF := CH_BUF1,
RAW => RAWs[1]);
CHDOFN3(CHCFG := CHCFG[2],  CHHMI := CHHMI[2],  PLCCFG := PLCCFG,  CHBUF := CH_BUF1,
RAW => RAWs[2]);
CHDOFN4(CHCFG := CHCFG[3],  CHHMI := CHHMI[3],  PLCCFG := PLCCFG,  CHBUF := CH_BUF1,
RAW => RAWs[3]);
CHDOFN5(CHCFG := CHCFG[4],  CHHMI := CHHMI[4],  PLCCFG := PLCCFG,  CHBUF := CH_BUF1,
RAW => RAWs[4]);
CHDOFN6(CHCFG := CHCFG[5],  CHHMI := CHHMI[5],  PLCCFG := PLCCFG,  CHBUF := CH_BUF1,
RAW => RAWs[5]);
CHDOFN7(CHCFG := CHCFG[6],  CHHMI := CHHMI[6],  PLCCFG := PLCCFG,  CHBUF := CH_BUF1,
RAW => RAWs[6]);
CHDOFN8(CHCFG := CHCFG[7],  CHHMI := CHHMI[7],  PLCCFG := PLCCFG,  CHBUF := CH_BUF1,
RAW => RAWs[7]);
CHDOFN9(CHCFG := CHCFG[8],  CHHMI := CHHMI[8],  PLCCFG := PLCCFG,  CHBUF := CH_BUF1,
RAW => RAWs[8]);
CHDOFN10(CHCFG := CHCFG[9],  CHHMI := CHHMI[9],  PLCCFG := PLCCFG,  CHBUF := CH_BUF1,
RAW => RAWs[9]);
CHDOFN11(CHCFG := CHCFG[10],  CHHMI := CHHMI[10],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[10]);
CHDOFN12(CHCFG := CHCFG[11],  CHHMI := CHHMI[11],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[11]);
CHDOFN13(CHCFG := CHCFG[12],  CHHMI := CHHMI[12],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[12]);
CHDOFN14(CHCFG := CHCFG[13],  CHHMI := CHHMI[13],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[13]);
CHDOFN15(CHCFG := CHCFG[14],  CHHMI := CHHMI[14],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[14]);
CHDOFN16(CHCFG := CHCFG[15],  CHHMI := CHHMI[15],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[15]);
CHDOFN17(CHCFG := CHCFG[16],  CHHMI := CHHMI[16],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[16]);
CHDOFN18(CHCFG := CHCFG[17],  CHHMI := CHHMI[17],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[17]);
CHDOFN19(CHCFG := CHCFG[18],  CHHMI := CHHMI[18],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[18]);
CHDOFN20(CHCFG := CHCFG[19],  CHHMI := CHHMI[19],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[19]);
CHDOFN21(CHCFG := CHCFG[20],  CHHMI := CHHMI[20],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[20]);
CHDOFN22(CHCFG := CHCFG[21],  CHHMI := CHHMI[21],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[21]);
CHDOFN23(CHCFG := CHCFG[22],  CHHMI := CHHMI[22],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[22]);
CHDOFN24(CHCFG := CHCFG[23],  CHHMI := CHHMI[23],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[23]);
CHDOFN25(CHCFG := CHCFG[24],  CHHMI := CHHMI[24],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[24]);
CHDOFN26(CHCFG := CHCFG[25],  CHHMI := CHHMI[25],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[25]);
CHDOFN27(CHCFG := CHCFG[26],  CHHMI := CHHMI[26],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[26]);
CHDOFN28(CHCFG := CHCFG[27],  CHHMI := CHHMI[27],  PLCCFG := PLCCFG,  CHBUF :=
CH_BUF1, RAW => RAWs[27]);

```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 106 |

```

CHDOFN29(CHCFG := CHCFG[28], CHHMI := CHHMI[28], PLCCFG := PLCCFG, CHBUF :=
CH_BUF1, RAW => RAWs[28]);
CHDOFN30(CHCFG := CHCFG[29], CHHMI := CHHMI[29], PLCCFG := PLCCFG, CHBUF :=
CH_BUF1, RAW => RAWs[29]);
CHDOFN31(CHCFG := CHCFG[30], CHHMI := CHHMI[30], PLCCFG := PLCCFG, CHBUF :=
CH_BUF1, RAW => RAWs[30]);
CHDOFN32(CHCFG := CHCFG[31], CHHMI := CHHMI[31], PLCCFG := PLCCFG, CHBUF :=
CH_BUF1, RAW => RAWs[31]);

```

moduls

Реалізація moduls, для обробки модулів рівню каналів:

```

(*перебір усіх модулів*)
FOR i := 0 TO PLC.MODULSCNT - 3 DO
  modtype := MODULES[i].TYPE1; (*//тип модуля*)
  modchcnts := MODULES[i].CHCNTS; (*//кількість каналів в кожному підмодулі*)
  modSTA := 0; (*стан*)
  modSTA2:=0; (*стан2*)
  (*проходження по підмодулям*)
  FOR j := 0 TO 3 DO
    zm := INT_TO_UINT(12 - 4 * j); (*зміщення для SHIFT*)
    (*тип підмодуля*)
    sbmtype := SHR(TO_BCD_WORD(TO_UINT(modtype)), zm) AND TO_WORD(16#000F);
    (*кількість каналів у підмодулі*)
    sbmchcnts:= TO_INT(SHR(TO_BCD_WORD(TO_UINT(modchcnts)), zm) AND
TO_WORD(16#000F)) + 1;
    (*початковий індекс каналу *)
    sbmstrtnmb := MODULES[i].STRTNMB[j];
    (*перевірка бітових команд*)
    mask := 16#0800; (*маска для зміщення біту команди*)
    cmdLoadsbm := (MODULES[i].STA1 AND SHR(TO_BCD_WORD(TO_UINT(mask)),
INT_TO_UINT(j))) <> 0 AND sbmtype <> 0;
    (*завантаження в буфер підмодуля*)
    IF cmdLoadsbm THEN
      SUBMODULE1.TYPE1 := sbmtype;
      SUBMODULE1.CNT := sbmchcnts;
      SUBMODULE1.STRTNMB := UINT_TO_INT(sbmstrtnmb);
    END_IF;
    (*визначення помилки на модулі по біту MERR першого каналу в модулі*)
    CASE TO_INT(sbmtype) OF
      1: (*DI*)
        sbmbad := CHDI_HMI[sbmstrtnmb].STA1.STA_ULNK; (*MERR*)
      2: (*DQ*)
        sbmbad := CHDO_HMI[sbmstrtnmb].STA1.STA_ULNK; (*MERR*)
      3: (*AI*)
        sbmbad := CHAI_HMI[sbmstrtnmb].STA1.STA_ULNK; (*MERR*)
      4: (*AO*)
        sbmbad := CHAO_HMI[sbmstrtnmb].STA1.STA_ULNK; (*MERR*)
    END_CASE;
    mask := 16#0008; (*маска для зміщення біту помилки*)
    IF sbmbad THEN
      modSTA := modSTA OR SHR(TO_BCD_WORD(TO_UINT(mask)), INT_TO_UINT(j));
    END_IF;
    (*визначення того, що цей підмодуль в буфері*)
    inbuf := sbmtype <> 0 AND (SUBMODULE1.TYPE1 = sbmtype) AND
(SUBMODULE1.STRTNMB = UINT_TO_INT(sbmstrtnmb));
    mask := 16#0080;
    IF inbuf THEN
      modSTA := modSTA OR SHR(TO_BCD_WORD(TO_UINT(mask)), INT_TO_UINT(j));
    END_IF;
    (*визначення що є хоча б один форсований канал hasfrc і хоча б один вільний
канал hasfree 11.10.21*)

```

```

hasfrc:=false;
hasfree:=false;
FOR k := 0 TO sbmchcnts-1 DO
    CASE TO_INT(sbmtype) OF
        1: (*DI*)
            hasfrc := hasfrc OR CHDI[UINT_TO_INT(sbmstrtnmb) +
k].STA1.STA_FRC;
            hasfree := hasfree OR NOT CHDI[UINT_TO_INT(sbmstrtnmb) +
k].STA1.STA_ULNK;
        2: (*DQ*)
            hasfrc := hasfrc OR CHDO[UINT_TO_INT(sbmstrtnmb) +
k].STA1.STA_FRC;
            hasfree := hasfree OR NOT CHDO[UINT_TO_INT(sbmstrtnmb) +
k].STA1.STA_ULNK;
        3: (*AI*)
            hasfrc := hasfrc OR CHAI[UINT_TO_INT(sbmstrtnmb) +
k].STA1.STA_FRC;
            hasfree := hasfree OR NOT CHAI[UINT_TO_INT(sbmstrtnmb) +
k].STA1.STA_ULNK;
        4: (*AO*)
            hasfrc := hasfrc OR CHAO[UINT_TO_INT(sbmstrtnmb) +
1].STA1.STA_FRC;
            hasfree := hasfree OR NOT CHAO[UINT_TO_INT(sbmstrtnmb) +
1].STA1.STA_ULNK;
    END_CASE;
END_FOR;
mask := 16#0008;
IF hasfrc THEN
    modSTA2 := modSTA2 OR SHR(TO_BCD_WORD(TO_UINT(mask)), INT_TO_UINT(j));
END_IF;
mask := 16#0080;
IF hasfree THEN
    modSTA2 := modSTA2 OR SHR(TO_BCD_WORD(TO_UINT(mask)), INT_TO_UINT(j));
END_IF;
(*робота підмодулем в буфері*)
IF inbuf THEN
    sbmCMD := SUBMODULE1.CMD; (*команда для підмодуля*)
    (*перевірка команди і завантаження значення каналів в буфер *)
    FOR k := 0 TO sbmchcnts-1 DO
        cmdLoadch := TO_INT(sbmCMD) = (k + 1); (*завантажити канал*)
        CASE TO_INT(sbmtype) OF
            1: (*DI*)
                CHDI_HMI[UINT_TO_INT(sbmstrtnmb) + k].STA1.STA_SML :=
cmdLoadch;
                SUBMODULE1.CH[k] := CHDI_HMI[UINT_TO_INT(sbmstrtnmb) + k];
            2: (*DQ*)
                CHDO_HMI[UINT_TO_INT(sbmstrtnmb) + k].STA1.STA_SML :=
cmdLoadch;
                SUBMODULE1.CH[k] := CHDO_HMI[UINT_TO_INT(sbmstrtnmb) + k];
            3: (*AI*)
                CHAI_HMI[UINT_TO_INT(sbmstrtnmb) + k].STA1.STA_SML :=
cmdLoadch;
                SUBMODULE1.CH[k] := CHAI_HMI[UINT_TO_INT(sbmstrtnmb) + k];
            4: (*AO*)
                CHAO_HMI[UINT_TO_INT(sbmstrtnmb) + 1].STA1.STA_SML :=
cmdLoadch;
                SUBMODULE1.CH[1] := CHAO_HMI[UINT_TO_INT(sbmstrtnmb) + 1];
        END_CASE;
    END_FOR;
    SUBMODULE1.CMD := 0;
END_IF;
END_FOR;
(*запис стану в модуль*)

```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 108 |

```

MODULES[i].STA1 := modSTA;
MODULES[i].STA2:= modSTA2;

END_FOR;

FOR i := 0 TO 15 DO
    SUBMODULE1.CH[i].STA1.STA_SHRT:=SUBMODULE1.CNT-1 < i;
END_FOR;

```

4.3. Обробка технологічних змінних

Для обробки вхідних технологічних змінних було реалізовано функції:

- aivars – обробка аналогових вхідних змінних
- divars - обробка дискретних вхідних змінних
- aovars - обробка аналогових вихідних змінних
- dovars - обробка дискретних вихідних змінних

aivars

Код функціонального блоку aivars:

```

AIVARFN1(CHCFG := CHAI[VAR1.T1_LT1.CHID], AIVARCFG := VAR1.T1_LT1, AIVARHMI :=
AIH1.T1_LT1, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHAI := CHAI); (*рівень T1*)
AIVARFN2(CHCFG := CHAI[VAR1.T2_LT1.CHID], AIVARCFG := VAR1.T2_LT1, AIVARHMI :=
AIH1.T2_LT1, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHAI := CHAI); (*рівень T2*)
AIVARFN3(CHCFG := CHAI[VAR1.T1_TT1.CHID], AIVARCFG := VAR1.T1_TT1, AIVARHMI :=
AIH1.T1_TT1, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHAI := CHAI); (*температура в
T1*)
AIVARFN4(CHCFG := CHAI[VAR1.T2_TT1.CHID], AIVARCFG := VAR1.T2_TT1, AIVARHMI :=
AIH1.T2_TT1, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHAI := CHAI); (*температура в
T2*)
AIVARFN5(CHCFG := CHAI[VAR1.REZAI1.CHID], AIVARCFG := VAR1.REZAI1, AIVARHMI :=
AIH1.REZAI1, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHAI := CHAI); (*rez*)
AIVARFN6(CHCFG := CHAI[VAR1.REZAI2.CHID], AIVARCFG := VAR1.REZAI2, AIVARHMI :=
AIH1.REZAI2, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHAI := CHAI); (*rez*)
AIVARFN7(CHCFG := CHAI[VAR1.REZAI3.CHID], AIVARCFG := VAR1.REZAI3, AIVARHMI :=
AIH1.REZAI3, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHAI := CHAI); (*rez*)
AIVARFN8(CHCFG := CHAI[VAR1.REZAI4.CHID], AIVARCFG := VAR1.REZAI4, AIVARHMI :=
AIH1.REZAI4, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHAI := CHAI); (*rez*)

```

divars

Код функціонального блоку divars:

```

DIVARFN1(CHCFG := CHDI[VAR1.T1_VSLD_SOPN.CHID], DIVARCFG := VAR1.T1_VSLD_SOPN,
DIVARHMI := DIH1.T1_VSLD_SOPN, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI);
(*клапан набору T1 відкритий*)
DIVARFN2(CHCFG := CHDI[VAR1.T2_VSLD_SOPN.CHID], DIVARCFG := VAR1.T2_VSLD_SOPN,
DIVARHMI := DIH1.T2_VSLD_SOPN, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI);
(*клапан набору T2 відкритий*)
DIVARFN3(CHCFG := CHDI[VAR1.D1_VSUNL_SCLS.CHID], DIVARCFG := VAR1.D1_VSUNL_SCLS,
DIVARHMI := DIH1.D1_VSUNL_SCLS, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI);
(*клапан зливу D1 закритий*)
DIVARFN4(CHCFG := CHDI[VAR1.D2_VSUNL_SCLS.CHID], DIVARCFG := VAR1.D2_VSUNL_SCLS,
DIVARHMI := DIH1.D2_VSUNL_SCLS, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI);
(*клапан зливу D2 закритий*)

```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 109 |

DIVARFN5(CHCFG := CHDI[VAR1.D1_LSH.CHID], DIVARCFG := VAR1.D1_LSH, DIVARHMI := DIH1.D1_LSH, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*сигналізатор верхнього рівня D1*)
 DIVARFN6(CHCFG := CHDI[VAR1.D1_LSL.CHID], DIVARCFG := VAR1.D1_LSL, DIVARHMI := DIH1.D1_LSL, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*сигналізатор нижнього рівня D1*)
 DIVARFN7(CHCFG := CHDI[VAR1.D2_LSH.CHID], DIVARCFG := VAR1.D2_LSH, DIVARHMI := DIH1.D2_LSH, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*сигналізатор верхнього рівня D2*)
 DIVARFN8(CHCFG := CHDI[VAR1.D2_LSL.CHID], DIVARCFG := VAR1.D2_LSL, DIVARHMI := DIH1.D2_LSL, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*сигналізатор нижнього рівня D2*)
 DIVARFN9(CHCFG := CHDI[VAR1.T1_VSLD_SCLS.CHID], DIVARCFG := VAR1.T1_VSLD_SCLS, DIVARHMI := DIH1.T1_VSLD_SCLS, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*клапан набору T1 закритий*)
 DIVARFN10(CHCFG := CHDI[VAR1.T2_VSLD_SCLS.CHID], DIVARCFG := VAR1.T2_VSLD_SCLS, DIVARHMI := DIH1.T2_VSLD_SCLS, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*клапан набору T2 закритий*)
 DIVARFN11(CHCFG := CHDI[VAR1.T1_VSUNL_SCLS.CHID], DIVARCFG := VAR1.T1_VSUNL_SCLS, DIVARHMI := DIH1.T1_VSUNL_SCLS, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*клапан зливу T1 закритий*)
 DIVARFN12(CHCFG := CHDI[VAR1.T2_VSUNL_SCLS.CHID], DIVARCFG := VAR1.T2_VSUNL_SCLS, DIVARHMI := DIH1.T2_VSUNL_SCLS, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*клапан зливу T2 закритий*)
 DIVARFN13(CHCFG := CHDI[VAR1.D1_VSLD_SCLS.CHID], DIVARCFG := VAR1.D1_VSLD_SCLS, DIVARHMI := DIH1.D1_VSLD_SCLS, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*клапан набору D1 закритий*)
 DIVARFN14(CHCFG := CHDI[VAR1.D2_VSLD_SCLS.CHID], DIVARCFG := VAR1.D2_VSLD_SCLS, DIVARHMI := DIH1.D2_VSLD_SCLS, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*клапан набору D2 закритий*)
 DIVARFN15(CHCFG := CHDI[VAR1.P1_VST1T2_SCLS.CHID], DIVARCFG := VAR1.P1_VST1T2_SCLS, DIVARHMI := DIH1.P1_VST1T2_SCLS, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*3-ходовий клапан в позиції T1*)
 DIVARFN16(CHCFG := CHDI[VAR1.P1_VST1T2_SOPN.CHID], DIVARCFG := VAR1.P1_VST1T2_SOPN, DIVARHMI := DIH1.P1_VST1T2_SOPN, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*3-ходовий клапан в позиції T2 *)
 DIVARFN17(CHCFG := CHDI[VAR1.T2_GSMIX_RUN.CHID], DIVARCFG := VAR1.T2_GSMIX_RUN, DIVARHMI := DIH1.T2_GSMIX_RUN, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*Контакт пускача двигуна мішалки T2*)
 DIVARFN18(CHCFG := CHDI[VAR1.T1_GSMIX_RUN.CHID], DIVARCFG := VAR1.T1_GSMIX_RUN, DIVARHMI := DIH1.T1_GSMIX_RUN, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*Контакт пускача двигуна мішалки T1*)
 DIVARFN19(CHCFG := CHDI[VAR1.REZDI1.CHID], DIVARCFG := VAR1.REZDI1, DIVARHMI := DIH1.REZDI1, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*rez*)
 DIVARFN20(CHCFG := CHDI[VAR1.REZDI2.CHID], DIVARCFG := VAR1.REZDI2, DIVARHMI := DIH1.REZDI2, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*rez*)
 DIVARFN21(CHCFG := CHDI[VAR1.REZDI3.CHID], DIVARCFG := VAR1.REZDI3, DIVARHMI := DIH1.REZDI3, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*rez*)
 DIVARFN22(CHCFG := CHDI[VAR1.REZDI4.CHID], DIVARCFG := VAR1.REZDI4, DIVARHMI := DIH1.REZDI4, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*rez*)
 DIVARFN23(CHCFG := CHDI[VAR1.REZDI5.CHID], DIVARCFG := VAR1.REZDI5, DIVARHMI := DIH1.REZDI5, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*rez*)
 DIVARFN24(CHCFG := CHDI[VAR1.REZDI6.CHID], DIVARCFG := VAR1.REZDI6, DIVARHMI := DIH1.REZDI6, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*rez*)
 DIVARFN25(CHCFG := CHDI[VAR1.REZDI7.CHID], DIVARCFG := VAR1.REZDI7, DIVARHMI := DIH1.REZDI7, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*rez*)
 DIVARFN26(CHCFG := CHDI[VAR1.REZDI8.CHID], DIVARCFG := VAR1.REZDI8, DIVARHMI := DIH1.REZDI8, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*rez*)
 DIVARFN27(CHCFG := CHDI[VAR1.REZDI9.CHID], DIVARCFG := VAR1.REZDI9, DIVARHMI := DIH1.REZDI9, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*rez*)
 DIVARFN28(CHCFG := CHDI[VAR1.REZDI10.CHID], DIVARCFG := VAR1.REZDI10, DIVARHMI := DIH1.REZDI10, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*rez*)

| | | | | | | | |
|-----|------|----------|--------|------|--|-----------------------|-----|
| | | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | | 110 |

```

DIVARFN29(CHCFG := CHDI[VAR1.REZDI11.CHID], DIVARCFG := VAR1.REZDI11, DIVARHMI :=
DIH1.REZDI11, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*rez*)
DIVARFN30(CHCFG := CHDI[VAR1.REZDI12.CHID], DIVARCFG := VAR1.REZDI12, DIVARHMI :=
DIH1.REZDI12, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*rez*)
DIVARFN31(CHCFG := CHDI[VAR1.REZDI13.CHID], DIVARCFG := VAR1.REZDI13, DIVARHMI :=
DIH1.REZDI13, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*rez*)
DIVARFN32(CHCFG := CHDI[VAR1.REZDI14.CHID], DIVARCFG := VAR1.REZDI14, DIVARHMI :=
DIH1.REZDI14, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDI := CHDI); (*rez*)

```

aovars

Код функціонального блоку aovars:

```

AOVARFN1(CHCFG := CHAO[VAR1.T1_VRHEA_CPOS.CHID], AOVARCFG := VAR1.T1_VRHEA_CPOS,
AOVARHMI := AOH1.T1_VRHEA_CPOS, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHAO := CHAO);
(*клапан нагрівання T1*)
AOVARFN2(CHCFG := CHAO[VAR1.T2_VRHEA_CPOS.CHID], AOVARCFG := VAR1.T2_VRHEA_CPOS,
AOVARHMI := AOH1.T2_VRHEA_CPOS, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHAO := CHAO);
(*клапан нагрівання T2*)
AOVARFN3(CHCFG := CHAO[VAR1.T1_SCMIX_CSPD.CHID], AOVARCFG := VAR1.T1_SCMIX_CSPD,
AOVARHMI := AOH1.T1_SCMIX_CSPD, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHAO := CHAO);
(*задана швидкість перемішування T1*)
AOVARFN4(CHCFG := CHAO[VAR1.T2_SCMIX_CSPD.CHID], AOVARCFG := VAR1.T2_SCMIX_CSPD,
AOVARHMI := AOH1.T2_SCMIX_CSPD, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHAO := CHAO);
(*задана швидкість перемішування T2*)

```

dovars

Код функціонального блоку dovars:

```

DOVARFN1(CHCFG := CHDO[VAR1.T1_VSLD_COPN.CHID], DOVARCFG := VAR1.T1_VSLD_COPN,
DOVARHMI := DOH1.T1_VSLD_COPN, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO);
(*клапан набору T1 відкрити/закрити*)
DOVARFN2(CHCFG := CHDO[VAR1.T1_VSUNL_COPN.CHID], DOVARCFG := VAR1.T1_VSUNL_COPN,
DOVARHMI := DOH1.T1_VSUNL_COPN, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO);
(*клапан зливу T1 відкрити/закрити*)
DOVARFN3(CHCFG := CHDO[VAR1.T2_VSLD_COPN.CHID], DOVARCFG := VAR1.T2_VSLD_COPN,
DOVARHMI := DOH1.T2_VSLD_COPN, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO);
(*клапан набору T2 відкрити/закрити*)
DOVARFN4(CHCFG := CHDO[VAR1.T2_VSUNL_COPN.CHID], DOVARCFG := VAR1.T2_VSUNL_COPN,
DOVARHMI := DOH1.T2_VSUNL_COPN, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO);
(*клапан зливу T2 відкрити/закрити*)
DOVARFN5(CHCFG := CHDO[VAR1.D1_VSLD_COPN.CHID], DOVARCFG := VAR1.D1_VSLD_COPN,
DOVARHMI := DOH1.D1_VSLD_COPN, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO);
(*клапан набору D1 відкрити/закрити*)
DOVARFN6(CHCFG := CHDO[VAR1.D2_VSLD_COPN.CHID], DOVARCFG := VAR1.D2_VSLD_COPN,
DOVARHMI := DOH1.D2_VSLD_COPN, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO);
(*клапан набору D2 відкрити/закрити*)
DOVARFN7(CHCFG := CHDO[VAR1.D1_VSUNL_COPN.CHID], DOVARCFG := VAR1.D1_VSUNL_COPN,
DOVARHMI := DOH1.D1_VSUNL_COPN, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO);
(*клапан зливу D1 відкрити/закрити*)
DOVARFN8(CHCFG := CHDO[VAR1.D2_VSUNL_COPN.CHID], DOVARCFG := VAR1.D2_VSUNL_COPN,
DOVARHMI := DOH1.D2_VSUNL_COPN, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO);
(*клапан зливу D2 відкрити/закрити*)
DOVARFN9(CHCFG := CHDO[VAR1.P1_VST1T2_COPN.CHID], DOVARCFG := VAR1.P1_VST1T2_COPN,
DOVARHMI := DOH1.P1_VST1T2_COPN, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO);
(*3-ходовий клапан (0 - на T1, 1 - на T2)*)
DOVARFN10(CHCFG := CHDO[VAR1.T1_GSMIX_CSTRT.CHID], DOVARCFG := VAR1.T1_GSMIX_CSTRT,
DOVARHMI := DOH1.T1_GSMIX_CSTRT, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO);
(*Пускач двигуна мішалки T1*)

```

| | | | | | | | |
|-----|------|----------|--------|------|--|-----------------------|-----|
| | | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | | 111 |

```

DOVARFN11(CHCFG := CHDO[VAR1.T2_GSMIX_CSTRT.CHID], DOVARCFG := VAR1.T2_GSMIX_CSTRT,
DOVARHMI := DOH1.T2_GSMIX_CSTRT, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO);
(*Пускач двигуна мішалки T2*)
DOVARFN12(CHCFG := CHDO[VAR1.REZDO1.CHID], DOVARCFG := VAR1.REZDO1, DOVARHMI :=
DOH1.REZDO1, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN13(CHCFG := CHDO[VAR1.REZDO2.CHID], DOVARCFG := VAR1.REZDO2, DOVARHMI :=
DOH1.REZDO2, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN14(CHCFG := CHDO[VAR1.REZDO3.CHID], DOVARCFG := VAR1.REZDO3, DOVARHMI :=
DOH1.REZDO3, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN15(CHCFG := CHDO[VAR1.REZDO4.CHID], DOVARCFG := VAR1.REZDO4, DOVARHMI :=
DOH1.REZDO4, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN16(CHCFG := CHDO[VAR1.REZDO5.CHID], DOVARCFG := VAR1.REZDO5, DOVARHMI :=
DOH1.REZDO5, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN17(CHCFG := CHDO[VAR1.REZDO6.CHID], DOVARCFG := VAR1.REZDO6, DOVARHMI :=
DOH1.REZDO6, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN18(CHCFG := CHDO[VAR1.REZDO7.CHID], DOVARCFG := VAR1.REZDO7, DOVARHMI :=
DOH1.REZDO7, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN19(CHCFG := CHDO[VAR1.REZDO8.CHID], DOVARCFG := VAR1.REZDO8, DOVARHMI :=
DOH1.REZDO8, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN20(CHCFG := CHDO[VAR1.REZDO9.CHID], DOVARCFG := VAR1.REZDO9, DOVARHMI :=
DOH1.REZDO9, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN21(CHCFG := CHDO[VAR1.REZDO10.CHID], DOVARCFG := VAR1.REZDO10, DOVARHMI :=
DOH1.REZDO10, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN22(CHCFG := CHDO[VAR1.REZDO11.CHID], DOVARCFG := VAR1.REZDO11, DOVARHMI :=
DOH1.REZDO11, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN23(CHCFG := CHDO[VAR1.REZDO12.CHID], DOVARCFG := VAR1.REZDO12, DOVARHMI :=
DOH1.REZDO12, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN24(CHCFG := CHDO[VAR1.REZDO13.CHID], DOVARCFG := VAR1.REZDO13, DOVARHMI :=
DOH1.REZDO13, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN25(CHCFG := CHDO[VAR1.REZDO14.CHID], DOVARCFG := VAR1.REZDO14, DOVARHMI :=
DOH1.REZDO14, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN26(CHCFG := CHDO[VAR1.REZDO15.CHID], DOVARCFG := VAR1.REZDO15, DOVARHMI :=
DOH1.REZDO15, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN27(CHCFG := CHDO[VAR1.REZDO16.CHID], DOVARCFG := VAR1.REZDO16, DOVARHMI :=
DOH1.REZDO16, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN28(CHCFG := CHDO[VAR1.REZDO17.CHID], DOVARCFG := VAR1.REZDO17, DOVARHMI :=
DOH1.REZDO17, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN29(CHCFG := CHDO[VAR1.REZDO18.CHID], DOVARCFG := VAR1.REZDO18, DOVARHMI :=
DOH1.REZDO18, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN30(CHCFG := CHDO[VAR1.REZDO19.CHID], DOVARCFG := VAR1.REZDO19, DOVARHMI :=
DOH1.REZDO19, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN31(CHCFG := CHDO[VAR1.REZDO20.CHID], DOVARCFG := VAR1.REZDO20, DOVARHMI :=
DOH1.REZDO20, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)
DOVARFN32(CHCFG := CHDO[VAR1.REZDO21.CHID], DOVARCFG := VAR1.REZDO21, DOVARHMI :=
DOH1.REZDO21, VARBUF1 := VARBUF1, PLCCFG := PLCCFG, CHDO := CHDO); (*rez*)

```

4.4.Обробка пристроїв

Для обробки виконавчих механізмів було реалізовано функції:

- resolution – формування дозволів на керування
- actrs – код для обробки виконавчих механізмів

resolution

Код функціонального блоку resolution:

```

(* Якщо не відправляється команда TRUE, ВМ не виконує жодної команди керування*)
ACT1.T1_VSLD.CMD.CMD2.CMD_RESOLUTION:=TRUE;
ACT1.T2_VSLD.CMD.CMD2.CMD_RESOLUTION:=TRUE;

```

| | | | | | | | |
|-----|------|----------|--------|------|--|-----------------------|-----|
| | | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | | 112 |

```

ACT1.T1_VSUNL.CMD.CMD2.CMD_RESOLUTION:=TRUE;
ACT1.T2_VSUNL.CMD.CMD2.CMD_RESOLUTION:=TRUE;
ACT1.D1_VSLD.CMD.CMD2.CMD_RESOLUTION:=TRUE;
ACT1.D2_VSLD.CMD.CMD2.CMD_RESOLUTION:=TRUE;
ACT1.D1_VSUNL.CMD.CMD2.CMD_RESOLUTION:=TRUE;
ACT1.D2_VSUNL.CMD.CMD2.CMD_RESOLUTION:=TRUE;
ACT1.P1_VST1T2.CMD.CMD2.CMD_RESOLUTION:=TRUE;
ACT1.T1_VRHEA.CMD.CMD2.CMD_RESOLUTION:=TRUE;
ACT1.T2_VRHEA.CMD.CMD2.CMD_RESOLUTION:=TRUE;
ACT1.T1_SCMIX.CMD.CMD2.CMD_RESOLUTION:=TRUE;
ACT1.T2_SCMIX.CMD.CMD2.CMD_RESOLUTION:=TRUE;

```

actrs

Код функціонального блоку actrs:

```

DIVAR_TMP.ID:=0;
DIVAR_TMP.CLSID:=0;
DOVAR_TMP.ID:=0;
DOVAR_TMP.CLSID:=0;
AIVAR_TMP.ID:=0;
AIVAR_TMP.CLSID:=0;
AOVAR_TMP.ID:=0;
AOVAR_TMP.CLSID:=0;
(* ----- Виконавчі механізми DRV1 - Двигун ВКЛ/ОТКЛ ----- *)
(* ----- Виконавчі механізми VLVD1 - Клапан ВІДКР/ЗАКР ----- *)
VLVDFN1(ACTCFG:=ACT1.T1_VSLD, АСТНМІ:=ACTH1.T1_VSLD, SCLS:=VARS1.T1_VSLD_SCLS,
SOPN:=VARS1.T1_VSLD_SOPN, COPN:=VARS1.T1_VSLD_COPN, PLCCFG := PLCCFG, ACTBUF :=
ACTBUF ); (* клапан набору T1 відкрити/закрити *)
VLVDFN2(ACTCFG:=ACT1.T2_VSLD, АСТНМІ:=ACTH1.T2_VSLD, SCLS:=VARS1.T2_VSLD_SCLS,
SOPN:=VARS1.T2_VSLD_SOPN, COPN:=VARS1.T2_VSLD_COPN, PLCCFG := PLCCFG, ACTBUF :=
ACTBUF ); (* клапан набору T2 відкрити/закрити *)
VLVDFN3(ACTCFG:=ACT1.T1_VSUNL, АСТНМІ:=ACTH1.T1_VSUNL, SCLS:=VARS1.T1_VSUNL_SCLS,
SOPN:=DIVAR_TMP, COPN:=VARS1.T1_VSUNL_COPN, PLCCFG := PLCCFG, ACTBUF := ACTBUF ); (*
клапан зливу T1 відкрити/закрити *)
VLVDFN4(ACTCFG:=ACT1.T2_VSUNL, АСТНМІ:=ACTH1.T2_VSUNL, SCLS:=VARS1.T2_VSUNL_SCLS,
SOPN:=DIVAR_TMP, COPN:=VARS1.T2_VSUNL_COPN, PLCCFG := PLCCFG, ACTBUF := ACTBUF ); (*
клапан зливу T2 відкрити/закрити *)
VLVDFN5(ACTCFG:=ACT1.D1_VSLD, АСТНМІ:=ACTH1.D1_VSLD, SCLS:=VARS1.D1_VSLD_SCLS,
SOPN:=DIVAR_TMP, COPN:=VARS1.D1_VSLD_COPN, PLCCFG := PLCCFG, ACTBUF := ACTBUF ); (*
клапан набору D1 відкрити/закрити *)
VLVDFN6(ACTCFG:=ACT1.D2_VSLD, АСТНМІ:=ACTH1.D2_VSLD, SCLS:=VARS1.D2_VSLD_SCLS,
SOPN:=DIVAR_TMP, COPN:=VARS1.D2_VSLD_COPN, PLCCFG := PLCCFG, ACTBUF := ACTBUF ); (*
клапан набору D2 відкрити/закрити *)
VLVDFN7(ACTCFG:=ACT1.D1_VSUNL, АСТНМІ:=ACTH1.D1_VSUNL, SCLS:=VARS1.D1_VSUNL_SCLS,
SOPN:=DIVAR_TMP, COPN:=VARS1.D1_VSUNL_COPN, PLCCFG := PLCCFG, ACTBUF := ACTBUF ); (*
клапан зливу D1 відкрити/закрити *)
VLVDFN8(ACTCFG:=ACT1.D2_VSUNL, АСТНМІ:=ACTH1.D2_VSUNL, SCLS:=VARS1.D2_VSUNL_SCLS,
SOPN:=DIVAR_TMP, COPN:=VARS1.D2_VSUNL_COPN, PLCCFG := PLCCFG, ACTBUF := ACTBUF ); (*
клапан зливу D2 відкрити/закрити *)
VLVDFN9(ACTCFG:=ACT1.P1_VST1T2, АСТНМІ:=ACTH1.P1_VST1T2, SCLS:=VARS1.P1_VST1T2_SCLS,
SOPN:=VARS1.P1_VST1T2_SOPN, COPN:=VARS1.P1_VST1T2_COPN, PLCCFG := PLCCFG, ACTBUF :=
ACTBUF ); (* 3-ходовий клапан *)
(* ----- Виконавчі механізми VLVA1 - Клапан регулюючий ----- *)
VLVAFN1(ACTCFG:=ACT1.T1_VRHEA, АСТНМІ:=ACTH1.T1_VRHEA, CPOS:=VARS1.T1_VRHEA_CPOS,
PLCCFG := PLCCFG, ACTBUF := ACTBUF ); (* клапан нагрівання T1 *)
VLVAFN2(ACTCFG:=ACT1.T2_VRHEA, АСТНМІ:=ACTH1.T2_VRHEA, CPOS:=VARS1.T2_VRHEA_CPOS,
PLCCFG := PLCCFG, ACTBUF := ACTBUF ); (* клапан нагрівання T2 *)
(* ----- Виконавчі механізми DRVS1 - Двигун з частотником ----- *)
DRVFN1(ACTCFG:=ACT1.T1_SCMIX, АСТНМІ:=ACTH1.T1_SCMIX, RUN:=VARS1.T1_GSMIX_RUN,
CSTRT:=VARS1.T1_GSMIX_CSTRT, CSPD:=VARS1.T1_SCMIX_CSPD, PLCCFG := PLCCFG, ACTBUF :=
ACTBUF ); (* задана швидкість перемішування T1 *)

```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 113 |

```
DRVFN2 (ACTCFG:=ACT1.T2_SCMIX, ACTHMI:=ACTH1.T2_SCMIX, RUN:=VARS1.T2_GSMIX_RUN,
CSTRT:=VARS1.T2_GSMIX_CSTRT, CSPD:=VARS1.T2_SCMIX_CSPD, PLCCFG := PLCCFG, ACTBUF :=
ACTBUF ); (* задана швидкість перемішування T2 *)
```

4.5. Формування карти ПЛК та ініціалізування

Було створено функціональні блоки для ініціалізації рівнів LVL0 та LVL1 при запуску програми.

plcmaps

Код функціонального блоку plcmaps:

```
(*кількість каналів та модулів*)
PLCCFG.DICNT := 32;
PLCCFG.DOCNT := 32;
PLCCFG.AICNT := 8;
PLCCFG.AOCNT := 4;
PLCCFG.MODULSCNT := 4;
(*завантажити в буфер підмодуль 0 модуля 0*)
MODULES[0].STA1.%X11 := true;
(*типи 1- DICH, 2- DOCH, 3- AICH, 4 - AOCH, 5 - COM*)
MODULES[0].TYPE1 := 16#1200; (*DI(1..16) DO(1..16) *)
MODULES[0].CHCNTS := 16#ff00; (*16-16-0-0*)
MODULES[0].STRTNMB[0] := 1;
MODULES[0].STRTNMB[1] := 1;
MODULES[0].STRTNMB[2] := 0;
MODULES[0].STRTNMB[3] := 0;

MODULES[1].TYPE1 := 16#3400; (*AI(1..4) AO(1..2) *)
MODULES[1].CHCNTS := 16#3100; (*4-2-0-0*)
MODULES[1].STRTNMB[0] := 1;
MODULES[1].STRTNMB[1] := 1;
MODULES[1].STRTNMB[2] := 0;
MODULES[1].STRTNMB[3] := 0;

MODULES[2].TYPE1 := 16#1200; (*DI(17..32) DO(17..32) *)
MODULES[2].CHCNTS := 16#ff00; (*16-16-0-0*)
MODULES[2].STRTNMB[0] := 17;
MODULES[2].STRTNMB[1] := 17;
MODULES[2].STRTNMB[2] := 0;
MODULES[2].STRTNMB[3] := 0;

MODULES[3].TYPE1 := 16#3400; (*AI(5..8) AO(3..4) *)
MODULES[3].CHCNTS := 16#3100; (*4-2-0-0*)
MODULES[3].STRTNMB[0] := 5;
MODULES[3].STRTNMB[1] := 3;
MODULES[3].STRTNMB[2] := 0;
MODULES[3].STRTNMB[3] := 0;
```

INITVARS

Код функціонального блоку INITVARS:

```
VARS1.T1_LT1.ID:=1; VARS1.T1_LT1.CHID:=1; VARS1.T1_LT1.CHIDDF:=1;
VARS1.T2_LT1.ID:=2; VARS1.T2_LT1.CHID:=2; VARS1.T2_LT1.CHIDDF:=2;
VARS1.T1_TT1.ID:=3; VARS1.T1_TT1.CHID:=3; VARS1.T1_TT1.CHIDDF:=3;
VARS1.T2_TT1.ID:=4; VARS1.T2_TT1.CHID:=4; VARS1.T2_TT1.CHIDDF:=4;
VARS1.REZAI1.ID:=5; VARS1.REZAI1.CHID:=0; VARS1.REZAI1.CHIDDF:=0;
VARS1.REZAI2.ID:=6; VARS1.REZAI2.CHID:=0; VARS1.REZAI2.CHIDDF:=0;
```

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 114 |

VARS1.REZAI3.ID:=7; VARS1.REZAI3.CHID:=0; VARS1.REZAI3.CHIDDF:=0;
 VARS1.REZAI4.ID:=8; VARS1.REZAI4.CHID:=0; VARS1.REZAI4.CHIDDF:=0;
 VARS1.T1_VRHEA_CPOS.ID:=5001;VARS1.T1_VRHEA_CPOS.CHID:=1;
 VARS1.T1_VRHEA_CPOS.CHIDDF:=1;
 VARS1.T2_VRHEA_CPOS.ID:=5002;VARS1.T2_VRHEA_CPOS.CHID:=2;
 VARS1.T2_VRHEA_CPOS.CHIDDF:=2;
 VARS1.T1_SCMIX_CSPD.ID:=5003;VARS1.T1_SCMIX_CSPD.CHID:=3;
 VARS1.T1_SCMIX_CSPD.CHIDDF:=3;
 VARS1.T2_SCMIX_CSPD.ID:=5004;VARS1.T2_SCMIX_CSPD.CHID:=4;
 VARS1.T2_SCMIX_CSPD.CHIDDF:=4;
 VARS1.T1_VSLD_SOPN.ID:=10001;VARS1.T1_VSLD_SOPN.CHID:=1;
 VARS1.T1_VSLD_SOPN.CHIDDF:=1;
 VARS1.T2_VSLD_SOPN.ID:=10002;VARS1.T2_VSLD_SOPN.CHID:=2;
 VARS1.T2_VSLD_SOPN.CHIDDF:=2;
 VARS1.D1_VSUNL_SCLS.ID:=10003;VARS1.D1_VSUNL_SCLS.CHID:=11;
 VARS1.D1_VSUNL_SCLS.CHIDDF:=11;
 VARS1.D2_VSUNL_SCLS.ID:=10004;VARS1.D2_VSUNL_SCLS.CHID:=12;
 VARS1.D2_VSUNL_SCLS.CHIDDF:=12;
 VARS1.D1_LSH.ID:=10005; VARS1.D1_LSH.CHID:=13; VARS1.D1_LSH.CHIDDF:=13;
 VARS1.D1_LSL.ID:=10006; VARS1.D1_LSL.CHID:=14; VARS1.D1_LSL.CHIDDF:=14;
 VARS1.D2_LSH.ID:=10007; VARS1.D2_LSH.CHID:=15; VARS1.D2_LSH.CHIDDF:=15;
 VARS1.D2_LSL.ID:=10008; VARS1.D2_LSL.CHID:=16; VARS1.D2_LSL.CHIDDF:=16;
 VARS1.T1_VSLD_SCLS.ID:=10009; VARS1.T1_VSLD_SCLS.CHID:=5;
 VARS1.T1_VSLD_SCLS.CHIDDF:=5;
 VARS1.T2_VSLD_SCLS.ID:=10010;VARS1.T2_VSLD_SCLS.CHID:=6;
 VARS1.T2_VSLD_SCLS.CHIDDF:=6;
 VARS1.T1_VSUNL_SCLS.ID:=10011;VARS1.T1_VSUNL_SCLS.CHID:=7;
 VARS1.T1_VSUNL_SCLS.CHIDDF:=7;
 VARS1.T2_VSUNL_SCLS.ID:=10012;VARS1.T2_VSUNL_SCLS.CHID:=8;
 VARS1.T2_VSUNL_SCLS.CHIDDF:=8;
 VARS1.D1_VSLD_SCLS.ID:=10013;VARS1.D1_VSLD_SCLS.CHID:=9;
 VARS1.D1_VSLD_SCLS.CHIDDF:=9;
 VARS1.D2_VSLD_SCLS.ID:=10014;VARS1.D2_VSLD_SCLS.CHID:=10;
 VARS1.D2_VSLD_SCLS.CHIDDF:=10;
 VARS1.P1_VST1T2_SCLS.ID:=10015;VARS1.P1_VST1T2_SCLS.CHID:=17;
 VARS1.P1_VST1T2_SCLS.CHIDDF:=17;
 VARS1.P1_VST1T2_SOPN.ID:=10016;VARS1.P1_VST1T2_SOPN.CHID:=18;
 VARS1.P1_VST1T2_SOPN.CHIDDF:=18;
 VARS1.T2_GSMIX_RUN.ID:=10017;VARS1.T2_GSMIX_RUN.CHID:=19;
 VARS1.T2_GSMIX_RUN.CHIDDF:=19;
 VARS1.T1_GSMIX_RUN.ID:=10018;VARS1.T1_GSMIX_RUN.CHID:=20;
 VARS1.T1_GSMIX_RUN.CHIDDF:=20;
 VARS1.REZDI1.ID:=10019; VARS1.REZDI1.CHID:=0; VARS1.REZDI1.CHIDDF:=0;
 VARS1.REZDI2.ID:=10020; VARS1.REZDI2.CHID:=0; VARS1.REZDI2.CHIDDF:=0;
 VARS1.REZDI3.ID:=10021; VARS1.REZDI3.CHID:=0; VARS1.REZDI3.CHIDDF:=0;
 VARS1.REZDI4.ID:=10022; VARS1.REZDI4.CHID:=0; VARS1.REZDI4.CHIDDF:=0;
 VARS1.REZDI5.ID:=10023; VARS1.REZDI5.CHID:=0; VARS1.REZDI5.CHIDDF:=0;
 VARS1.REZDI6.ID:=10024; VARS1.REZDI6.CHID:=0; VARS1.REZDI6.CHIDDF:=0;
 VARS1.REZDI7.ID:=10025; VARS1.REZDI7.CHID:=0; VARS1.REZDI7.CHIDDF:=0;
 VARS1.REZDI8.ID:=10026; VARS1.REZDI8.CHID:=0; VARS1.REZDI8.CHIDDF:=0;
 VARS1.REZDI9.ID:=10027; VARS1.REZDI9.CHID:=0; VARS1.REZDI9.CHIDDF:=0;
 VARS1.REZDI10.ID:=10028; VARS1.REZDI10.CHID:=0; VARS1.REZDI10.CHIDDF:=0;
 VARS1.REZDI11.ID:=10029; VARS1.REZDI11.CHID:=0; VARS1.REZDI11.CHIDDF:=0;
 VARS1.REZDI12.ID:=10030; VARS1.REZDI12.CHID:=0; VARS1.REZDI12.CHIDDF:=0;
 VARS1.REZDI13.ID:=10031; VARS1.REZDI13.CHID:=0; VARS1.REZDI13.CHIDDF:=0;
 VARS1.REZDI14.ID:=10032; VARS1.REZDI14.CHID:=0; VARS1.REZDI14.CHIDDF:=0;
 VARS1.T1_VSLD_COPN.ID:=20001;VARS1.T1_VSLD_COPN.CHID:=2;
 VARS1.T1_VSLD_COPN.CHIDDF:=2;
 VARS1.T1_VSUNL_COPN.ID:=20002;VARS1.T1_VSUNL_COPN.CHID:=3;
 VARS1.T1_VSUNL_COPN.CHIDDF:=3;
 VARS1.T2_VSLD_COPN.ID:=20003;VARS1.T2_VSLD_COPN.CHID:=4;
 VARS1.T2_VSLD_COPN.CHIDDF:=4;

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 115 |

VARS1.T2_VSUNL_COPN.ID:=20004;VARS1.T2_VSUNL_COPN.CHID:=5;
 VARS1.T2_VSUNL_COPN.CHIDDF:=5;
 VARS1.D1_VSLD_COPN.ID:=20005;VARS1.D1_VSLD_COPN.CHID:=6;
 VARS1.D1_VSLD_COPN.CHIDDF:=6;
 VARS1.D2_VSLD_COPN.ID:=20006;VARS1.D2_VSLD_COPN.CHID:=7;
 VARS1.D2_VSLD_COPN.CHIDDF:=7;
 VARS1.D1_VSUNL_COPN.ID:=20007;VARS1.D1_VSUNL_COPN.CHID:=8;
 VARS1.D1_VSUNL_COPN.CHIDDF:=8;
 VARS1.D2_VSUNL_COPN.ID:=20008;VARS1.D2_VSUNL_COPN.CHID:=9;
 VARS1.D2_VSUNL_COPN.CHIDDF:=9;
 VARS1.P1_VST1T2_COPN.ID:=20009;VARS1.P1_VST1T2_COPN.CHID:=10;
 VARS1.P1_VST1T2_COPN.CHIDDF:=10;
 VARS1.T1_GSMIX_CSTRT.ID:=20010;VARS1.T1_GSMIX_CSTRT.CHID:=11;
 VARS1.T1_GSMIX_CSTRT.CHIDDF:=11;
 VARS1.T2_GSMIX_CSTRT.ID:=20011;VARS1.T2_GSMIX_CSTRT.CHID:=12;
 VARS1.T2_GSMIX_CSTRT.CHIDDF:=12;
 VARS1.REZDO1.ID:=20012; VARS1.REZDO1.CHID:=0; VARS1.REZDO1.CHIDDF:=0;
 VARS1.REZDO2.ID:=20013; VARS1.REZDO2.CHID:=0; VARS1.REZDO2.CHIDDF:=0;
 VARS1.REZDO3.ID:=20014; VARS1.REZDO3.CHID:=0; VARS1.REZDO3.CHIDDF:=0;
 VARS1.REZDO4.ID:=20015; VARS1.REZDO4.CHID:=0; VARS1.REZDO4.CHIDDF:=0;
 VARS1.REZDO5.ID:=20016; VARS1.REZDO5.CHID:=0; VARS1.REZDO5.CHIDDF:=0;
 VARS1.REZDO6.ID:=20017; VARS1.REZDO6.CHID:=0; VARS1.REZDO6.CHIDDF:=0;
 VARS1.REZDO7.ID:=20018; VARS1.REZDO7.CHID:=0; VARS1.REZDO7.CHIDDF:=0;
 VARS1.REZDO8.ID:=20019; VARS1.REZDO8.CHID:=0; VARS1.REZDO8.CHIDDF:=0;
 VARS1.REZDO9.ID:=20020; VARS1.REZDO9.CHID:=0; VARS1.REZDO9.CHIDDF:=0;
 VARS1.REZDO10.ID:=20021;VARS1.REZDO10.CHID:=0; VARS1.REZDO10.CHIDDF:=0;
 VARS1.REZDO11.ID:=20022;VARS1.REZDO11.CHID:=0; VARS1.REZDO11.CHIDDF:=0;
 VARS1.REZDO12.ID:=20023;VARS1.REZDO12.CHID:=0; VARS1.REZDO12.CHIDDF:=0;
 VARS1.REZDO13.ID:=20024;VARS1.REZDO13.CHID:=0; VARS1.REZDO13.CHIDDF:=0;
 VARS1.REZDO14.ID:=20025;VARS1.REZDO14.CHID:=0; VARS1.REZDO14.CHIDDF:=0;
 VARS1.REZDO15.ID:=20026;VARS1.REZDO15.CHID:=0; VARS1.REZDO15.CHIDDF:=0;
 VARS1.REZDO16.ID:=20027;VARS1.REZDO16.CHID:=0; VARS1.REZDO16.CHIDDF:=0;
 VARS1.REZDO17.ID:=20028;VARS1.REZDO17.CHID:=0; VARS1.REZDO17.CHIDDF:=0;
 VARS1.REZDO18.ID:=20029;VARS1.REZDO18.CHID:=0; VARS1.REZDO18.CHIDDF:=0;
 VARS1.REZDO19.ID:=20030;VARS1.REZDO19.CHID:=0; VARS1.REZDO19.CHIDDF:=0;
 VARS1.REZDO20.ID:=20031;VARS1.REZDO20.CHID:=0; VARS1.REZDO20.CHIDDF:=0;
 VARS1.REZDO21.ID:=20032;VARS1.REZDO21.CHID:=0; VARS1.REZDO21.CHIDDF:=0;

VLVDS1.VNabor_T1.ID:=1001;
 VLVDS1.VNabor_T2.ID:=1002;
 VLVDS1.VSliiv_T1.ID:=1003;
 VLVDS1.VSliiv_T2.ID:=1004;
 VLVDS1.VNabor_D1.ID:=1005;
 VLVDS1.VNabor_D2.ID:=1006;
 VLVDS1.VSliiv_D1.ID:=1007;
 VLVDS1.VSliiv_D2.ID:=1008;
 VLVDS1.VDoz_T1toT2.ID:=1009;

VLVDS1.VNabor_T1.CLSID:=16#2011;
 VLVDS1.VNabor_T2.CLSID:=16#2011;
 VLVDS1.VSliiv_T1.CLSID:=16#2011;
 VLVDS1.VSliiv_T2.CLSID:=16#2011;
 VLVDS1.VNabor_D1.CLSID:=16#2011;
 VLVDS1.VNabor_D2.CLSID:=16#2011;
 VLVDS1.VSliiv_D1.CLSID:=16#2011;
 VLVDS1.VSliiv_D2.CLSID:=16#2011;
 VLVDS1.VDoz_T1toT2.CLSID:=16#2011;

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Лист | № докум. | Підпис | Дата |

4.6. Основна програма

Тоді відповідно до вимог розгортання каркасу головна функція Main приймає

ВИГЛЯД:

```
PLCFN (START:=TRUE, PLC=>PLC);
(*виклик при старті*)
IF PLC.STA1.SCN1 THEN
  Plcmaps1 (PLCCFG := PLC, MODULES := MODULS);
  INITVARS1 (VARS1 := ALL_VARS, VLVDS1 := ALL_VLVDS); (*ініціалізація змінних при
  старті*)
END_IF;
(*обробка входів*)
Dichs1 (CHCFG := CHDI, CHHMI := CHDI_HMI, PLCCFG := PLC, CH_BUF1 := CH_BUF1);
Divars1 (VARS1 := ALL_VARS, DIH1 := DIH1, DIVARCFG := DIVAR_CFG1, DIVARHMI :=
DIVAR_HMI1, VARBUF1 := VARBUF1, PLCCFG := PLC, CHDI := CHDI);
Aichs1 (CHCFG := CHAI, CHHMI := CHAI_HMI, PLCCFG := PLC, CH_BUF1 := CH_BUF1);
Aivars1 (VARS1 := ALL_VARS, AIH1 := AIH1, AIVARCFG := AIVAR_CFG1, AIVARHMI :=
AIVAR_HMI1, VARBUF1 := VARBUF1, PLCCFG := PLC, CHAI := CHAI);
(*обробка виконавчих механізмів*)
Resolution1 (ACT1 := ACT1);
Actrs1 (ACT1 := ACT1, ACTH1 := ACTH1, VARS1 := ALL_VARS, ACTBUF := ACTBUF, PLCCFG :=
PLC);
(*обробка виходів*)
Dochs1 (CHCFG := CHDO, CHHMI := CHDO_HMI, PLCCFG := PLC, CH_BUF1 := CH_BUF1, RAWs =>
dochs_RAWs);
Dovars1 (VARS1 := ALL_VARS, DOH1 := DOH1, DOVARCFG := DOVAR_CFG1, DOVARHMI :=
DOVAR_HMI1, VARBUF1 := VARBUF1, PLCCFG := PLC, CHDO := CHDO);
Aochs1 (CHCFG := CHAO, CHHMI := CHAO_HMI, PLCCFG := PLC, CH_BUF1 := CH_BUF1, RAWs =>
aochs_RAWs);
Aovars1 (VARS1 := ALL_VARS, AOH1 := AOH1, AOVARCFG := AOVAR_CFG1, AOVARHMI :=
AOVAR_HMI1, VARBUF1 := VARBUF1, PLCCFG := PLC, CHAO := CHAO);
(*обробка модулів плк*)
Moduls1 (CHAI:=CHAI, CHAI_HMI:=CHAI_HMI, CHAO:=CHAO, CHAO_HMI:=CHAO_HMI, CHDI:=CHDI,
CHDI_HMI:=CHDI_HMI,
CHDO:=CHDO, CHDO_HMI:=CHDO_HMI, MODULES:=MODULS, PLC:=PLC, SUBMODULE1:=SUBMODULE1);
```

4.7. Перевірка роботи з використанням людино-машинного інтерфейсу

Для перевірки роботи каркасу було розроблено НМІ рівня технологічних змінних, для аналогових та дискретних вхідних змінних. Для демонстрації роботи каркасу запускається об'єкт у режимі симуляції та проводяться наступні дії:

- 1) відкрити клапани набору та клапани для нагріву кожуху танків T1 та T2.
- 2) на НМІ для аналогових вхідних змінних виставити бажані налаштування змінних (див. рис. 4.8).
- 3) при перевищенні меж конфігурації алармів спрацьовує сповіщення у вигляді лампочки (для Lo та Ni - жовтим, для LoLo та NiNi - червоним), див. рис.4.8

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 117 |

4) виставити межі максимуму та мінімуму для маштабованого та немаштабованого значення, гістерезис в реальних одиницях або відсотках і перевірити реакцію плинного значення

5) перевірити роботу режимів технологічних змінних (FRC - режим форсування, BAD - дані недостовірні або відбувся обрив каналу, SML - змінна в режимі симуляції).

simulator.hmismulator

| | | | | | | | |
|--------|-----------|--------------|---|--|---|--|-------|
| T1_LT1 | рівень T1 | 23513 | Lo- <input checked="" type="checkbox"/> 0 | Hi- <input checked="" type="checkbox"/> 5000 | LoLo- <input checked="" type="checkbox"/> 0 | HiHi- <input checked="" type="checkbox"/> 9000 | |
| Lo | Hi | LoLo | HiHi | Status value - | BAD | SML | FRC |
| HYST | 0 | Raw limits - | Max 10000 | Min 0 | End limits - | Max 100 | Min 0 |

| | | | | | | | |
|--------|-----------|--------------|---|--|---|---|-------|
| T2_LT1 | рівень T2 | 22938 | Lo- <input checked="" type="checkbox"/> 0 | Hi- <input checked="" type="checkbox"/> 3500 | LoLo- <input checked="" type="checkbox"/> 0 | HiHi- <input checked="" type="checkbox"/> 29999 | |
| Lo | Hi | LoLo | HiHi | Status value - | BAD | SML | FRC |
| HYST | 0 | Raw limits - | Max 10000 | Min 0 | End limits - | Max 100 | Min 0 |

| | | | | | | | |
|--------|------------------|--------------|--|--|--|--|-------|
| T1_TT1 | температура в T1 | 2374 | Lo- <input checked="" type="checkbox"/> 1600 | Hi- <input checked="" type="checkbox"/> 2100 | LoLo- <input checked="" type="checkbox"/> 1400 | HiHi- <input checked="" type="checkbox"/> 2300 | |
| Lo | Hi | LoLo | HiHi | Status value - | BAD | SML | FRC |
| HYST | 0 | Raw limits - | Max 10000 | Min 0 | End limits - | Max 100 | Min 0 |

| | | | | | | | |
|--------|------------------|--------------|--|--|--|--|-------|
| T2_TT1 | температура в T2 | 2065 | Lo- <input checked="" type="checkbox"/> 2100 | Hi- <input checked="" type="checkbox"/> 2200 | LoLo- <input checked="" type="checkbox"/> 2000 | HiHi- <input checked="" type="checkbox"/> 2500 | |
| Lo | Hi | LoLo | HiHi | Status value - | BAD | SML | FRC |
| HYST | 0 | Raw limits - | Max 10000 | Min 0 | End limits - | Max 100 | Min 0 |

OBJECT DIVARs_1 DIVARs_2 DIVARs_3

Рисунок 4.8 - Реалізація використання функцій аналогових вхідних змінних на НМІ для середовище розробки PLCnext Engineer

У подальшому НМІ буде доповнюватись іншим функціоналом каркасу.

На рис.4.9-4.11 показані операторські екрани на яких можна перевірити роботу інших рівнів каркасу.

simulator.hmisimulator

| | | | | | |
|--------------------------------|---|--|--|---------------------------|-----------------------|
| T1_VSLD_SCLS | клапан набору T1 закритий | Status value - BAD | <input type="radio"/> SML | <input type="radio"/> FRC | <input type="radio"/> |
| <input type="text" value="0"/> | Як аварійний - <input checked="" type="checkbox"/> | Інвертувати значення - <input type="checkbox"/> | Попередження <input type="radio"/> Аварія <input checked="" type="radio"/> | | |
| | Як попереджений - <input checked="" type="checkbox"/> | Значення норми - <input type="checkbox"/> | | | |
| T2_VSLD_SCL | клапан набору T2 закритий | Status value - BAD | <input type="radio"/> SML | <input type="radio"/> FRC | <input type="radio"/> |
| <input type="text" value="0"/> | Як аварійний - <input type="checkbox"/> | Інвертувати значення - <input checked="" type="checkbox"/> | Попередження <input type="radio"/> Аварія <input type="radio"/> | | |
| | Як попереджений - <input type="checkbox"/> | Значення норми - <input type="checkbox"/> | | | |
| T1_VSUNL_SCLS | клапан зливу T1 закритий | Status value - BAD | <input type="radio"/> SML | <input type="radio"/> FRC | <input type="radio"/> |
| <input type="text" value="1"/> | Як аварійний - <input checked="" type="checkbox"/> | Інвертувати значення - <input type="checkbox"/> | Попередження <input type="radio"/> Аварія <input checked="" type="radio"/> | | |
| | Як попереджений - <input checked="" type="checkbox"/> | Значення норми - <input type="checkbox"/> | | | |
| T2_VSUNL_SCLS | клапан зливу T2 закритий | Status value - BAD | <input type="radio"/> SML | <input type="radio"/> FRC | <input type="radio"/> |
| <input type="text" value="1"/> | Як аварійний - <input type="checkbox"/> | Інвертувати значення - <input checked="" type="checkbox"/> | Попередження <input type="radio"/> Аварія <input type="radio"/> | | |
| | Як попереджений - <input type="checkbox"/> | Значення норми - <input type="checkbox"/> | | | |
| D1_VSLD_SCLS | клапан набору D1 закритий | Status value - BAD | <input type="radio"/> SML | <input type="radio"/> FRC | <input type="radio"/> |
| <input type="text" value="1"/> | Як аварійний - <input type="checkbox"/> | Інвертувати значення - <input type="checkbox"/> | Попередження <input type="radio"/> Аварія <input type="radio"/> | | |
| | Як попереджений - <input type="checkbox"/> | Значення норми - <input checked="" type="checkbox"/> | | | |
| D2_VSLD_SCLS | клапан набору D2 закритий | Status value - BAD | <input type="radio"/> SML | <input type="radio"/> FRC | <input type="radio"/> |
| <input type="text" value="1"/> | Як аварійний - <input type="checkbox"/> | Інвертувати значення - <input type="checkbox"/> | Попередження <input type="radio"/> Аварія <input type="radio"/> | | |
| | Як попереджений - <input checked="" type="checkbox"/> | Значення норми - <input checked="" type="checkbox"/> | | | |
| D1_VSUNL_SCLS | клапан зливу D1 закритий | Status value - BAD | <input type="radio"/> SML | <input type="radio"/> FRC | <input type="radio"/> |
| <input type="text" value="1"/> | Як аварійний - <input checked="" type="checkbox"/> | Інвертувати значення - <input type="checkbox"/> | Попередження <input type="radio"/> Аварія <input type="radio"/> | | |
| | Як попереджений - <input type="checkbox"/> | Значення норми - <input type="checkbox"/> | | | |

OBJECT AIVARs DIVARs_2 DIVARs_3

Рисунок 4.9 - Реалізація використання функцій дискретних вхідних змінних на HMI для середовище розробки PLCnext Engineer

simulator.hmisimulator

| | | | | | |
|--------------------------------|---|--|--|--------------------------------------|-----------------------|
| D2_VSUNL_SCLS | клапан зливу D2 закритий | Status value - BAD | <input type="radio"/> SML | <input type="radio"/> FRC | <input type="radio"/> |
| <input type="text" value="1"/> | Як аварійний - <input checked="" type="checkbox"/> | Інвертувати значення - <input type="checkbox"/> | Попередження <input type="radio"/> Аварія <input type="radio"/> | | |
| | Як попереджений - <input checked="" type="checkbox"/> | Значення норми - <input checked="" type="checkbox"/> | | | |
| D1_LSH | сигналізатор верхнього рівня D1 | Status value - BAD | <input type="radio"/> SML | <input checked="" type="radio"/> FRC | <input type="radio"/> |
| <input type="text" value="0"/> | Як аварійний - <input type="checkbox"/> | Інвертувати значення - <input checked="" type="checkbox"/> | Попередження <input type="radio"/> Аварія <input type="radio"/> | | |
| | Як попереджений - <input type="checkbox"/> | Значення норми - <input checked="" type="checkbox"/> | | | |
| D1_LSL | сигналізатор нижнього рівня D1 | Status value - BAD | <input type="radio"/> SML | <input checked="" type="radio"/> FRC | <input type="radio"/> |
| <input type="text" value="0"/> | Як аварійний - <input checked="" type="checkbox"/> | Інвертувати значення - <input type="checkbox"/> | Попередження <input type="radio"/> Аварія <input checked="" type="radio"/> | | |
| | Як попереджений - <input checked="" type="checkbox"/> | Значення норми - <input type="checkbox"/> | | | |
| D2_LSH | сигналізатор верхнього рівня D2 | Status value - BAD | <input type="radio"/> SML | <input checked="" type="radio"/> FRC | <input type="radio"/> |
| <input type="text" value="0"/> | Як аварійний - <input checked="" type="checkbox"/> | Інвертувати значення - <input type="checkbox"/> | Попередження <input type="radio"/> Аварія <input type="radio"/> | | |
| | Як попереджений - <input type="checkbox"/> | Значення норми - <input checked="" type="checkbox"/> | | | |
| D2_LSL | сигналізатор нижнього рівня D2 | Status value - BAD | <input type="radio"/> SML | <input checked="" type="radio"/> FRC | <input type="radio"/> |
| <input type="text" value="0"/> | Як аварійний - <input type="checkbox"/> | Інвертувати значення - <input type="checkbox"/> | Попередження <input type="radio"/> Аварія <input type="radio"/> | | |
| | Як попереджений - <input checked="" type="checkbox"/> | Значення норми - <input checked="" type="checkbox"/> | | | |
| P1_VST1T2_SCLS | 3-ходовий клапан в позиції T1 | Status value - BAD | <input type="radio"/> SML | <input type="radio"/> FRC | <input type="radio"/> |
| <input type="text" value="1"/> | Як аварійний - <input type="checkbox"/> | Інвертувати значення - <input checked="" type="checkbox"/> | Попередження <input type="radio"/> Аварія <input type="radio"/> | | |
| | Як попереджений - <input type="checkbox"/> | Значення норми - <input type="checkbox"/> | | | |
| P1_VST1T2_SOPN | 3-ходовий клапан в позиції T2 | Status value - BAD | <input type="radio"/> SML | <input type="radio"/> FRC | <input type="radio"/> |
| <input type="text" value="0"/> | Як аварійний - <input type="checkbox"/> | Інвертувати значення - <input type="checkbox"/> | Попередження <input type="radio"/> Аварія <input type="radio"/> | | |
| | Як попереджений - <input type="checkbox"/> | Значення норми - <input checked="" type="checkbox"/> | | | |

OBJECT AIVARs DIVARs_2 DIVARs_3

Рисунок 4.10 - Реалізація використання функцій дискретних вхідних змінних на HMI для середовище розробки PLCnext Engineer

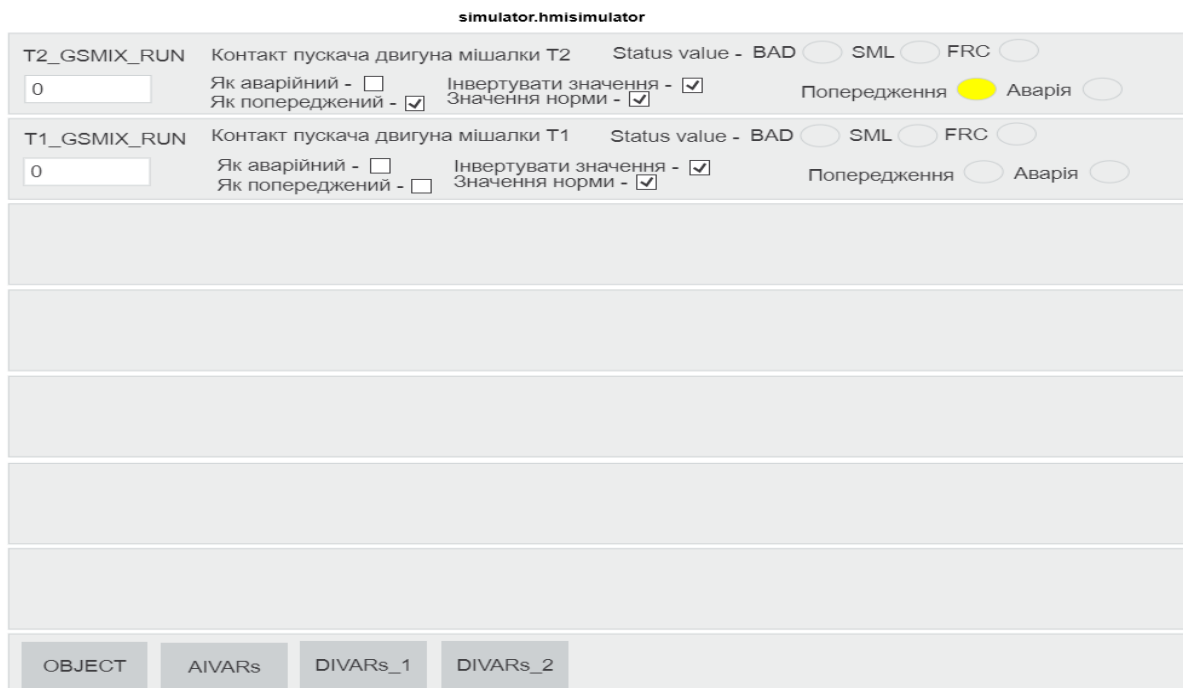


Рисунок 4.11 - Реалізація використання функцій дискретних вхідних змінних на HMI для середовище розробки PLCnext Engineer

4.8. Перевірка роботи ПоТ

Для перевірки роботи ПоТ відкриємо на деякий час клапан на подачу речовини в перший танк, тоді температура кожуха буде дорівнювати 1817, а рівень в танку прийме значення - 17550 (Рис. 4.12).

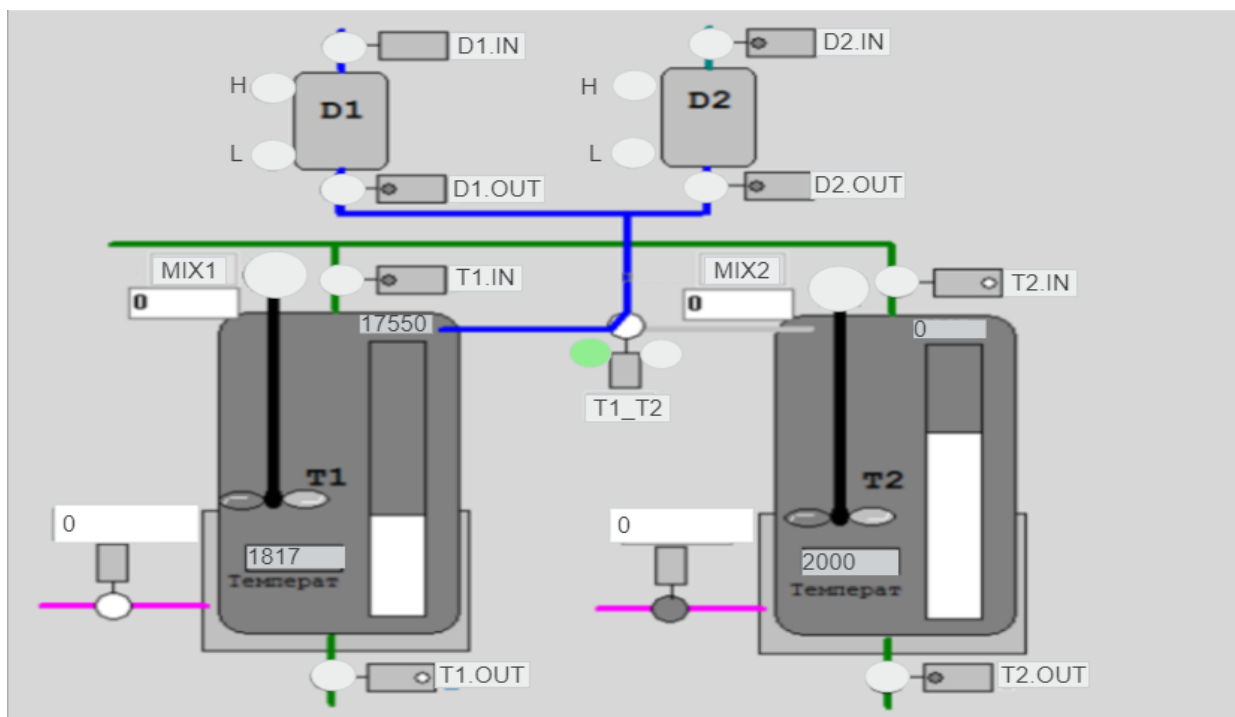


Рисунок 4.12 - Стан системи для перевірки роботи ПоТ

Додатково перевіримо відправляемі значення через OPC UA. Використаємо UaExpert та звіримо значення (Рис. 4.13).

| # | Server | Node Id | Display Name | Value | Datatype | Source Timestamp | Server Timestamp | Statuscode |
|----|------------------|--|--------------|-------|----------|------------------|------------------|------------|
| 1 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_LT1.VAL | VAL | 17550 | Float | 13:13:53.073 | 13:13:53.073 | Good |
| 2 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_LT1.VALPRCSTA2 | VALPRCSTA2 | 25600 | UInt16 | 13:12:36.073 | 13:12:36.074 | Good |
| 3 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_STA1_STA_ALDIS | STA_ALDIS | false | Boolean | 12:14:27.818 | 12:14:27.818 | Good |
| 4 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_STA1_STA_ALM | STA_ALM | false | Boolean | 12:14:27.818 | 12:14:27.818 | Good |
| 5 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_STA1_STA_BAD | STA_BAD | false | Boolean | 12:14:27.818 | 12:14:27.818 | Good |
| 6 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_STA1_STA_BRK | STA_BRK | false | Boolean | 12:14:27.819 | 12:14:27.819 | Good |
| 7 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_STA1_STA_CMDLOAD | STA_CMDLOAD | false | Boolean | 12:14:27.819 | 12:14:27.819 | Good |
| 8 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_STA1_STA_DLNK | STA_DLNK | true | Boolean | 12:14:27.819 | 12:14:27.819 | Good |
| 9 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_STA1_STA_ENBL | STA_ENBL | true | Boolean | 12:14:27.819 | 12:14:27.819 | Good |
| 10 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_STA1_STA_FRC | STA_FRC | false | Boolean | 12:14:27.820 | 12:14:27.820 | Good |
| 11 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_STA1_STA_HI | STA_HI | false | Boolean | 12:14:27.820 | 12:14:27.820 | Good |
| 12 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_STA1_STA_HIH | STA_HIH | false | Boolean | 12:14:27.820 | 12:14:27.820 | Good |
| 13 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_STA1_STA_INBUF | STA_INBUF | false | Boolean | 12:14:27.820 | 12:14:27.820 | Good |
| 14 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_STA1_STA_LO | STA_LO | false | Boolean | 12:14:27.821 | 12:14:27.821 | Good |
| 15 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_STA1_STA_LOLO | STA_LOLO | false | Boolean | 12:14:27.821 | 12:14:27.821 | Good |
| 16 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_STA1_STA_OVRD | STA_OVRD | false | Boolean | 12:14:27.821 | 12:14:27.821 | Good |
| 17 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_STA1_STA_SML | STA_SML | true | Boolean | 12:14:27.821 | 12:14:27.821 | Good |
| 18 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_STA1_STA_WRN | STA_WRN | false | Boolean | 12:14:27.822 | 12:14:27.822 | Good |
| 19 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1.VAL | VAL | 1817 | Float | 13:16:05.780 | 13:16:05.780 | Good |
| 20 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1.VALPRCSTA2 | VALPRCSTA2 | 25600 | UInt16 | 13:16:05.780 | 13:16:05.780 | Good |
| 21 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1_STA1_STA_ALDIS | STA_ALDIS | false | Boolean | 13:16:05.781 | 13:16:05.781 | Good |
| 22 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1_STA1_STA_ALM | STA_ALM | false | Boolean | 13:16:05.781 | 13:16:05.781 | Good |
| 23 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1_STA1_STA_BAD | STA_BAD | false | Boolean | 13:16:05.781 | 13:16:05.781 | Good |
| 24 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1_STA1_STA_BRK | STA_BRK | false | Boolean | 13:16:05.782 | 13:16:05.782 | Good |
| 25 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1_STA1_STA_CMDLOAD | STA_CMDLOAD | false | Boolean | 13:16:05.782 | 13:16:05.782 | Good |
| 26 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1_STA1_STA_DLNK | STA_DLNK | true | Boolean | 13:16:05.782 | 13:16:05.782 | Good |
| 27 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1_STA1_STA_ENBL | STA_ENBL | true | Boolean | 13:16:05.783 | 13:16:05.783 | Good |
| 28 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1_STA1_STA_FRC | STA_FRC | false | Boolean | 13:16:05.784 | 13:16:05.784 | Good |
| 29 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1_STA1_STA_HI | STA_HI | false | Boolean | 13:16:05.784 | 13:16:05.784 | Good |
| 30 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1_STA1_STA_HIH | STA_HIH | false | Boolean | 13:16:05.785 | 13:16:05.785 | Good |
| 31 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1_STA1_STA_INBUF | STA_INBUF | false | Boolean | 13:16:05.785 | 13:16:05.785 | Good |
| 32 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1_STA1_STA_LO | STA_LO | false | Boolean | 13:16:05.785 | 13:16:05.785 | Good |
| 33 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1_STA1_STA_LOLO | STA_LOLO | false | Boolean | 13:16:05.786 | 13:16:05.786 | Good |
| 34 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1_STA1_STA_OVRD | STA_OVRD | false | Boolean | 13:16:05.786 | 13:16:05.786 | Good |
| 35 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1_STA1_STA_SML | STA_SML | true | Boolean | 13:16:05.786 | 13:16:05.786 | Good |
| 36 | eUAServer@axc... | N5S[String]Arp.Plc.Eclr/Main1.AIH1.T1_TT1_STA1_STA_WRN | STA_WRN | false | Boolean | 13:16:05.786 | 13:16:05.786 | Good |

Рисунок 4.13 - Змінні T1_TT1 та T1_LT1 в UaExpert

Для збереження отриманих даних у Node-Red було написано функцію (Рис. 4.14), яка зберігає дані у глобальній контекст Node-Red. Створено ноду OpcUa-Client та в її налаштуваннях виставлено дію Subscribe (Рис. 4.15). Дія Subscribe підписується на вибрані вузли і говорить серверу стежити за цими елементами.

Изменить узел function

Удалить Отмена Готово

Свойства

Имя:

Setup Настройка **Функция** Закрытие

```

1 let itemsarray = global.get ('itemsarray') || [];
2 let itemsob = global.get ('itemsob') || {};
3
4 if (!msg.payload.dataValue) return
5 let itemname = msg.payload.nodeId.toString().replace("ns=5;s=Arp.Plc.Ec
6 let item = itemsob;
7 let path = itemname.split('.');
8 for (let levelname of path) {
9     if (!item[levelname]) item[levelname]={};
10    item = item[levelname]
11 }
12 item.datatype = msg.datatype;
13 item.nodeId= msg.payload.nodeId.toString();
14 item.name = itemname;
15 item.value = msg.payload.dataValue.value.value;
16
17 itemsarray.push(item);
18 global.set ('itemsarray', itemsarray);
19 global.set ('itemsob', itemsob);
20 msg.topic = item.nodeId;
21 msg.payload = "2000";
22 msg.itemname = itemname;
23
24 return msg;

```

Рис. 4.14. Функція для запису отриманих даних у глобальний контекст

Изменить узел OpсUa-Client

Удалить Отмена Готово

Свойства

Endpoint:

Action:

Interval:

Certificate:

Local certificate file with absolute path:

Local private key file with absolute path:

PKI certificate folder:

Name:

Рис. 4.15. Налаштування ноди OpсUa-Client

Відкривши глобальний контекст у Node-Red та знайшовши змінні, які були обрані для перевірки можна спостерігати, що змінні успішно відправлені та записані (Рис. 4.16).

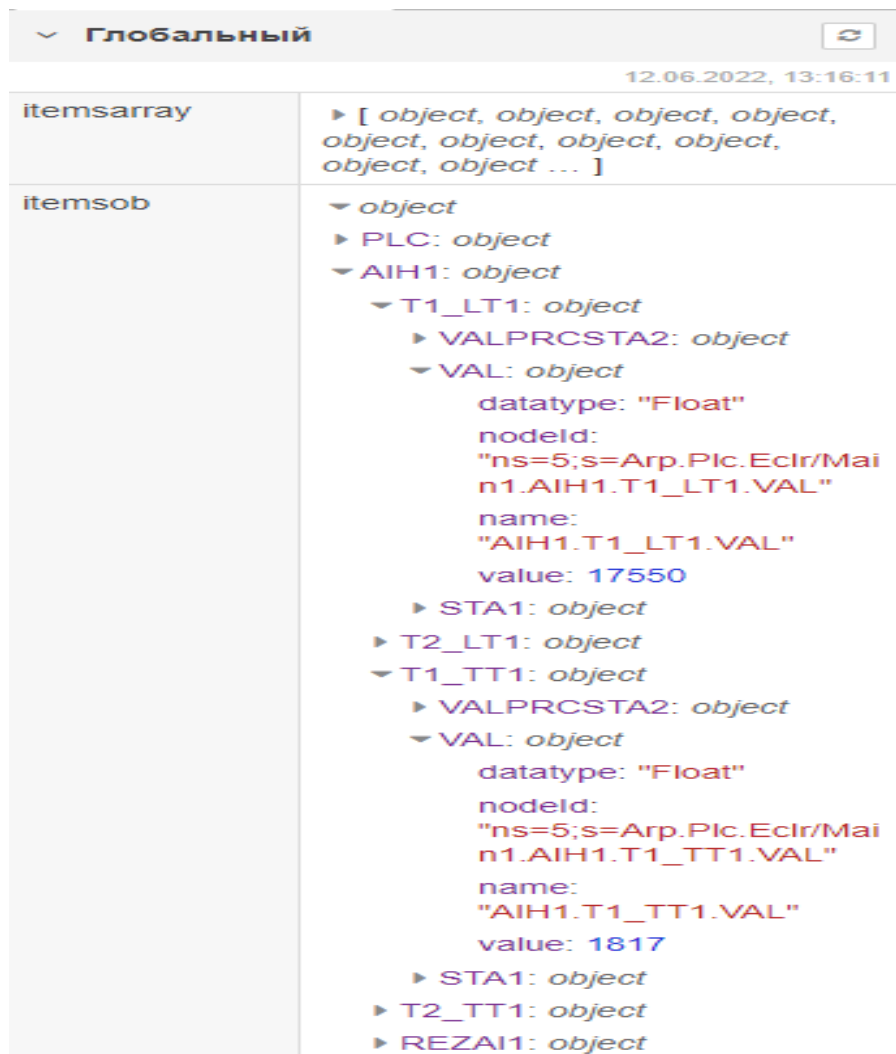


Рис. 4.16. Глобальний контекст Node-Red

5. ОХОРОНА ПРАЦІ

Охорона праці – це система законодавчих актів, соціально-економічних, організаційних, технічних, гігієнічних та лікувально-профілактичних заходів і засобів, які забезпечують безпеку, збереження здоров'я та працездатності людини у процесі праці. У час, коли виробництво все більше розвивається, охорона праці набуває все більшого значення.

Тема дипломного проекту - “Розробка IoT рішень на базі PLCnext”. Розроблення проекту буде використовуватись на виробництві та спростить роботу для персоналу.

Технічні засоби системи автоматизації розміщуються: на технологічному обладнанні та біля нього – засоби відбору та отримання технологічної інформації, нормуючі перетворювачі (давачі, первинні перетворювачі), виконавчі механізми,

місцеві органи керування; розроблений проект взаємодіє поряд з автоматизованою системою для відображення інформації та оперативного керування на НМІ.

В цьому розділі запропоновані технічні рішення та організаційні заходи з безпечної експлуатації засобів автоматизації.

5.1. Технічні рішення та організаційні заходи з безпечної експлуатації засобів автоматизації, технологічного обладнання та засобів обчислювальної техніки.

Так як характер праці оператора монотонний, стомлюючий і він отримує величезний потік інформації, то найменші помилки в його діях можуть призвести до важких аварій, пожеж та нещасних випадків. Тому робота проходить за встановленим графіком, позмінно. Протяжність робочої зміни обмежено 8 годинами.

Вагома частина уваги приділяється розташуванню кнопок, органів керування, перемикачів. Вони розташовані на визначеній висоті від підлоги. Стіни приміщення пофарбовані в неяскравий, ясно-сірий колір, що не розсіює уваги оператора. Підлога рівна, неслизька. Висота приміщення - 4 м.

Сходи та майданчики обгороджені перилами висотою не менше 1м. Вони розроблені з листової просіченої сталі товщиною 5 мм. Навколо котла споруджені майданчики обслуговування на відмітках 3, 6, 9 м і вище через кожні 3 м.

Для контролю за параметрами технологічного устаткування встановлено датчик температури, тиску, рівня що встановлюються в трубопроводі, датчики рівня, монтуються біля трубопроводів і вимірюють тиск з відборів з трубопроводу та датчик рівня, що вимірює різницю тисків.

Засувки і вентилі, для відкривання яких потрібно велике зусилля, забезпечені обвідними лініями і механічними чи електричними приводами, а всі двигуни керуються частотними перетворювачами, які стоять біля двигунів. Всі наявні

гарячі частини устаткування, трубопроводи, баки, при доторканні до яких може виникнути опік, мають теплову ізоляцію. Матеріал ізоляції підбирається, щоб температура її поверхні не перевищувала 45 °С при температурі навколишнього повітря 25 °С .

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Лист | № докум. | Підпис | Дата |

Для забезпечення безпечної і зручної експлуатації засобів автоматизації проводять наступні вимоги:

- передбачені безпечні, зручні по конструкції і розмірам проходи й обладнання для ведення робіт - робочі площадки (ширина вільного проходу площадок - 0,8 м);
- виконавчі механізми розміщені на паро- і водопроводах, де заборонений випадковий доступ людини і захищені захисною сіткою, що забезпечує безпечну експлуатацію обладнання;
- аварійні органи керування (кнопки і вимикачі) – червоного кольору, відрізняються від інших зовнішнім виглядом та мимовільне відключення їх неможливе.

При монтажі щита планується спеціальний прохід між стіною і відкритими дверми відстанню не менш 0,8 м, для його експлуатації.

1.1.1. Електробезпека

Основні нормативні документи, за якими регламентуються вимоги з електробезпеки у виробничих приміщеннях, це ПУЕ-2017, НПАОП 40.1-1.21- 98, НПАОП 40.1-1.01-97, НПАОП-40.1-1.32-01 та Технічний регламент з безпеки низьковольтного електричного обладнання. Діюча на основі цих нормативних документів система організаційних і технічних заходів і засобів з електробезпеки забезпечує захист людей від шкідливої і небезпечної дії електричного струму, електричної дуги, електромагнітного поля і статичної електрики.

Згідно ДСТУ 7237:2011 для забезпечення захисту від випадкового дотику людини до струмовідних частин ЕУ застосовують такі види захисту:

- основне (робоче) ізолювання струмовідних частин (захисне ізолювання);
- додаткове, посилене, подвійне ізолювання струмовідних частин;
- захисні оболонки;
- захисні огорожі (тимчасові або стаціонарні);
- безпечне розташування струмовідних частин;

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 125 |

- ізолювання робочого місця;
- мала напруга;
- захисне вимкнення;
- попереджувальна сигналізація (звукова, світлова тощо);
- блокування;
- встановлення знаків безпеки;
- електрозахисні засоби; засоби індивідуального захисту.

Для запобігання ураженню електричним струмом під час дотику до металевих неструмовідних частин, які можуть бути під напругою внаслідок пошкодження ізоляції, застосовують окремо або в поєднанні такі види захисту:

- захисне заземлення;
- автоматичне вимкнення живлення;
- зрівнювання потенціалів;
- обладнання відповідного класу за електрозахистом;
- захисний електричний поділ кіл;
- ізолювальні (непровідні) приміщення, зони, майданчики;
- системи наднизької напруги (безпечної, захисної); вирівнювання потенціалів.

Приміщення операторської відноситься до сухих, так як відносна вологість не перевищує 60%.

За ступенем небезпеки ураження електричним струмом дане приміщення відноситься до приміщень без підвищеної небезпеки, тому що в ньому відсутні умови, які створюють підвищену або особливу небезпеку: підвищений рівень вологості, струмопровідний пил, хімічноактивне середовище, струмопровідна підлога, підвищена температура, можливість одночасного дотику до заземлених металоконструкцій будівель з одного боку та до струмопровідних частин електрообладнання з іншого боку. ЕОМ, периферійні пристрої, інше устаткування, електропроводи та кабелі мають апаратуру захисту від струму короткого замикання та інших аварійних режимів ЕУ.

Питання надійності електропостачання вирішується в залежності від категорії відповідальності електроприймачів. Так як в приміщенні є електроприймачі, у яких

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 126 |

порушення електропостачання призводять до виникнення небезпеки для життя людей, а також до порушення складного технологічного процесу (відповідає 1 категорії), тому для електроприймачів 1 категорії відповідальності повинно бути 2 джерела живлення, з автоматичним включенням резервного джерела при автоматичному вимиканні основного.

Розетки змонтовані на негорючих пластинах і мають сучасну трифазну конструкцію, захисний контакт виконаний у вигляді бічних контактів, які взаємодіють першими при включенні вилки в розетку й відключаються останніми при витягуванні вилки з розетки, що відповідає вимогам ПУЕ-2017.

Використання N-провідника як РЕ-провідника в системі захисного заземлення з підсистемою TN-S забороняється, а також неприпустиме підключення цих провідників на щитку до одного контактного затискача.

Для сприятливих умов роботи всі елементи електроприладів та устаткування виконані відповідно до вимог з електробезпеки, мають необхідне ізоляційне покриття (подвійна ізоляція) і властивості, що виключають можливість ураження електричним струмом при підключенні та експлуатації устаткування.

За способом захисту людини від ураження електричним струмом згідно ДСТУ ІЕС 61140:2015 ВДТ, ПЕОМ та периферійні пристрої відносяться до I, II та III класів за електрозахистом.

Висновок: вимоги техніки безпеки та охорони праці з експлуатації засобів автоматизації, технологічного обладнання і засобів обчислювальної техніки (ВДТ ПЕОМ) в приміщенні операторської відповідають діючим нормам.

5.2. Вимоги гігієни праці та виробничої санітарії при використанні засобів обчислювальної техніки

Умови праці осіб, які працюють з ВДТ ПЕОМ, повинні відповідати I або II класу згідно з Гігієнічною класифікацією праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу №4137-86, затвердженою МОЗ СРСР 12.08.86.

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 127 |

При організації робочих місць користувачів ВДТ ПЕОМ необхідно дотримуватись відповідних ергономічних вимог за наступними стандартами:

- ДСТУ ISO 9241-3-2001_Ергономічні вимоги до роботи з відеотерміналами в офісі. Частина 3. Вимоги до відеотерміналів (ISO 9241-31992, IDT)
- ДСТУ ISO 9241-5-2004_Ергономічні вимоги до роботи з відеотерміналами в офісі. Частина 5. Вимоги до компонування робочого місця та до робочої пози
- 6ДСТУ ISO 9241-6-2004 Ергономічні вимоги до роботи з відеотерміналами в офісі. Частина 6. Вимоги до робочого середовища
- ДСТУ ISO 9241-7-2004_Ергономічні вимоги до роботи з відеотерміналами в офісі. Частина 7. Вимоги до дисплеїв з відбитками

Шкідливими виробничими факторами, яким піддаються користувачі ПЕОМ та ВДТ є:

- м'яке рентгенівське випромінювання;
- ультрафіолетове та інфрачервоне випромінювання;
- наявність шуму та вібрації;
- електростатичне поле між екраном та оператором;
- наявність пилу, озону, оксидів азоту та аероіонізації.
- електромагнітне випромінювання.

1.1.2. Мікроклімат робочої зони

Нормовані параметри мікроклімату на робочих місцях: температура, відносна вологість та швидкість руху повітря повинні відповідати вимогам ДСН 3.3.6-042-99 «Санітарні норми мікроклімату виробничих приміщень», затверджених Міністерством охорони здоров'я та ДСанПіН 3.3.2.007-98. Оптимальні значення параметрів мікроклімату для приміщень з ВДТ та ПЕОМ наведені в Таблиця 5.

Таблиця 5.1 - Оптимальні норми мікроклімату для приміщень з ВДТ та ПЕОМ

| Пора року | Категорія робіт згідно з ГОСТ 12.1- | Температура повітря, °С | Відносна вологість повітря, % | Швидкість руху повітря, м/с |
|-----------|-------------------------------------|-------------------------|-------------------------------|-----------------------------|
| | | | | |

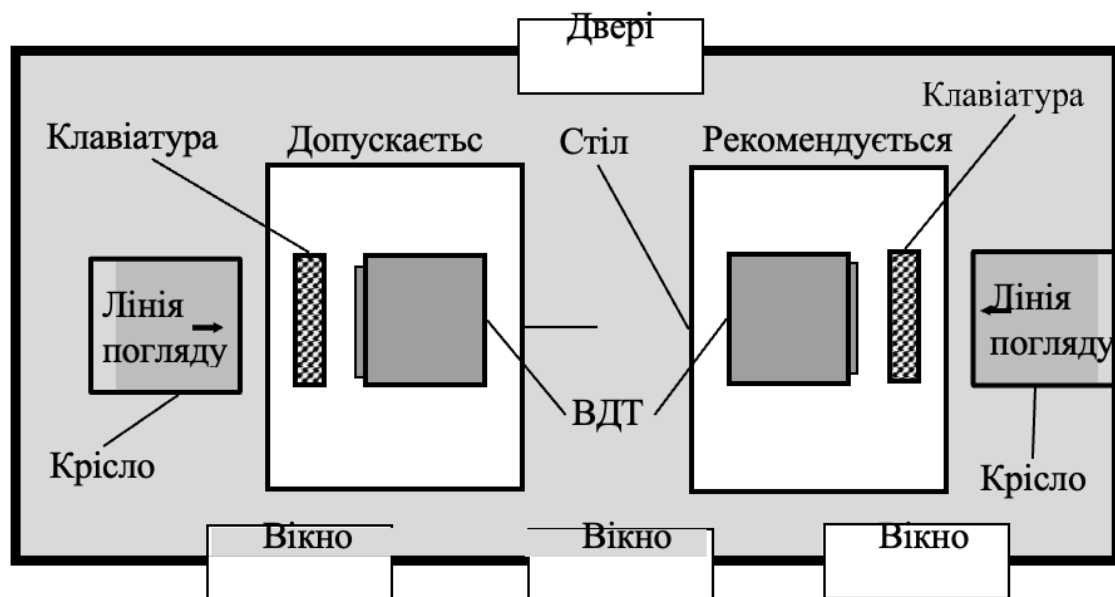


Рисунок 5.1 - Схема розміщення робочих місць відносно світлових отворів

Рівень освітленості на робочому столі в зоні розташування документів має бути в межах 300-500 лк. У разі неможливості забезпечити даний рівень освітленості системою загального освітлення допускається застосування світильників місцевого освітлення, але при цьому не повинно бути відблисків на поверхні екрану та збільшення освітленості екрану більше ніж до 300 лк.

4.2.4. Виробничі випромінювання

Рівні електромагнітного випромінювання та магнітних полів повинні відповідати вимогам ДСанПіН 3.3.2-007-98.

Таблиця 5.2 - Допустимі значення параметрів неіонізуючих електромагнітних випромінювань.

| | |
|---|--------------------|
| Напруженість електромагнітного поля по магнітній складовій на відстані 50 см від поверхні відеомонітора | 0,3 А/м |
| Напруженість електростатичного поля не повинна перевищувати: для дорослих користувачів для дітей дошкільних закладів та учнів середніх спеціальних і вищих навчальних закладів | 20 кВ/м 15 кВ/м |

| | |
|---|-------------------|
| Напруженість електромагнітного поля на відстані 50 см навколо ВДТ по електричній складовій повинна бути не більше: в діапазоні частот 5 Гц - 2 кГц; в діапазоні частот 2 - 400 кГц | 52 В/м 2,5 В/м |
| Густина магнітного потоку повинна бути не більше: в діапазоні частот 5 Гц - 2 кГц; в діапазоні частот 2 - 400 кГц | 250 нТл 25 нТл |
| Поверхневий електростатичний потенціал не повинен перевищувати | 500 нТл |

Потужність експозиційної дози рентгенівського випромінювання на відстані 0,05 м від екрана та корпусу відеотерміналу при будь-яких положеннях регулювальних пристроїв відповідно до Норм радіаційної безпеки України (НРБУ-97), затверджених постановою державного санітарного лікаря Міністерства охорони здоров'я України від 18.08.97 №58, не повинна перевищувати 7,74-12 А/кГ, що відповідає еквівалентній дозі 0,1 мбер/год (100 мкР/год).

4.2.5. Виробничий шум

У щитовому приміщенні шум непостійний, тому в якості нормованого параметру буде еквівалентний рівень звуку. Головне джерело шуму – це двигуни шнеків, конвеєрів і вентиляторів, а також димососу. Для боротьби із шумом передбачене поліпшення режиму роботи й експлуатації устаткування, центрування й балансування механізмів. Реальний еквівалентний рівень звуку складає 50 дБА. Для зменшення шуму стіни покриті плиткою (керамічною), що поглинає звук.

Здійснюється зниження шуму у виконавчих двигунах (за місцем), виробляються заміна негнучкими, звуковбирними, звукоізолюючими матеріалами.

4.2.6. Виробничі вібрації

Параметри вібрації нормуються:

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 131 |

- Допустимі величина віброшвидкості – V , м/с
- Допустима величина віброприскорення – W м/с²

Параметри вібрацій нормуються в залежності від частоти та напрямку коливань, їх виду, часу дії протягом зміни.

Оператор піддається незначним вібраціям, так як рівень загальної безпеки не перевищує припустимого рівня вібрації на робочому місці, амплітуда вібрацій меншу 0,009 у діапазоні частот від 15Гц до 200Гц. Для виміру вібрацій використовуються вібро-вимірювальний комплект ИШВ-1, що реєструє амплітуди вібрацій від 0,005 мм до 1,5 мм. Заходи, що проводяться з метою не допустити вібрації: контроль за експлуатацією, монтажем, ремонтом обладнання, виконання правил технічної експлуатації машин, впроваджуються нові технології і засоби автоматизації на базі СВТ.

5.3. Пожежна безпека та профілактика

Будівлі і ті їх частини, в яких розташовуються ЕОМ, а саме до них відноситься приміщення операторської, повинні мати не нижче II ступеня вогнестійкості. За вибухопожежною небезпекою приміщення операторської згідно з ДСТУ Б.В.1.1-36:2016 має категорію В, а робочі зони приміщення за пожежонебезпекою відносяться згідно з НПАОП 40.1-1.32-01- до класу П-Па.

Джерелами загоряння в приміщенні можуть бути: коротке замикання, перевантаження електромереж, несправність освітлювальних приладів, діяльність оператора.

Тому для запобігання пожежі прийняті наступні заходи:

- Застосовуються плавкі запобіжники для захисту від короткого замикання і струмів тривалих перевантажень;
- мережа з напругою 220В виконана проводами у захисних трубах, прокладена в підлозі і стінах;
- Освітлювальні установки піддаються постійному нагляду: регулярному очищенню освітлювальних приладів, своєчасній заміні перегорілих ламп, поточному і профілактичному ремонту;

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Лист | № докум. | Підпис | Дата |

- Проводиться контроль за справністю електропроводки;
- Виконавчі пристрої виконані у вибухозахищеному вигляді.’

Технічні рішення системи протипожежного захисту спрямовані на обмеження поширення пожежі, захист людей і матеріальних цінностей.

Вибір пожежних сповіщувачів здійснюється залежно від характерних приміщень, виробництв, технологічних процесів згідно з ДБН В 2.5-56-2014 «Системи протипожежного захисту».

Відстань між пожежними сповіщувачами не повинна перевищувати 9 м, а відстань до стіни повинна становити не більше 4,5 м. При виявленні задимлення повідомлення передається на пульт охорони будівлі, а також вмикається звукове сповіщення.

Приміщення з ЕОМ повинні бути оснащені переносними вуглекислотними вогнегасниками з розрахунку 2 шт. на кожні 20 м² площі приміщення з урахуванням граничнодопустимих концентрацій вогнегасної рідини відповідно до вимог ДСТУ 3675-98 та ISO 3941-2007.

У приміщеннях повинна бути встановлена система об’ємного газового пожежогасіння з децентралізованим зберіганням вогнегасної речовини. Як вогнегасна речовина використовується зріджений двоокис вуглецю. Розпилювачі мають бути розміщені не більше, ніж 4 м один від одного, а оскільки двоокис вуглецю в зрідженому стані має густину більшу, ніж повітря, то відстань від стін має бути не більше 0,5 м.

Оскільки в приміщенні перебувають люди, то система пожежогасіння має бути обладнана пристроєм відключення автоматичного пуску. Резервування балонів з вогнегасною речовиною проводиться із розрахунку по одному балону на кожний відпрацьований. Додатково до цього додається по одному балону на кожні вісім балонів для резервування несправних балонів.

| | | | | | | |
|-----|------|----------|--------|------|-----------------------|-----|
| | | | | | ТО8151.0014.001.АТХ.П | Арк |
| Зм. | Лист | № докум. | Підпис | Дата | | 133 |

ВИСНОВКИ

Результатом виконання кваліфікаційної роботи є розроблене рішення ПоТ з використанням PLCnext. Використання та адаптування бібліотеки каркасу з відкритим кодом PACFramework до PLCnext дало можливість стандартизувати рівень Edge в ПоТ рішеннях та покращити контроль і обслуговування виробничого обладнання та зменшити витрати часу на рутинну роботу персоналу.

Першим етапом було ознайомлення з новим середовищем розробки PLCnext Engineer з її базовими можливостями та написання імітаційного об'єкту рецептурно-змішувального комплексу для приготування продукції з НМІ для наладки установки.

Наступним етапом було ознайомлення з технологією побудови ПЛК - PACFramework. Розробка власного проекту з використанням концепцій каркасу вирішує багато проблем виробництва.

Також було реалізований обмін даними через OPC UA між PLCnext та Node-Red з подальшим зберіганням даних у глобальному контексті. Було інтегровано PLCnext в рішення ПоТ.

Інтеграція PACFramework без засобів автоматизації (PACFramework-tools) є дуже великою і об'ємною роботою, тому в дипломній роботі було реалізовано лише необхідний, або базовий функціонал. Створений проект дає можливість добавляти в проект інші сучасні технології, наприклад доповнену реальність, що планується в майбутньому.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. PLCnext Technology
(<https://www.phoenixcontact.com/ru-pc/otrasli-promyshlennosti/plcnext-technology>)
2. PLCnext Engineer
(<https://www.phoenixcontact.com/ru-pc/produkty/programmnoe-obespechenie-plcnext-engineer-1046008>)
3. PLCnext Engineer Simulation
(<https://www.plcnext-community.net/news/plcnext-engineer-simulation/>)
4. Бібліотека OPC UA
(<https://github.com/pupenasan/NodeREDGuidUKR/tree/master/opcu>)
5. ТЕХНОЛОГІЇ ІНТЕРНТУ РЕЧЕЙ: Б.Ю. Жураковський, І.О. Зенів
(https://ela.kpi.ua/bitstream/123456789/42078/1/Zhurakovskiy_B_Zeniv_Tehnologii_internet_rechey.pdf)
6. Основна сторінка PACFramework
(<https://www.tda.in.ua/pacframework>)
7. PAC Framework V1.02. Функціональний каркас для розробки прикладного програмного забезпечення для промислових контролерів: О.М.Пупена, Р.М.Міркевич, О.М.Клименко, В.В.Полупан, Д.В.Мацебула
(<https://sites.google.com/view/i4uinua/%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%97/pac-framework>)
8. Сторінка Facebook PACFramework
(<https://www.facebook.com/groups/pacframeworkcommunity>)
9. Сторінка GitHub PACFramework
(<https://github.com/pupenasan/PACFramework>)
10. Зеркалов, Д.В. Охорона праці в галузі: Загальні вимоги [Текст] : навч. посібник / Д.В. Зеркалов. – К.: Основа, 2011. – 551 с.
11. Демченко, В.А. Автоматизация и моделирование технологических процессов АЭС и ТЭС. - О.:Астропринт, 2001. - с. 305.

12. ДБН Д.2.4-17-2000. Електромонтажні роботи. — Введений 01.10.2000 — Київ : Мінрегіон України, 2000 — [14 с.]
13. ДБН В.2.5-23:2010. Проектування електрообладнання об'єктів цивільного призначення. — Введений 01.10.2010 — Київ : Мінрегіон України, 2010 — [52 с.]
14. ДСТУ Б А.2.4-3-95 (ГОСТ 21.408-93). Правила виконання робочої документації автоматизації технологічних процесів.
15. ДСТУ Б А.2.4-16:2008. Автоматизація технологічних процесів. Зображення умовних приладів і засобів автоматизації в схемах.

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Лист | № докум. | Підпис | Дата |