

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ЛЮДИНО-МАШИННОГО ІНТЕРФЕЙСУ ОПЕРАТОРА
ХІМІЧНОЇ ПРОМІСЛОВОСТІ
HUMAN-MACHINE INTERFACE DEVELOPMENT
FOR A CHEMICAL OPERATOR

Виконав: студент 4 курсу, групи КНТз-117
Спеціальності 121 Інженерія програмного
забезпечення
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інженерія програмного забезпечення
Червоненко А.О.
(прізвище та ініціали)

Керівник Сердюк С.М.
(прізвище та ініціали)

Рецензент Неласа Г.В.
(прізвище та ініціали)

2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)

Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2021 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

Червоненко Анатолія Олександровича
 (прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка людино-машинного інтерфейсу оператора хімічної промисловості

Development of a Human-Machine Interface for the Operator of the Chemical Industry

керівник проєкту (роботи) Сердюк Сергій Микитович, к.т.н., доцент
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом закладу вищої освіти від “ 30 ” березня 2021 року № 103

2. Строк подання студентом проєкту (роботи) 26.05.2021

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Розробка програми. 4. Керівництво оператора.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Сердюк С.М., доцент		
Нормоконтроль	Дейнега Л.Ю., ст. викладач		

7. Дата видачі завдання “28” лютого 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2 тиждень	Розділ 1
3	Розробка та удосконалення методів, моделей й алгоритмів вирішення задачі.	3 тиждень	Розділ 2
4	Розробка архітектури програми.	4 тиждень	Розділ 3
5	Розробка програми.	5 тиждень	Розділ 3
6	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
7	Нормоконтроль та рецензування.	7 тиждень	
8	Захист роботи.	8 тиждень	

Студент

(підпис)

Червоненко А.О.

(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис)

Сердюк С.М.

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
109 с., 10 табл., 51 рис., 3 дод., 24 джерел.

ІНТЕРФЕЙС, СИСТЕМА УПРАВЛІННЯ, ПРОМИСЛОВІСТЬ,
ПРОГРАМОВАНИЙ ЛОГІЧНИЙ КОНТРОЛЕР, SCADA-СИСТЕМА.

Об'єкт дослідження – процес автоматизації діяльності оператора АСУТП хімічної промисловості.

Предмет дослідження – методи розробки людино-машинного інтерфейсу оператора АСУТП хімічної промисловості.

Мета роботи – розробка людино-машинного інтерфейсу оператора АСУТП хімічної промисловості на основі методів, що підвищують ситуаційну обізнаність оператора в промисловому процесі.

Матеріали, методи та технічні засоби: стандарти ергономіки та людино-орієнтовного проектування, метод високопродуктивної людино-машинної взаємодії, метод аналізу і управління ситуаційною обізнаністю, середовища розробки CoDeSys та Galileo, персональний IBM-сумісний комп'ютер з процесором Intel Celeron N2830, оперативною пам'яттю 8 Гб, 19'' монітором, клавіатурою та маніпулятором типу «миша», що функціонує під управлінням операційної системи Microsoft Windows 10.

Результати. Спроектовано та розроблено АСУТП лінії виробництва хімічних розчинів з людино-машинним інтерфейсом оператора.

Галузь використання – хімічна промисловість.

ABSTRACT

Explanatory note to the bachelor's thesis: 109 pages,
10 tables, 51 figures, 3 appendices, 24 sources.

INTERFACE, CONTROL SYSTEM, INDUSTRY, PROGRAMMED
LOGICAL CONTROLLER, SCADA SYSTEM.

The object of research is the process of automation of the activity of the process control system operator of the chemical industry.

The subject of research - methods of development of the human-machine interface of the operator of ASUTP of the chemical industry.

The purpose of the work is to develop a human-machine interface of the operator of the chemical industry process control system on the basis of methods that increase the situational awareness of the operator in the industrial process.

Materials, methods and technical means: standards of ergonomics and human-oriented design, method of high-performance human-machine interaction, method of analysis and management of situational awareness, development environments CoDeSys and Galileo, personal IBM-compatible computer with Intel Celeron N2830 processor, RAM 8 GB, 19 " monitor, keyboard and manipulator type "mouse", which runs on the operating system Microsoft Windows 10.

Results. The process control system for the production of chemical solutions with a human-machine operator interface has been designed and developed.

Field of use - chemical industry.

ЗМІСТ

Вступ.....	8
1 Аналіз предметної області.....	10
1.1 Автоматизація систем управління та способи її побудови.....	10
1.2 Огляд програмованого логічного контролера та мов програмування.....	15
1.2.1 Мова сходових діаграм LD.....	20
1.2.2 Мова функціональних блокових діаграм FBD.....	21
1.2.3 Мова структурованого тексту ST.....	24
1.2.4 Мова програмування IL.....	25
1.3 Огляд існуючих SCADA – систем.....	29
1.3.1 CitectSCADA.....	35
1.3.2 Master SCADA.....	36
1.3.3 TRACE MODE.....	36
1.3.4 Galileo.....	37
2 Матеріали і методи.....	40
2.1 Огляд існуючих людино-машинних інтерфейсів.....	40
2.2 Вимоги до розробки людино-машинних інтерфейсів.....	45
2.2.1 Послідовне використання кольору.....	45
2.2.2 Представлення даних.....	48
2.2.3 Ієрархія екранів людино-машинного інтерфейсу.....	51
3 Розробка програми.....	53
3.1 Розробка структури автоматизованої системи управління технологічним процесом.....	53
3.2 Розробка функціональної схеми автоматизованої системи управління технологічним процесом.....	57
3.2.1 Обробка аналогових сигналів.....	58
3.2.2 Керування пневматичними клапанами.....	61
3.2.3 Керування насосами.....	64

3.3 Розробка людино-машинного інтерфейсу.....	67
4 Керівництво оператора.....	80
4.1 Призначення програми.....	80
4.2 Умови виконання програми.....	80
4.3 Виконання програми.....	80
Висновки.....	85
Перелік джерел посилання.....	87
Додаток А Зображення людино-машинного інтерфейсу.....	90
Додаток Б Текст програми.....	96
Додаток В Слайди презентації.....	99

ВСТУП

В сучасному світі взаємодія з електронними пристроями є невід’ємною частиною життя кожної людини. Неможливо уявити функціонування індустріальної промисловості, шкіл і університетів, системи охорони здоров’я, фінансових і правничих установ без інформаційних технологій. В даному випадку, взаємодія – це обмін діями і реакціями на ці дії між електронним пристроєм і людиною, що дозволяє досягати відповідних цілей. Такий вид взаємодії називається людино-машинною (англ. Human-machine interface, HMI). До засобів взаємодії можна віднести миш, клавіатуру, дисплей, а у випадку промисловості до цього списку додаються кнопки та світлова індикація.

Пандемія коронавірусу зумовила підвищений попит на онлайн сервіси, такі як інтернет банкінг, доставка на дім, засоби відеозв’язку, тощо. Переважна більшість невідготтовлених людей використовує всі доступні функції цих сервісів, за рахунок досвіду розробників людино-машинних інтерфейсів. Зі збільшенням можливостей комп’ютера, апаратного та програмного забезпечення, дизайнери інтерфейсів почали звертати увагу на те, як люди взаємодіють з комп’ютерами і як повинні бути розроблені інтерфейси для полегшення цієї взаємодії, тому змогли відповісти на попит, інтуїтивно зрозумілим інтерфейсом.

В той же час у світі автоматизованої промисловості, відбулося достатньо надзвичайних подій з вини людини, однією з таких подій є вибух на Техаському нафтопереробному заводі [1]. Вибух відбувся 23 березня 2005 року, коли дистиляційна колона для ізомеризації вуглеводнів переповнилася і послала гейзер рідини в повітря, вироблена хмара легкозаймистих парів була швидко запалена від стоячого на холостому ході автомобіля. Загинули п’ятнадцять робочих і 180 були поранені, також були вибиті вікна в будинках в радіусі 1,2 км. Дослідницька рада з хімічної безпеки та ризиків США (U.S. Chemical Safety and Hazard Investigation Board - CSB) витратила майже два роки на розслідування нещасного випадку. В їх заключну доповідь увійшов ряд причин, включаючи недостатню підготовку, недотримання процедур безпечного запуску, неточний інструментарій, погане

обслуговування, і «погано розроблена комп'ютеризована система управління, що перешкоджає здатності операційного персоналу визначити ситуацію переповнення колони». Екран системи комп'ютеризованого управління, надавав дані про те, скільки рідини входить в блок, а на іншому екрані було показано, скільки продукту залишає блок. Наявність двох різних місць для зчитування, зменшила видимість і важливість моніторингу потоків рідини в порівнянні один з одним і в результаті не вдалося зробити очевидний висновок про дисбаланс між двома потоками.

З моменту аварії пройшло більше шістнадцяти років, за цей час було проведено багато досліджень людських чинників і ергономіки, в результаті яких було створені нові стандарти ергономіки [2], людино-орієнтовного проєктування [3], але не ефективні людино-машинні інтерфейси залишаються звичайним явищем у всій галузі.

Актуальність дипломної роботи полягає саме у запобіганні виникненню надзвичайних подій, за рахунок підвищення ситуаційної обізнаності людини в промисловому процесі.

Об'єктом дослідження даної роботи є автоматизація промисловості.

Предметом дослідження виступають людино-машинні інтерфейси в промисловості.

Метою роботи є розробка людино-машинного інтерфейсу на основі методів, що підвищують ситуаційну обізнаність оператора в промисловому процесі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Автоматизована система управління технологічним процесом та способи її побудови

Хімічна галузь промисловості характеризується високими вимогами до контролю поточних значень технологічних параметрів. Це пояснюється тим, що протікання хімічних процесів повинно відбуватися в строго заданих умовах. Недотримання цих умов може спричинити брак вихідного продукту або порушення норм техніки безпеки. Для виконання таких задач застосовуються автоматизовані системи управління.

Автоматизована система управління (АСУ) – комплекс апаратних і програмних засобів для управління технологічним процесом. АСУ застосовуються в різноманітних галузях промисловості, енергетики, транспорті тощо. Термін «автоматизована» на противагу терміну «автоматична», зберігає роль управління процесом за людиною. Автоматизовані системи розрізняють за об'єктами управління: технологічними процесами – АСУ ТП; підприємством – АСУП [4].

Функціонування сучасної автоматизованої системи управління технологічним процесом залежить від злагодженої роботи її частин. Високу якість взаємодії всередині програмно-апаратного комплексу забезпечують високопродуктивні обчислювальні засоби, що підтримують відкриті протоколи обміну даними, і уніфіковані сигнали контролю і управління. А належний рівень зв'язку між таким комплексом і оперативним персоналом досягається за допомогою засобів людино-машинного інтерфейсу (НМІ). Ступінь важливості НМІ переоцінити складно: в АСУ, незалежно від ступеня автоматизації, саме оператори грають ключову роль. Від них залежить штатне функціонування всього автоматизованого технологічного комплексу. Основні цілі автоматизації управління :

- надання особі, що приймає рішення (ОПР), релевантних даних для прийняття цих рішень;
- прискорення виконання окремих операцій з збору й обробки даних;

- зниження кількості рішень, які повинна приймати ОПР;
- підвищення рівня контролю і виконавчої дисципліни;
- підвищення оперативності управління;
- зниження затрат ОПР на виконання допоміжних процесів;
- підвищення ступеня обґрунтованості прийнятих рішень.

У склад типових функцій АСУ ТП входять :

- збір інформації про технологічні параметри і стан обладнання;
- перерахунок сигналів у фізичні величини;
- контроль технологічних параметрів на фізичну достовірність, на відповідність технологічному регламенту, на досягнення аварійних меж;
- оцінка стану обладнання;
- ручне введення інформації в систему з використанням пульта оператора або клавіатури;
- обмін інформацією між обчислювальними засобами АСУ ТП (контролер, робочі і інженерні станції, сервери);
- формування і видача сигналів світлової і звукової сигналізації;
- візуалізація інформації в зручному вигляді для оперативного персоналу;
- ведення бази даних реального часу;
- підрахунок техніко-економічних показників виробництва;
- прогнозування аварійних ситуацій;
- аварійне відключення (приклад, захист двигуна від холостого ходу);
- логічне керування обладнанням;
- обмін даними з суміжними системами управління;
- формування звітів.

Найважливіша мета, яку переслідують розробники АСУ – підвищення продуктивності об'єкту за рахунок вдосконалення методів управління технологічним процесом. На сьогоднішній день існує три основних способи створення НМІ. Перший і традиційний спосіб – це застосування світлосигнальної арматури (рис.1.1) у вигляді перемикачів, кнопок, сигнальних ламп, колон тощо.

Цей спосіб підходить для окремих агрегатів і установок (електродвигуни, насосні агрегати, вентилятори і т.п.), на яких реалізовані прості технологічні процеси і використовуються системи управління на базі релейно-контактних схем.



Рисунок 1.1 – Світлосигнальна арматура [5]

Другий спосіб являється розвитком першого. Справа в тому, що для керування складними технологічними об'єктами з великою кількістю сигналів контролю і управління, застосування НМІ реалізованих тільки на базі світлосигнальної апаратури, буде не ефективним. Громіздкі пульти управління з великою кількістю сигнальних ламп та перемикачів, не сприяють підвищенню якості взаємодії з оперативним персоналом. Тому другий метод заснований на застосуванні таких технічних рішень, як панельні комп'ютери і панелі оператора (рис.1.2). Операторська панель, що розміщується в лівому верхньому куті пульту управління, як видно з рисунки нижче, зазвичай являє собою дисплей з сенсором або допоміжними кнопками, що використовується для моніторингу і управління технологічним процесом. В деяких випадках комплектація панелей передбачає функцію програмованого логічного контролера (ПЛК). Панелі оператора використовуються в різних сферах людської діяльності. Їх активно застосовують для автоматизації окремих агрегатів і установок або цілого технологічного процесу у структурі АСУ ТП.



Рисунок 1.2 – Пульт управління на базі панелі оператора [5]

Третій спосіб – це реалізація людино-машинної взаємодії на базі автоматизованих робочих місць (АРМ), що представляють собою промисловий персональний комп'ютер з встановленою SCADA – системою (рис.1.3). SCADA (аббр. з англ. Supervisory Control And Data Acquisition) – програмне забезпечення для відображення інформації, управління і збору даних про об'єкт.



Рисунок 1.3 – АРМ на базі промислового ПК [6]

Вибір того чи іншого способу організації людино-машинної взаємодії залежить від ряду факторів: складності і архітектури автоматизованої системи, доцільності застосування тих чи інших технічних рішень тощо. Трирівнева архітектура сучасних АСУ ТП (рис.1.4) визначила комбінований підхід до реалізації НМІ: на верхньому рівні розгорнута SCADA – система; на середньому рівні розміщують ПЛК, панелі оператора, або панельні комп'ютери; на нижньому рівні знаходяться датчики та виконавчі механізми.

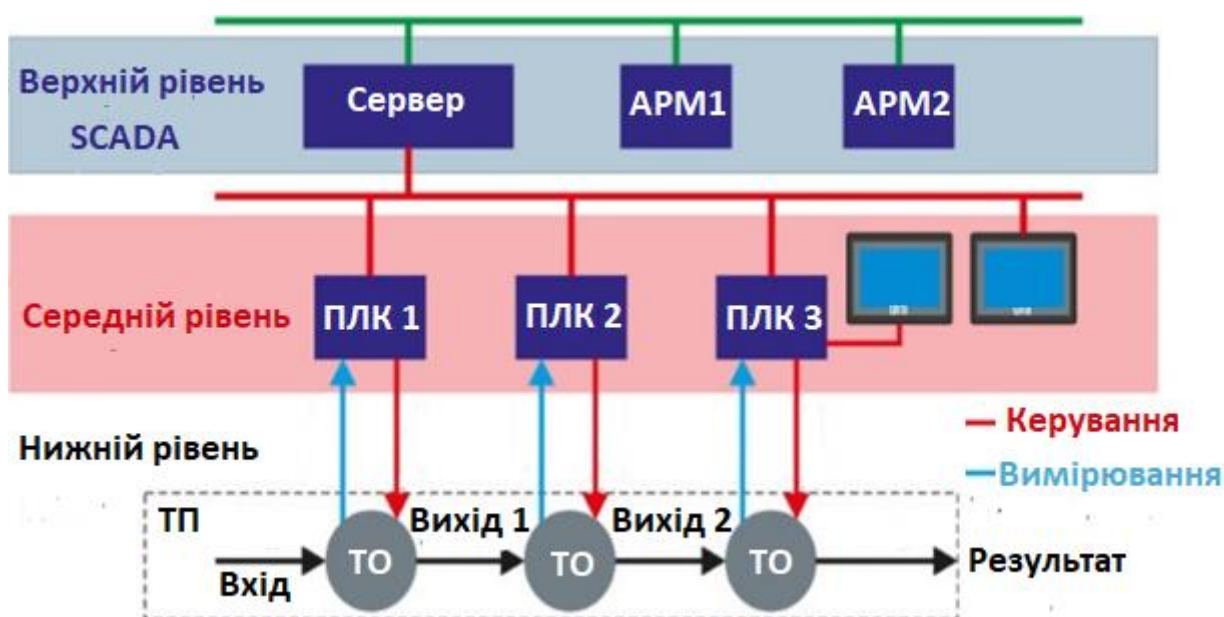


Рисунок 1.4 – Трирівнева структура АСУ ТП

Застосування панелей оператора на середньому рівні дозволяє передусім підвищити надійність роботи автоматизованої системи. Наприклад, у разі виходу з ладу центрального АРМ на базі SCADA – системи, оперативний персонал може локально робити налаштування і контроль параметрів технологічного процесу. Використання операторських панелей також дозволяє підвищити швидкість і ефективність пусконаладжувальних робіт. Більш того при автоматизації невеликих об'єктів НМІ-панель може стати хорошою альтернативою повноцінній SCADA – системі і промисловим панельним комп'ютерам.

1.2 Огляд програмованого логічного контролера та мов програмування

Програмовані логічні контролери широко використовуються в галузі промислової автоматизації різноманітних технологічних процесів на великих та малих підприємствах. Їх використання значно спрощує створення і експлуатацію, як складних автоматизованих систем так і окремих агрегатів, в тому числі – побутового призначення. ПЛК дозволяє скоротити етап розробки, спрощує процес монтування і налагоджування за рахунок стандартизації окремих апаратних і програмних компонентів, а також забезпечує підвищену надійність в агресивному середовищі, зручний ремонт і модернізацію за необхідністю.

Прийнято вважати, що задача створення прообразу сучасного ПЛК виникла в кінці 60-х років минулого століття. Зокрема, в 1968 році вона була сформульована провідними фахівцями General Motors. Тоді ця компанія намагалася знайти заміну для складної релейної системи управління. Відповідно до отриманої задачі на проектування, нова система управління мала відповідати таким критеріям як:

- просте і зручне створення технологічних програм;
- можливість зміни робочої керуючої програми без втручання в саму систему;
- просте і недороге обслуговування;
- підвищена надійність при зниженій вартості, в порівнянні з подібними релейними системами.

Наступні розробки в General Motors, Allen-Bradley і інших компаніях привели до створення системи управління на базі мікроконтролерів, яка аналізувала вхідні сигнали від технологічних датчиків і керувала електроприводами виконавчих пристроїв [7].

Термін ПЛК (з англ. Programmable Logic Controller, PLC) згодом був визначений в стандартах EN 61131 [8]. ПЛК - це уніфікована цифрова керуюча електронна система, спеціально розроблена для використання у виробничих умовах. ПЛК постійно контролює стан пристроїв введення та приймає рішення на основі користувальницької програми для управління станом вихідних пристроїв.

Спрощене уявлення складу і принципу дії ПЛК добре продемонстровано на рис.1.5. З нього видно, що ПЛК має три основні секції:

- вхідні;
- вихідну;
- центральну.



Рисунок 1.5 – Структура і принцип дії ПЛК

Центральна секція містить центральний процесор (ЦП), пам'ять і систему комунікацій. Вхідна секція ПЛК забезпечує введення в центральну секцію стану перемикачів, датчиків і смарт-пристроїв. Через вихідну секцію ЦП керує зовнішніми виконавчими пристроями, серед яких можуть бути електромагнітні пускачі моторів, джерела світла, клапани та смарт-пристрої.

Слід відразу зазначити, що в великих ПЛК, крім ЦП, що діє в режимі «ведучий», можуть бути додаткові «ведені» ПЛК зі своїми ЦП. В якості ЦП невеликого ПЛК використовуються стандартні мікропроцесори (МП). Зазвичай 8- і 16-розрядні МП цілком справляються з усіма стандартними завданнями. Але, як

зазначено в МЕК 61131, вибір конкретного МП все ж залежить від завдань, покладених на даний тип ПЛК.

Сучасні ПЛК (рис.1.6), що використовують інноваційні технології, далеко пішли від перших спрощених реалізацій промислового контролера, але закладені в систему управління універсальні принципи були стандартизовані і успішно розвиваються вже на базі новітніх технологій. Найбільшими світовими виробниками ПЛК сьогодні є компанії Siemens AG, Allen-Bradley, Rockwell Automation, Schneider Electric, Eaton.



Рисунок 1.6 – Сучасні програмовані логічні контролери [9]

Один сучасний ПЛК здатний замінити десятки регуляторів, сотні таймерів і тисячі реле. Використовуючи РС запрограмувати таку систему дуже легко. Застосування РС в якості програмуючої станції ПЛК є сьогодні домінуючим рішенням. Це не тільки спрощує програмування, але і вирішує завдання

архівування проєктів, підготовки документації, візуалізації і моделювання. Комп'ютер дає зручний універсальний інструмент як для програмування найпростіших локальних задач на ПЛК, так і для АСУ ТП.

Під час створення програмованого логічного контролера виникало питання, яким чином він буде програмуватися. Алгоритмічні мови програмування комп'ютерів того часу були орієнтовані на рішення обчислювальних задач. Професія програміста вважалася винятково рідкісною і важкою, таких фахівців не було ні в одному підприємстві. Ідеальним варіантом могла б стати автоматична трансляція принципових схем релейних автоматів в програми для ПЛК. Згідно з цим, в США у період релейної автоматизації, була винайдена мова релейно-контактних схем або LD (аббр. з англ. Ladder Diagram). Спеціаліст-технолог міг "перемалювати" схему управління на дисплеї програмуючої станції ПЛК. Природньо, що схема зображувалася не графічно, а за допомогою умовних символів. До Європи мода на ПЛК прийшла трохи пізніше, коли релейні шафи були вже успішно замінені на шафи з логічними мікросхемами. Тому виникла необхідність винаходу інших мов програмування зрозумілих новому поколінню інженерів. Так в Німеччині з'явилися мови простих текстових інструкцій нагадуючих асемблер (IL, Instruction List). У Франції виникла графічна мова функціональних блокових діаграм (FBD, Functional Block Diagram) і високорівневі діаграми опису етапів і умов переходів (SFC, Sequential Function Chart). Застосовувалися також мови, які використовуються для програмування комп'ютерів (Pascal, Basic) [10].

В кінці сімдесятих років склалася вкрай складна ситуація. Кожен виробник ПЛК розробляв власну мову програмування, тому ПЛК різних виробників були програмно несумісні, крім того існувала проблема апаратної несумісності. Заміна ПЛК на продукт іншого виробника перетворилася на величезну проблему. Покупець ПЛК був змушений використовувати вироби тільки однієї фірми або витратити сили на вивчення різних мов і кошти на придбання відповідних інструментів.

У підсумку в 1979 році в рамках Міжнародної Електротехнічної Комісії (МЕК) була створена спеціальна група технічних експертів з проблем ПЛК. Їй було поставлено завдання виробити стандартні вимоги до апаратних засобів, програмного забезпечення, правил монтажу, тестування, документування та засобів зв'язку ПЛК.

У 1982 році був опублікований перший чорновий варіант стандарту, який отримав найменування МЕК 61131-3 [8]. Зважаючи на складність отриманого документа, було вирішено розбити його на кілька частин, питанню програмування присвячена третя частина стандарту "Мови програмування ПЛК". Робочою групою МЕК було прийнято досить оригінальне рішення. З усього різноманіття, існуючих на момент розробки стандарту, мов програмування ПЛК були виділені 5 мов, які отримали найбільше поширення.

Специфікації мов були доопрацьовані, так що стало можливим використовувати в програмах написаних на будь-якій з цих мов стандартизований набір елементів і типів даних. Такий підхід МЕК не раз піддавався критиці, але час довів правильність цього рішення.

Реалізація подібного підходу дозволила залучити до програмування одного і того ж ПЛК фахівців різних галузей знань (і що особливо важливо - різної кваліфікації): фахівців з релейного автоматизму, що програмують в LD, фахівців в області напівпровідникової схемотехніки і автоматичного регулювання для яких звична мова FBD, програмістів, що мають досвід написання програм для комп'ютерів на мові асемблера (йому відповідає мова IL для ПЛК), на мовах високого рівня (мова ST).

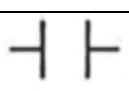
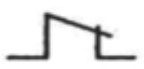
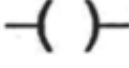
1.2.1 Мова сходових діаграм (LD)

Мова LD - це графічна мова, що застосовується для опису логічних виразів різного рівня складності, графічного представлення булевих рівнянь. Вона містить контакти (вхідні аргументи) і котушки (вихідні змінні). Елементи організовуються в мережу релейно-контактних схем [10].

Кожному контакту ставиться у відповідність логічна змінна, яка визначає його стан. Її ім'я ставиться над контактом і служить його назвою. Якщо контакт замкнутий, то змінна має значення true, якщо розімкнутий - false. Послідовне з'єднання контактів або ланцюгів відповідає логічній операції - І / AND, паралельне - АБО / OR. Нормально замкнутий (інверсний) контакт рівнозначний логічній операції НЕ.

Релейна схема являє собою 2 вертикальні шини живлення, між якими розташовані горизонтальні ланцюги з контактів і котушок реле (рис.1.7). Графічні символи мови LD відповідають елементам електричних ланцюгів і мають ті ж назви і позначення (табл.1.1).

Таблиця 1.1 – Позначення графічних символів мови LD

LD	Умовно-графічне позначення	Позначення
		Нормально розімкнутий контакт
		Нормально замкнутий контакт
		Котушка реле

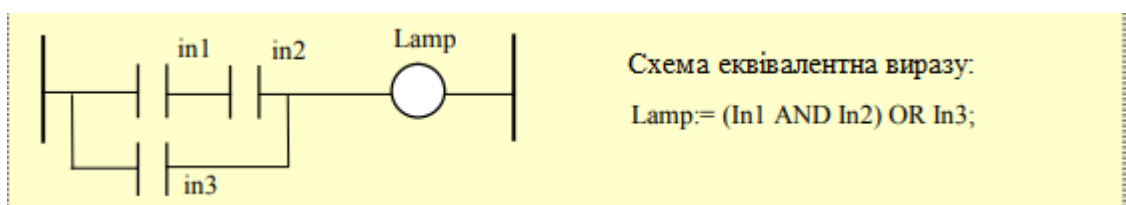


Рисунок 1.7 – Приклад схеми на мові LD [11]

LD-програма виконується послідовно зліва направо і зверху вниз. У кожному робочому циклі одноразово виконуються всі ланцюги, що входять в мережу. Будь-яка змінна в рамках одного ланцюга завжди має одне і те ж значення. Якщо навіть реле в ланцюзі змінить змінну, то нове значення надійде на контакти тільки в наступному циклі. Ланцюги, розташовані нижче, отримають нове значення змінної відразу, а розташовані вище - тільки в наступному циклі. Строгий порядок виконання ланцюгів дуже важливий. Завдяки жорсткому порядку виконання LD-програми зберігають стійкість при наявності зворотних зв'язків.

1.2.2 Мова функціональних блокових діаграм FBD

FBD - графічна мова, що застосовується для побудови комплексних процедур та складаються з різних функціональних бібліотечних блоків - арифметичних, тригонометричних, регуляторів, мультиплексорів тощо [10]. Найбільш підходить для управління безперервними процесами і регулювання. При цьому здійснюється:

- уявлення функцій за допомогою блоків, пов'язаних між собою;
- з'єднання між виходами функціональних блоків в явному вигляді можуть бути відсутні;
- вихід блоку може з'єднуватися зі входами одного або декількох блоків.

Об'єктами мови FBD являються елементарні функції та елементарні функціональні блоки (ФБ). Вони знаходяться в бібліотеці функціональних блоків, логіка їх роботи написана на мові С і не може бути змінена в редакторі FBD, змінювати можна тільки їх параметри. Також об'єктами мови являються функції, ФБ користувача і змінні, які конструюються користувачем з елементів мови FBD. Змінні бувають входні, вихідні і входні/вихідні. На рис.1.8 показані: входна змінна – «in_var», вихідна змінна – «out_var», і входна/вихідна змінна «in_out_var».

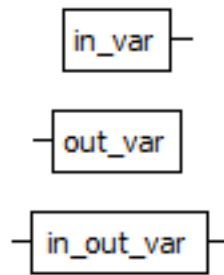


Рисунок 1.8 – Зображення змінних в мові FBD

Графічне зображення функції наведено на рис.1.9. Зліва розташовуються входи (IN1,IN2), з права виходи(OUT).

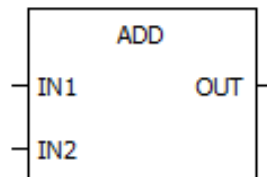


Рисунок 1.9 – Зображення функції в мові FBD

Відповідно, змінні з'єднуються з вхідними та вихідними параметрами функцій і функціональних блоків. Вхідні змінні можуть бути пов'язані лише з вхідними параметрами функції чи функціонального блоку, вихідні змінні - тільки з вихідними параметрами функції чи функціонального блоку, вхідні / вихідні змінні - як входами, так і з виходами функції чи функціонального блоку.

Якщо функціональний блок викликається з параметрами EN/ENO і при цьому значення EN дорівнює нулю, то алгоритми, які визначаються в функціональному блоці, не виконуватимуться. У цьому випадку значення ENO автоматично встановлюється рівним 0. Якщо ж значення EN дорівнює 1, то алгоритми, які визначаються функціональним блоком, будуть виконані. Після виконання цих алгоритмів без помилок, значення ENO автоматично встановлюється рівним 1. Якщо ж виникає помилка під час виконання цих алгоритмів, то значення ENO буде встановлено рівним 0. Поведінка

функціонального блоку однакова, як у разі виклику функціонального блоку з EN = 1, так і при виклику без параметрів EN / ENO.

На рис.1.10 приведена FBD діаграма, що складається з наступних функціональних блоків: SR0, AND, TP0.

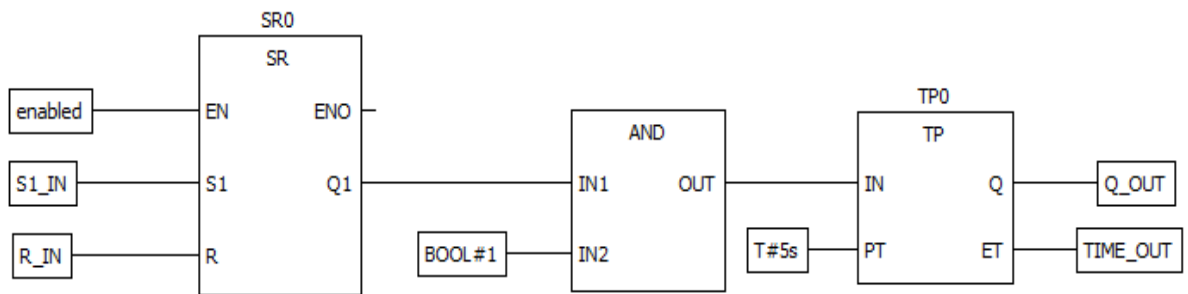


Рисунок 1.10 – Приклад FBD діаграми

Функціональний блок SR0 є SR-тригером. У нього є входи S1, R і вихід Q1, а також додатковий вхід EN і вихід ENO, що дозволяють вмикати і вимикати виконання SR0. Вихід Q1 з'єднаний з входом IN1 блоку AND, що представляє собою «Логічне І». Вхід IN2 типу BOOL з'єднаний з літералом «BOOL # 1», який завжди позитивний. Вихід OUT блоку AND з'єднаний з входом IN функціонального блоку TP0, що представляє собою повторювач імпульсів. Вхід PT типу TIME, з'єднаний з літералом «T # 5s», який задає значення 5 секунд.

Якщо після запуску виконання даного функціонального блоку «enabled» дорівнює True і змінна «S1_IN» теж True, функціональний блок SR0 починає виконуватися. На виході OUT функціонального блоку AND буде значення True як тільки Q1 у SR0 буде дорівнювати True. Отже, як тільки OUT стає True вхід IN функціонального блоку TP0 приймає теж True і починається відлік таймера. До поки таймер не досягне значення PT вихід Q функціонального блоку TP0 буде дорівнювати True. При досягненні таймером ET значення PT, тобто через 5 секунд вихід Q становиться False. Як тільки вхід IN функціонального блоку TP0 дорівнюватиме значенню FALSE, лічильник скидається .

FBD дозволяє використовувати потужні алгоритми простим викликом функцій і функціональних блоків. Чудово підходить для невеликих додатків і зручна для реалізації складних речей подібно пропорційно-інтегрально-диференціальному (ПІД) регулятору, масивам тощо. Дана мова запозичує символіку булевої алгебри і, так як булеві символи мають входи і виходи, які можуть бути з'єднані між собою, FBD є більш ефективною для подання структурної інформації, ніж мова релейно - контактних схем.

1.2.3. Мова структурованого тексту ST

ST (Structured Text) - це текстова мова високого рівня загального призначення, по синтаксису схожа з мовою Pascal. Згідно IEC 61131-3 ключові слова повинні бути введені в символах верхнього регістру. Прогалини і мітки табуляції не впливають на синтаксис, вони можуть використовуватися всюди [10].

Основою ST – програми служать вирази. Вирази складаються з операндів (констант і змінних) і операторів. Оператори - є "командами" мови програмування ST. Вони повинні закінчуватися крапкою з комою. Один рядок може містити кілька операторів (відокремлюваних крапкою з комою). Результат обчислення виразу присвоюється змінній за допомогою оператора присвоювання ": =". Кожен вираз обов'язково закінчується крапкою з комою ";". Вираз складається з змінних, констант і функцій, розділених операторами, порядок їх виконання – зліва на право, наприклад: `VAR1 := 1+VAR2 / ABS(VAR2) ; .`

Відповідно до стандарту IEC 61131-3, мова ST підтримує весь необхідний набір типів даних , аналогічний класичним мовам програмування:

- цілочисельні типи: SINT (char), USINT (unsigned char), INT (short int), UINT (unsigned int), DINT (long), UDINT (unsigned long), LINT (64 біт ціле), ULINT (64 біт ціле без знака);
- дійсні типи: REAL (float), LREAL (double);
- спеціальні типи BYTE, WORD, DWORD, LWORD представляють собою бітові рядки довжиною 8, 16, 32 і 64 біт відповідно.

До конструкцій мови ST відносяться:

- арифметичні операції;
- логічні (побітові) операції;
- операції порівняння;
- операція присвоєння;
- конструкція IF - ELSEIF - ELSE;
- цикл FOR;
- цикл WHILE;
- цикл REPEAT UNTIL;
- конструкція CASE.

До логічних(побітових) операцій відносяться:

- «OR» - Логічне (побітове) додавання;
- «AND» - Логічне (побітове) множення;
- «XOR» - Логічне (побітове) «виключаюче АБО»;
- «NOT» - Логічне (побітове) заперечення.

Мова ST зручна для програм, що включають числовий аналіз або складні алгоритми. Може використовуватися в програмах, в тілі функції або функціонального блоку, а також для опису дії і переходу всередині елементів SFC.

1.2.4 Мова програмування IL

IL - це текстова мова низького рівня, класу асемблера, застосовується для програмування булевих функцій і ефективних, оптимізованих процедур. Це одноадресна мова, тому в більшості її команд використовується невказаний явно поточний результат. У мікропроцесорах для цих цілей використовується внутрішній регістр загального користування, званий акумулятором (АК), тому далі для простоти будемо використовувати це позначення [10].

Програма на мові ІЛ являється списком інструкцій / команд. Кожна команда повинна починатися з нового рядка і містити оператор, який закінчується необов'язковим модифікатором і, якщо необхідно, операндом.

Оператор - це символ арифметичних або логічних операцій, або виклик функціонального блоку (ФБ). Він вказує на операцію, яка повинна бути виконана над вмістом АК і другим операндом, явно заданим в цьому операторі. АК містить поточний результат, що представляє собою неявно заданий операнд. В табл.1.2 приведений перелік операторів мови ІЛ.

Команда - це один оператор плюс один або кілька операндів, розділених комами. Перед командою може стояти позначка (необов'язково), після якої ставиться символ ":".

Таблиця 1.2 – Оператори мови ІЛ [10]

Оператор	Опис
1	2
LD	Завантажити значення операнду в акумулятор
LDN	Завантажити зворотне значення операнду в акумулятор
ST	Присвоїти значення акумулятора операнду
STN	Присвоїти зворотне значення акумулятора операнду
S	Якщо значення акумулятора TRUE, встановити логічний операнд
R	Якщо значення акумулятора FALSE, скинути логічний операнд
AND	Порозрядне І акумулятора і операнду
ANDN	Порозрядне І акумулятора і зворотного операнду
OR	Порозрядне АБО акумулятора і операнду
ORN	Порозрядне АБО акумулятора та зворотного операнду
XOR	Порозрядне розділову АБО акумулятора і операнду
XORN	Порозрядне розділову АБО акумулятора та зворотного операнду

Продовження таблиці 1.2

1	2
NOT	Порозрядна інверсія акумулятора
ADD	Додавання акумулятора і операнду, результат записується в акумулятор
SUB	Віднімання операнду з акумулятора, результат записується в акумулятор
MUL	Множення акумулятора на операнд, результат записується в акумулятор
DIV	Розподіл акумулятора на операнд, результат записується в акумулятор
GT	Значення акумулятора порівнюється зі значенням операнду (greater than). Значення (TRUE або FALSE) записується в акумулятор
GE	Значення акумулятора порівнюється зі значенням операнду (greater than or equal). Значення (TRUE або FALSE) записується в акумулятор
EQ	Значення акумулятора порівнюється зі значенням операнду (equal). Значення (TRUE або FALSE) записується в акумулятор
NE	Значення акумулятора порівнюється зі значенням операнду (not equal). Значення (TRUE або FALSE) записується в акумулятор
LE	Значення акумулятора порівнюється зі значенням операнду (less than or equal to). Значення (TRUE або FALSE) записується в акумулятор
LT	Значення акумулятора порівнюється зі значенням операнду (less than). Значення (TRUE або FALSE) записується в акумулятор
JMP	Перехід до мітки
JMPC	Перехід до мітки за умови, що значення акумулятора TRUE
JMPCN	Перехід до мітки за умови, що значення акумулятора FALSE
CAL	Виклик програмного або функціонального блоку
CALC	Виклик програмного або функціонального блоку за умови, що значення акумулятора TRUE
CALCN	Виклик програмного або функціонального блоку за умови, що значення акумулятора FALSE

Продовження таблиці 1.2

1	2
RET	Вихід з ФБ і повернення в зухвалу програму
RETC	Вихід з ФБ і повернення в зухвалу програму за умови, що значення акумулятора TRUE
RETCN	Вихід з ФБ і повернення в зухвалу програму за умови, що значення акумулятора FALSE

ФБ запускаються за допомогою оператора CAL або за допомогою використання входів ФБ. Програма починається з команди LD – завантаження операнду в акумулятор. Основними операторами для передачі даних є команди LD і ST: LD (LoaD) здійснює завантаження в АК операнду, а ST (STore) здійснює збереження вмісту АК в операнді. Розглянемо приклад використання операторів LD і ST:

LD 10

ADD 25

ST B

В АК завантажуються константа 10, потім додається число 25 (оператор ADD), а результат присвоюється змінній В. Тепер значення змінної В і вміст АК дорівнює 35. Будь-яка наступна команда буде працювати з вмістом АК, рівним 35 (якщо це не команда LD) [14].

Однією з ключових переваг ІЛ є її простота і можливість домогтися оптимізованого коду для реалізації критичних секторів програм. В той же час особливості ІЛ роблять її незручною для опису складних алгоритмів з великою кількістю розгалужень.

1.3 Огляд існуючих SCADA - систем

На сьогоднішній день існує досить багатий вибір різних SCADA – систем, але незважаючи на таке розмаїття на ринку, більшість з них має приблизно однаковий набір функціональних можливостей, які в свою чергу можна поділити на групи:

- налаштування SCADA на конкретну задачу (тобто розробка програмної частини системи автоматизації);
- диспетчерське управління;
- автоматичне керування;
- зберігання історії процесів;
- виконання функцій безпеки;
- виконання загальносистемних функцій.

Однією з основних функцій SCADA є розробка людино-машинного інтерфейсу, тобто SCADA одночасно є і ЧМІ, і інструментом для його створення. Швидкість розробки істотно впливає на рентабельність фірми, яка виконує роботу по впровадженню системи автоматизації, тому швидкість розробки є основним показником якості SCADA з точки зору системного інтегратора. У процес розробки входять наступні операції:

- створення графічного інтерфейсу (мнемосхем, графіків, таблиць, спливаючих вікон, елементів для введення команд оператора і т. д.);
- програмування і налагодження алгоритмів роботи системи автоматизації. Багато SCADA - систем дозволяють виконувати налагодження системи як в режимі емуляції обладнання, так і з підключеним обладнанням;
- налаштування системи комунікації (мереж, модемів, комунікацій контролерів і т. п.);
- створення баз даних і підключення до них SCADA.

Як система диспетчерського управління SCADA може виконувати наступні завдання:

- взаємодія з оператором (видача візуальної і слухової інформації, передача в систему команд оператора);
- допомога оператору в прийнятті рішень (функції експертної системи);
- автоматична сигналізація про аварії і критичні ситуації;
- видача інформаційних повідомлень на пульт оператора;
- ведення журналу подій в системі;
- вилучення інформації з архіву та подання її оператору в зручному для сприйняття вигляді;
- підготовка звітів (наприклад, роздруківка таблиці температур, графіків зміни операторів, переліку дій оператора);
- облік напруцювання технологічного устаткування.

Основна частина завдань автоматичного управління виконується, як правило, за допомогою ПЛК, однак частина завдань може покладатися на SCADA. Крім того, у багатьох невеликих системах управління ПЛК можуть взагалі бути відсутніми і тоді комп'ютер зі встановленою SCADA є єдиним засобом управління. SCADA зазвичай виконує наступні завдання автоматичного управління:

- автоматичне регулювання;
- управління послідовністю операцій в системі автоматизації;
- адаптація до зміни умов протікання технологічного процесу;
- автоматичне блокування виконавчих пристроїв при виконанні заздалегідь заданих умов.

Знання передісторії керованого процесу дозволяє поліпшити майбутню поведінку системи, проаналізувати причини виникнення небезпечних ситуацій або браку продукції, виявити помилки оператора. Для створення історії система виконує наступні операції:

- збір даних і їх обробка (цифрова фільтрація, інтерполяція, стиснення, нормалізація, масштабування і т. д.);

- архівування даних (дій оператора, зібраних і оброблених даних, подій, аварій, графіків, екранних форм, файлів конфігурації, звітів і т. п.);
- управління базами даних (реального часу і архівних).

Застосування SCADA в системах віддаленого доступу через інтернет різко підвищило вразливість SCADA до дій ворожих осіб. Нехтування цією проблемою може призводити, наприклад, до відмови в роботі мереж електропостачання, життєзабезпечення, зв'язку, відмови морських маяків, дорожніх світлофорів, до зараження води неочищеними стоками і т.п. Можливі й більш важкі наслідки з людськими жертвами або великим економічним збитком. Для підвищення безпеки SCADA використовують такі методи:

- розмежування доступу до системи між різними категоріями користувачів (у змінного оператора, технолога, програміста і директора повинні бути різні права доступу до інформації та до модифікації налаштувань системи);
- захист інформації (шляхом шифрування інформації і забезпечення секретності протоколів зв'язку);
- забезпечення безпеки оператора завдяки його віддаленню від небезпечного керованого процесу (дистанційне керування). Дистанційний контроль і дистанційне керування є типовими вимогами і виконуються по провідній мережі, радіоканалу (через GSM- або радіомодем), через інтернет і т.д.;
- спеціальні методи захисту від кібер-атак;
- застосування міжмережевих екранів.

Оскільки SCADA зазвичай є єдиною програмою для управління системою автоматизації, на неї можуть покладатися також деякі загальносистемні функції:

- здійснення взаємодій між декількома SCADA, між SCADA та іншими програмами (MS Office, базою даних, MATLAB і т.п.);
- діагностика апаратури, каналів зв'язку та програмного забезпечення.

Аналіз властивостей різних SCADA дозволяє вибрати систему, оптимальну для вирішення поставленого завдання. Все різноманіття властивостей SCADA-пакетів можна розбити на наступні групи:

- інструментальні властивості;
- експлуатаційні властивості;
- властивості відкритості;
- економічна ефективність.

До інструментальних відносяться властивості SCADA, що впливають на ефективність роботи системних інтеграторів:

- швидкість розробки проєкту;
- легкість освоєння;
- підтримувані засоби комунікації;
- наявність функцій для складної обробки даних;
- наявність мов МЕК 61131-3 і універсальної алгоритмічної мови типу Visual Basic;
- ступінь відкритості для розробника (підтримка COM і ActiveX для підключення програмних модулів користувача, а також OPC, ODBC, OLE DB;
- якість технічної документації (повнота, ясність викладу, кількість помилок);
- наявність режиму емуляції обладнання для налагодження;
- наявність внутрішніх графічних редакторів, що дозволяють відмовитися від застосування зовнішніх редакторів типу CorelDraw або Photoshop;
- підтримка типових графічних форматів файлів;
- якість технічної підтримки (час реакції на запитання користувачів, наявність "гарячої лінії" технічної підтримки.

Якість SCADA в процесі експлуатації оцінюється кінцевими користувачами і характеризується наступним набором властивостей:

- робастної (нечутливість до помилок користувача, захищеність від вандалів і ворожих елементів, стійкість до помилок у вихідних даних);
- надійність;
- інформаційна захищеність;
- наявність засобів збереження даних при нештатних ситуаціях, відключеннях живлення ;
- наявність автоматичного перезапуску системи при її зависанні або після переривання живлення;
- підтримка резервування SCADA (операторської станції, мережесерверів, клієнтських робочих станцій, резервне копіювання даних);
- підтримка перемикавання екранів з різною деталізацією зображень;
- підтримка декількох моніторів.

Ступінь відкритості дуже сильно впливає на економічну ефективність системи, однак цей вплив носить випадковий характер, оскільки залежить від ступеня використання властивостей відкритості в конкретному проєкті.

Відкритість для програмування користувачем SCADA забезпечується можливістю підключення програмних модулів, написаних користувачем або іншими виробниками. Це зазвичай досягається тим, що SCADA розробляється як контейнер для COM-об'єктів і ActiveX елементів. Сумісність з апаратурою і базами даних інших виробників досягається за допомогою стандарту OPC, застосуванням інтерфейсу ODBC або OLE DB. Відкритість системи програмування досягається підтримкою мов MEK 61131-3. Особливо цікаво з точки зору відкритості застосування веб-інтерфейсу, оскільки він забезпечує доступ до SCADA з будь-якого комп'ютера з будь-якої точки світу, незалежно від апаратної платформи, типу каналу зв'язку, операційної системи і використовуваного веб-навігатора.

Економічну ефективність SCADA можна визначити як відношення економічного ефекту від її впровадження до загальної суми витрат на впровадження і підтримку системи в працездатному стані. На економічну

ефективність в кінцевому рахунку впливають практично всі властивості SCADA, проте в першу чергу можна виділити наступні:

- масштабованість (можливість застосування як для великих, так і для малих систем);
- модульність. Модульність дозволяє зробити замовну комплектацію системи в залежності від поставленого завдання. Типовими модулями можуть бути, наприклад, модуль введення-виведення, модуль візуалізації, модуль аварій, модуль трендів, модуль звітів, модуль комерційного обліку енергоресурсів та ін. ;
- вартість обслуговування;
- умови оновлення версій;
- надійність постачальника, наявність досвіду практичного застосування;
- вартість навчання;
- вартість технічної підтримки;
- методи ціноутворення.

Загальним недоліком універсальних SCADA є їх низька економічна ефективність при використанні для вирішення простих завдань. Незважаючи на те, що ціна SCADA-пакетів істотно знижується при зменшенні кількості доступних користувачеві тегів і набору модулів, залишається високою ціна технічної підтримки. Також дорогою (трудомісткою) залишається адаптація універсальної SCADA до конкретного завдання. Тому ряд фірм пропонують більш вузькоспеціалізовані, але досить прості в налаштуванні мікро-SCADA з скороченою функціональністю [12].

1.3.1 CitectSCADA

CitectSCADA [13] – програмний продукт, що представляє собою повнофункціональну систему моніторингу, управління і збору даних, яка дозволяє забезпечити:

- візуалізацію процесу в графічному режимі;
- «просунуте» управління аваріями;
- відстеження трендів в реальному часі і доступ до архівних трендів;
- підготовку деталізованих звітів;
- статичний контроль процесу;
- багато поточність виконуваних програм, розроблених на CitectVBA і CiCode.

CitectSCADA побудована на базі мультизадачного ядра реального часу, що забезпечує продуктивність збору до 5000 значень в секунду при роботі в мережевому режимі з декількома станціями. Модульна клієнт-серверна архітектура дозволяє однаково ефективно застосовувати CitectSCADA, як в малих проєктах з використанням тільки одного автоматизованого робочого місця (APM), так і у великих з розподілом завдань на кілька комп'ютерів.

У CitectSCADA резервування є вбудованим і легко конфігурованим. Резервування дозволяє захищати всі зони потенційних відмов як функціональних модулів (серверів та клієнтів), так і мережевих з'єднань між вузлами та пристроями введення / виводу. CitectSCADA має вбудовану мову програмування CiCode, а також підтримку VBA (Visual Basic for Applications - спрощена реалізація мови Visual Basic). CitectSCADA працює як 32-розрядний додаток Windows NT / 2000 / XP / 2003 / Vista / 7. Збір даних, формування аварійних повідомлень і побудова трендів відбувається одночасно з редагуванням і компіляцією.

1.3.2 Master SCADA

Master SCADA [14] – програмний пакет для проектування систем диспетчерського управління та збору даних. Основними властивостями є модульність, масштабованість і об'єктний підхід до розробки. Система призначена для збору, архівування, відображення даних, а також для управління різними технологічними процесами. Крім створення так званого верхнього рівня, система дозволяє програмувати контролери з відкритою архітектурою. Таким чином, Master SCADA дозволяє створювати єдиний комплексний проект автоматизації (SCADA-система і ПЛК). Вся система, включаючи всі комп'ютери і контролери, конфігурується в єдиному проекті, в результаті чого не потрібно конфігурувати внутрішні зв'язки в системі. Основними особливостями Master SCADA є:

- єдине середовище розробки всього проекту;
- роздільна конфігурація структури системи і логічної структури об'єкта;
- відкритість і дотримання стандартів;
- необмежена гнучкість обчислювальних можливостей;
- об'єктний підхід.

1.3.3 TRACE MODE

TRACE MODE [15] – інтегрована інформаційна система для управління промисловим виробництвом, яка об'єднує в єдине ціле продукти класу SOFTLOGIC-SCADA / HMI-MES-EAM-HRM. TRACE MODE дає рішення для управління технологічними процесами в реальному часі, що здійснюється в тісній інтеграції з управлінням виробничим бізнесом. На одній платформі об'єднані продукти для автоматизації технологічних процесів і бізнес-процесів (АСУП). Інтегроване середовище розробки TRACE MODE є єдиною програмною оболонкою, що об'єднує всі основні компоненти інструментальної системи:

- SOFTLOGIC - систему програмування контролерів;

- SCADA / HMI - систему розробки розподіленої АСУ ТП;
- MES-EAM-HRM – економічні модулі для створення АСУП, повністю інтегрованих з АСУ ТП.

Основними особливостями TRACE MODE є:

- легкість інтеграції зі стороннім програмним / апаратним забезпеченням;
- інтеграція SCADA і Softlogic систем;
- інтегроване середовище розробки, яка об'єднує в собі більше 10 різних редакторів АСУ ТП і АСУП;
- принцип єдиного проєкту для розподіленої АСУ;
- велика бібліотека драйверів для контролерів і пристроїв зв'язку з об'єктом;
- підтримка мов програмування алгоритмів міжнародного стандарту IEC 61131-3 ;
- власний генератор звітів, що дозволяє створювати повнофункціональні HTML-звіти в реальному часі;
- технології гарячого резервування - дублювання і трювання;
- інтеграція з базами даних і іншими додатками.

1.3.4 Galileo

Galileo [16] - високопродуктивне багатофункціональне середовище розробки, ідеально підходяще для застосувань, близьких до машин і технологічних процесів в машинобудуванні та промисловості. Galileo призначено для використання в усіх галузях і всюди забезпечує інтуїтивну розробку проєкту для всіх графічних пристроїв управління - від панелей оператора Eaton Automation HMI до автономних ПК. Galileo надає розробнику повний набір функцій без обмежень по кількості міток або екранів, і приймає до уваги рівень продуктивності використовуваної панелі. Основними особливостями цього середовища розробки, являються:

- симуляція проєкту на ПК;
- легко освоюваний інтуїтивний графічний інтерфейс з вікном огляду проєкту;
- вбудовані стилі оформлення;
- позиціонування об'єктів за допомогою миші;
- табличні властивості об'єктів, просте і швидке завдання атрибутів за допомогою копіювання і вставки;
- векторні об'єкти індикаторів величин;
- покращена обробка паролів зі складним паролем і термінами дії;
- розширена робота з наборами параметрів (рецептами);
- обробка аварій із записом часу, журналу і підтримкою ілюстрованої діагностики;
- зручна прив'язка тексту і малюнків до змінних;
- безліч графічних об'єктів, наприклад: стовпцеві індикатори, регулятори, графіки;
- список параметрів об'єкта, будь-яка кількість даних на одному екрані;
- динамічна зміна одиниць виміру, наприклад, ° C - ° F, дюйми – мм;
- безліч спеціальних об'єктів і функцій;
- прямий друк з панелі (звіти, форми);
- імпорт з 15 різних форматів малюнків;
- простий імпорт змінних ПЛК;
- зміна мови під час роботи програми;
- підтримка Unicode (включаючи азійські алфавіти);
- імпорт / експорт тексту в форматі XML, наприклад, таблиці Excel;
- завжди доступна повна функціональність без часткових обмежень;
- динамічні об'єкти.

Судячи з вищеописаної інформації кожна SCADA-система орієнтована на свої задачі (табл.1.3). Наприклад, CitectSCADA і Galileo призначені для створення

людино-машинного інтерфейсу, в той час, як Master SCADA і TRACE MODE виконують задачі не тільки створення HMI, а і програмування ПЛК. При виборі середовища розробки людино-машинного інтерфейсу зроблений акцент саме на вузькоспеціалізовані SCADA-системи, тобто CitectSCADA і Galileo. Основним критерієм вибору між ними є питання сумісності з програмованим логічним контролером. Таким чином, була обрана SCADA «Galileo», що розроблена компанією Eaton, яка є виробником обраного ПЛК.

Таблиця 1.3 – Вибір SCADA-системи

Назва	Виробник	Єдине середовище проекту АСУ ТП	Гнучкість інструментів
CitectSCADA	Schneider Electric	—	+
Master SCADA	InSAT	+	+
Trace Mode	AdAstra	+	+
Galileo	Eaton	—	—

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Огляд існуючих людино-машинних інтерфейсів

Галузь АСУ ТП витратила мільярди на підвищення безпечності процесів за допомогою комплексних апаратних систем. Тим не менш, ми до сих пір часто спостерігаємо промислові інциденти та нещасні випадки. Лише декілька випадків привели до вибухів і летальних результатів, але в будь-якому разі усі ситуації являються дорогими з точки зору затримок, зниження якості або ушкодження обладнання. Виходячи з результатів дослідження, консорціум ASM оцінив, що нештатні ситуації обходяться 3-8 % від обороту в рік [17]. Від трьох до восьми відсотків це значні витрати для будь-якого підприємства.

Причини, як правило, не у відказі автоматизованих систем, а у результаті широкого спектру людських помилок. Провідні європейські компанії, такі як ASM, PAS, Rockwell Automation, впевнені в тому, що усунення причин людських помилок і підвищення ситуаційної обізнаності має першорядне значення [1].

Сучасний людино-машинний інтерфейс або НМІ використовується оператором для моніторингу і взаємодії з системою управління, зазвичай оператор має справу з SCADA. Дизайн НМІ грає важливу роль у здатності оператора ефективно керувати роботою, особливо при швидкому виявленні і вирішенні нештатної ситуації, що являється найважливішим завданням оператора.

До появи складних цифрових систем управління людино-машинний інтерфейс оператора зазвичай складався з величезного пульта управління та стіни з мнемосхемою на якій розташовувалися лампи індикації. Приклад такого НМІ зображено на рис.2.1. Вид такої людино-машинної взаємодії мав переваги у представленні всього технологічного процесу, ключових тенденцій і кількості аварійних сигналів. Це в свою чергу давало можливість оператору бачити всю роботу автоматизованого об'єкту та виявляти ріст ненормованих ситуацій. Але в цих системах були і недоліки, зокрема, на рахунок модернізації. Їх було складно і затратно змінювати, а також вони не давали можливості швидко архівувати і

аналізувати дані. Тому такий вид взаємодії був змінений в 1980 - 1990 роках на електронні системи управління (SCADA).

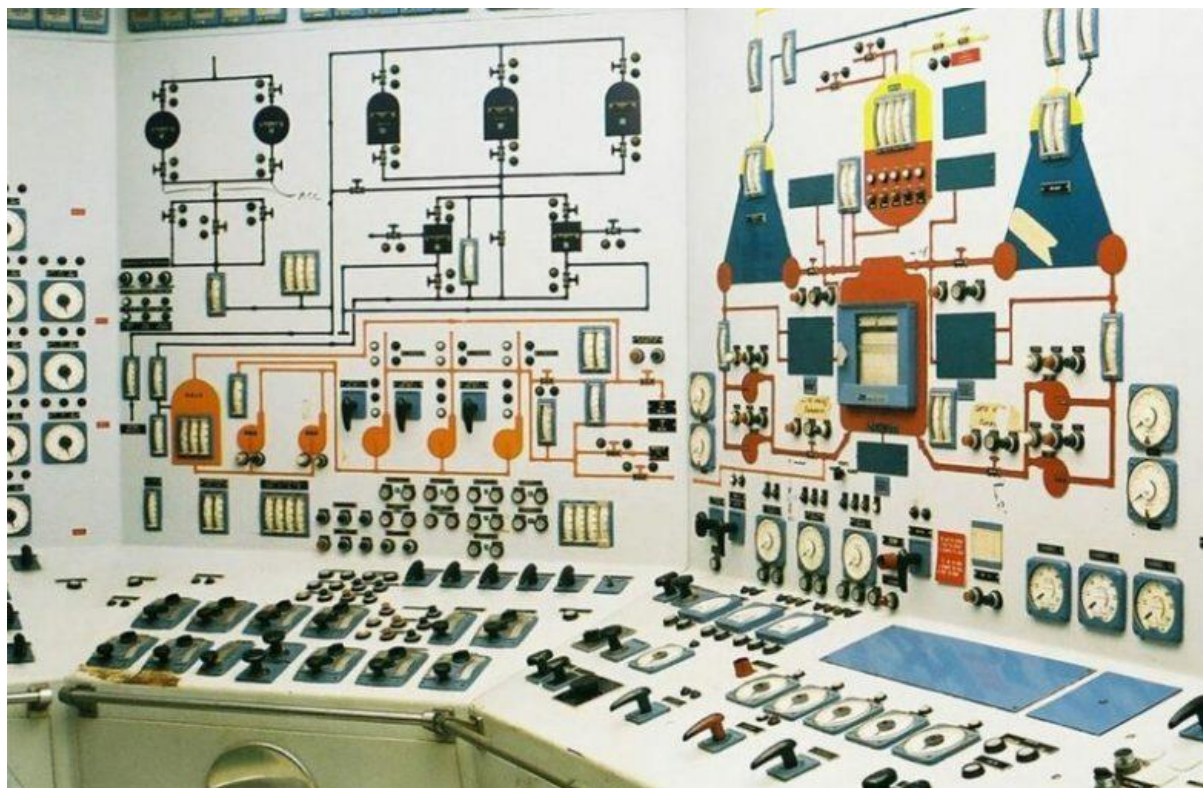


Рисунок 2.1 – Приклад НМІ минулого сторіччя [18]

Нове програмне і апаратне забезпечення давало можливість створювати і відображати графічні елементи для допомоги в управлінні технологічним процесом. Однак тоді не існувало ніяких інструкцій відносно того, як створювати ефективний інтерфейс. Автори книги «The High Performance HMI» Білл Холліфілд і Ян Ниммо зазначають: «Коли здатності комп'ютера до відображення графіки покращились, дизайнери інтерфейсів звернулись до зображення трубопроводу і приборів (мнемосхем), тому що вони були легкодоступні і представлялись, як логічне джерело інформації». Оскільки перші комп'ютерні монітори не могли легко відображати аналогове представлення даних, замість датчиків з'явилися прості числові значення. Також було ускладнене відображення трендів, замість них використовувалися аварійні повідомлення, але кількість повідомлень зросла до тих пір, поки вони не стали захаращувати один одного. Обмежена кольорова палітра

моніторів тих часів, теж давалась у знаки, кольори використовувались не послідовно і більшість інтерфейсів мали чорний фон. Внаслідок цього, рівень освітленості у кімнаті управління був понижений, що призвело до підвищеної втоми операторів. В результаті з'явився НМІ, приклад якого зображений на рис.2.2.

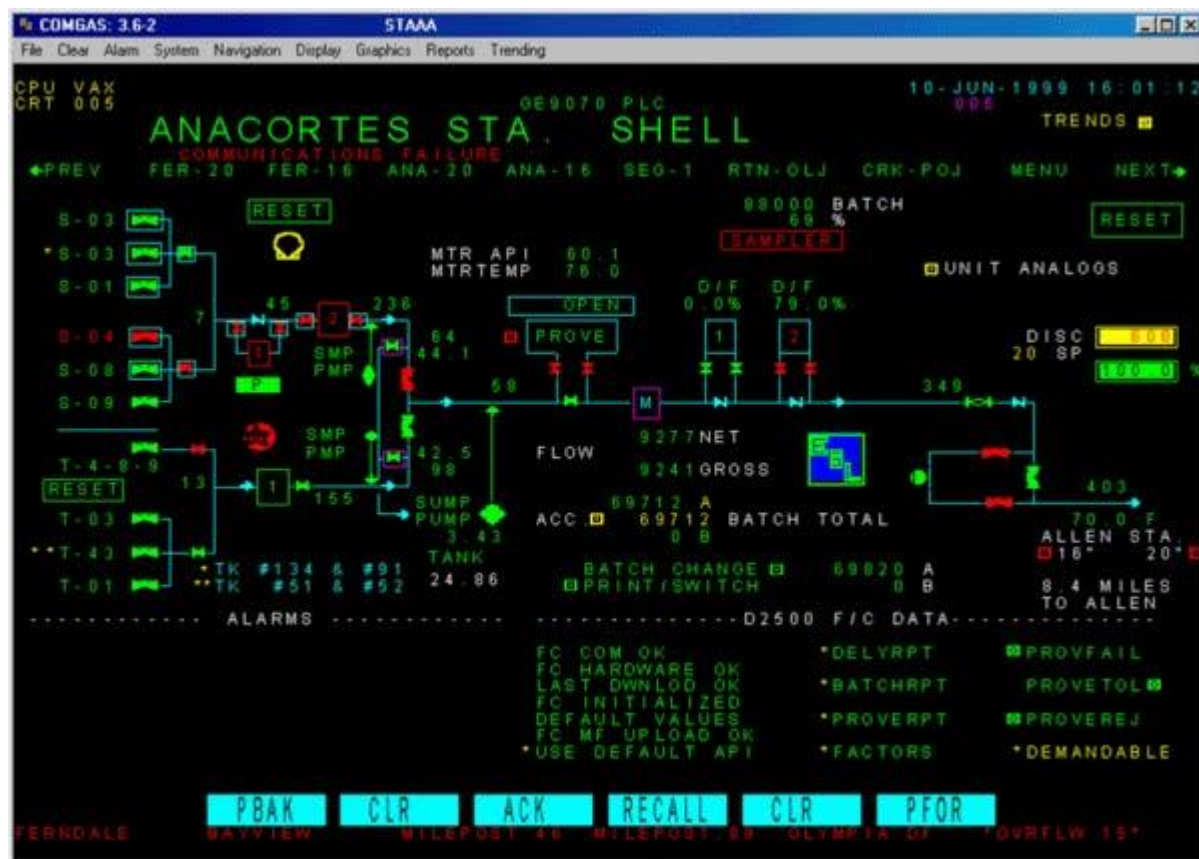


Рисунок 2.2 – Типовий екран НМІ 90-х років [19]

Такі НМІ були розроблені більше 20 років назад і залишаються звичайним явищем у всій галузі. Інженери і оператори звикли до такого стилю графіки і на основі цього стилю сформувалося помилкове розуміння, як повинен виглядати НМІ, що в переважній кількості випадків, не зазнає змін. Внаслідок цього, склалася ситуація, коли усі галузі промисловості використовують дорогі системи управління з примітивними НМІ, які були створені декілька десятиліть тому. Це дає підстави вважати, що інерція, а не вартість являється однією з основних проблем покращення людино-машинного інтерфейсу.

З часом апаратні і програмні можливості систем управління зросли. На зміну обмеженості кольорів прийшло їх розмаїття, що негативно позначилося на ергономіці та інформативності людино-машинних інтерфейсів (рис.2.3). Тенденція зображення мнемосхем та даних в необробленому вигляді залишилась. Додалося не послідовне застосування яскравих кольорів і недоречне застосування тривимірних об'єктів, що більше підходить для маркетингових цілей. Багато компаній навіть зараз створюють інтерфейси подібні цьому прикладу. В результаті ми отримуємо гарний на вигляд НМІ, але абсолютно не ефективний для контролю технологічного процесу.

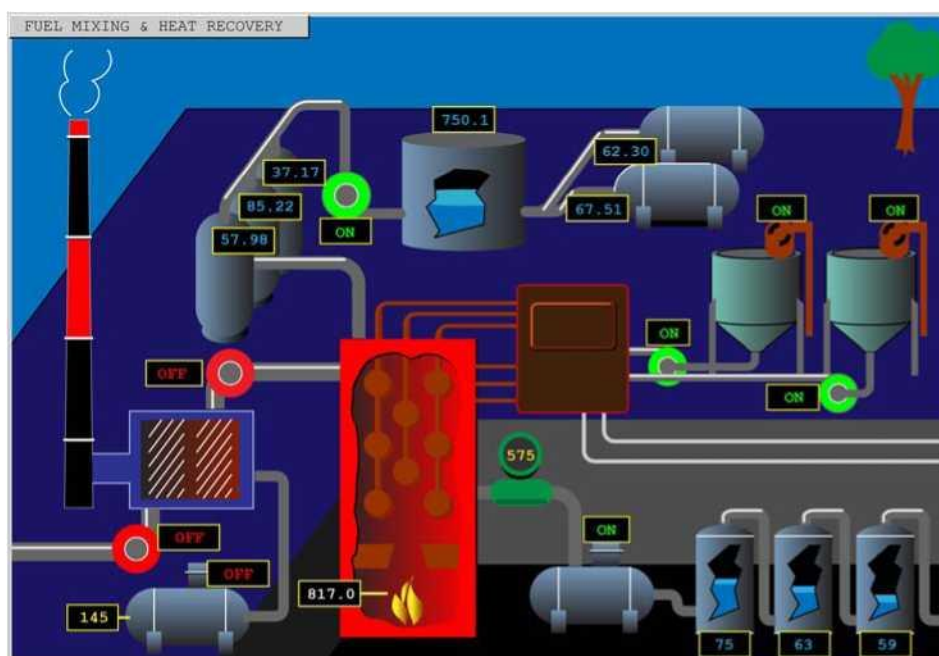


Рисунок 2.3 – НМІ з зухвалим використання графіки [1]

На даному рисунку 90 % екранного простору займають тривимірні об'єкти, яскраво забарвлені технологічні лінії, анімація вогню, що заважає розгледіти корисну інформацію у вигляді індикаторів стану. Оператор не зможе швидко і точно визначити, в якому стані перебуває процес, тому що даний НМІ не розрахований на такі цілі.

З приводу створення людино-машинних інтерфейсів інженерам з автоматизації є на кого рівнятись і в кого позичати досвід. Гарним прикладом є

досягнення в комерційній галузі UX / UI інженерів, зусиллями яких був створений «flat» дизайн, який на даний момент є наближеним до стандарту з створення людино-машинної взаємодії. Плоский дизайн є протилежністю до реалізму, його основою являються прості форми з двовимірними зображеннями без ефектів, що робить більший акцент на текстовому та графічному контенті (рис.2.4). Зрозуміла структура та лаконічні форми роблять інтерфейс інтуїтивно зрозумілим, що в свою чергу дає змогу швидко знайти потрібну інформацію і здійснити цільову дію.

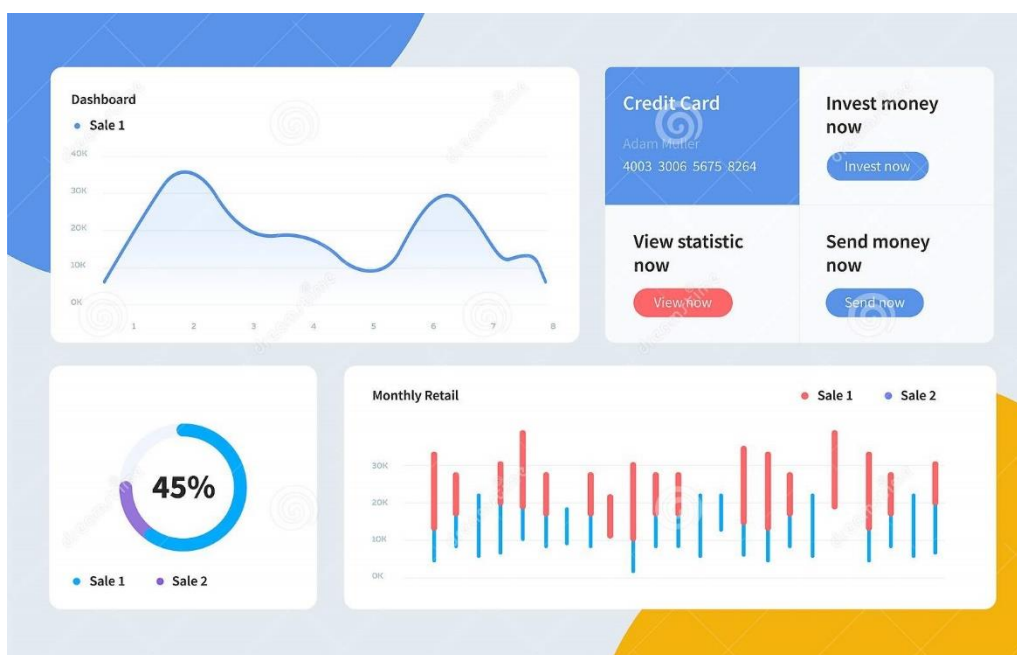


Рисунок 2.4 – Приклад додатку з «flat» дизайном [20]

Якщо проводити аналогію з галуззю автоматизації, то у більшості сучасних промислових інтерфейсів основний акцент робиться на зображенні агрегату та його механічних частин, але, наприклад, водієві не обов'язково знати, як рухаються шестерні в двигуні, для того, щоб керувати автомобілем.

В результаті усього вищезазначеного, можна описати основні недоліки більшості сучасних НМІ:

- відображення даних у вигляді чисел;
- низька кількість графіків або вони відсутні;
- анімація технологічного забезпечення промислових об'єктів;

- яскраве кольорове тривимірне обладнання;
- деталізовані зображення агрегатів;
- не послідовне використання кольорів для індикації;
- обмежена навігація по екранам;
- відсутність ієрархії відображення.

2.2 Вимоги до розробки людино-машинних інтерфейсів

Для ефективної взаємодії людини з технологічним процесом, НМІ повинен будуватися за такими принципами :

- використання кольору спеціально і послідовно;
- сірий фон для мінімізації проблем з відблисками та відображенням;
- відсутність анімацій, за винятком аварійних сигналів;
- графіки важливих параметрів;
- аналогове представлення даних, з визначеними межами нормальних, попереджувальних і аварійних умов;
- прості та зрозумілі двовимірні зображення;
- логічні та послідовні методи навігації;
- мінімізація помилок введення даних;
- заходи перевірки та безпеки.

2.2.1 Послідовне використання кольору

Забезпечення надійного управління та оперативного втручання – основні завдання персоналу, який здійснює контроль і управління устаткуванням та технологічними процесами. Інформація, що надходить від контрольних пристроїв, повинна відповідати потребам операторів, контролюючих та керуючих роботою обладнання, систем обладнання і технологічних процесів.

Кольори повинні бути легко ідентифіковані і відмінні від кольору фону і будь-яких інших певних кольорів. Деякі кольори повинні бути зарезервовані для позначення сигналів безпечного застосування. Рекомендується, щоб число використовуваних кольорів зводилося до мінімуму [2]. Кожен колір повинен мати певне смислове значення (табл.2.1), навіть при використанні великої кількості відтінків основних кольорів (табл.2.2). Екрани інтерфейсу при нормальному стані технологічного процесу, не повинні містити яскравих насичених кольорів. Наприклад, яскраво-червоне і зелене відображення обладнання(). Червоний та жовтий кольори використовуються виключно для аварійної та попереджувальної сигналізації відповідно.

Як вже було зазначено вище, через апаратні обмеження у 80-90-х роках, фоновим кольором НМІ був чорний і для запобігання відблисків на екрані, освітлення в приміщенні було зазвичай приглушене, що в свою чергу призводило до втоми операторів. Тому, найкращим рішенням, з точки зору запобігання появи відблисків, при яскравому освітленні, є застосування сірого кольору фону.

Таблиця 2.1 – Призначення кольорів за стандартом МЕК 60073-2000 [21]

Колір	Смислове значення		
	Безпека людини або обладнання	Стан процесу	Стан обладнання
Червоний	Небезпека	Критичний	Несправність
Жовтий	Увага	Перехідний	Перехідний
Зелений	Безпека	Нормальний	Нормальний
Синій	Спеціальне (може мати будь-яке значення окрім функціонального для червоного, жовтого, зеленого)		
Білий або Сірий	Не мають спеціального значення		

Таблиця 2.2 – Призначення кольорів згідно [1]

Колір	Використання
Сірий	Загальний графічний фон
Білий	Виділення декотрих дрібних елементів
Світло-сірий	Індикація вмикання обладнання
Сірий	Індикація вимикання обладнання
Темно-сірий	Текст, технологічні лінії
Чорний	Текст, маркування, контури обладнання
Темно-синій	Спеціальне значення, лінії графіків
Темно-зелений	Спеціальне значення, лінії графіків
Світло-зелений	Спеціальне значення, лінії графіків
Блакитний	Робочі діапазони
Коричневий	Лінії графіків
Блідо-червоний	Декотрі покази
Червоний	Аварія, перший пріоритет
Жовтий	Попередження другого пріоритету
Помаранчевий	Попередження третього пріоритету
Пурпурний	Четвертий пріоритет, для діагностики
Темно-пурпурний	Лінії графіків

Числова інформація повинна відрізнятись від статичного тексту. Рішенням цього питання являється використання темно-синього кольору, що добре співвідноситься с сірим фоном. У разі, якщо числа стоять поряд, їх потрібно вирівнювати за десятковою комою.

Відображення сигналів оповіщення також повинно бути закодовано кольором та формою (рис.2.5).

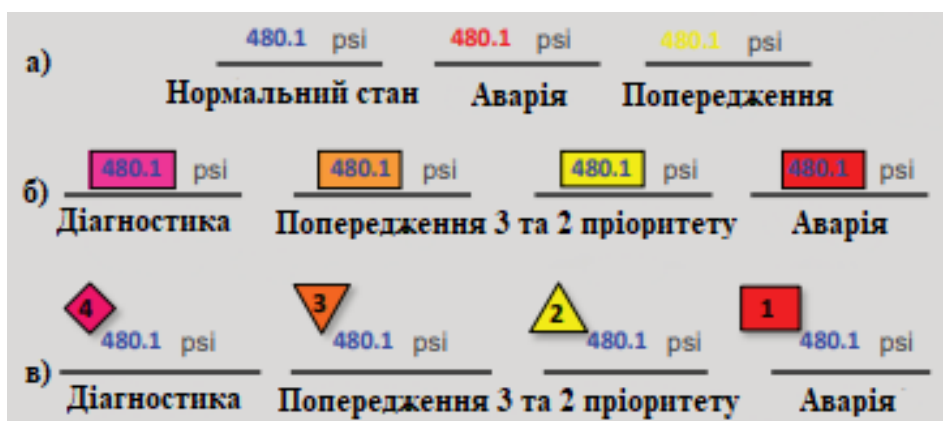


Рисунок 2.5 – Сигнали оповіщення

Розглянемо варіанти зображення сигналів оповіщення про стан процесу. На даному рисунку рядок в) краще сприймається людиною при виникненні сигналу оповіщення на відміну від забарвлених чисел у рядках а) і б).

2.2.2 Представлення даних

Представлення даних у вигляді аналогових значень успішно використовувалось до переходу на електронні системи управління. За рахунок цього оператори мали ситуаційну обізнаність при управлінні технологічним процесом. Аналогові значення є потужним інструментом, оскільки люди інтуїтивно краще розуміють аналогові зображення. Доказ цього розташовується на стіні майже в кожному домі та офісі, людям зручніше розподіляти свій час за стрілочним годинником, хоча у кожного є цифровий варіант на дисплеї телефону.

В книзі «Situation Awareness Analysis and Measurement», доктор Міка Єндслі визначає, три кроки для досягнення ситуаційної обізнаності:

- оператор повинен сприймати важливі данні;
- оператор повинен розуміти поточну ситуацію;
- оператор повинен передбачити майбутній стан процесу.

Також є корисною її цитата: «Справжня обізнаність про ситуацію існує тільки в свідомості людини-оператора. Таким чином, надання великої кількості даних буде не достатньо, якщо вони не будуть успішно і своєчасно передані,

сприйнятті і засвоєні» [22]. Основною перевагою покращення існуючих НМІ, являється представлення даних у аналоговому вигляді і їх сортування за пріоритетом.

Якщо ж звернутись до сучасних інтерфейсів, то ми отримаємо ситуацію (рис.2.6), коли людина, яка не пройшла спеціальну підготовку не може однозначно відповісти на питання про стан процесу, чи проходить він з піковою ефективністю або ж виходить за межі нормального.

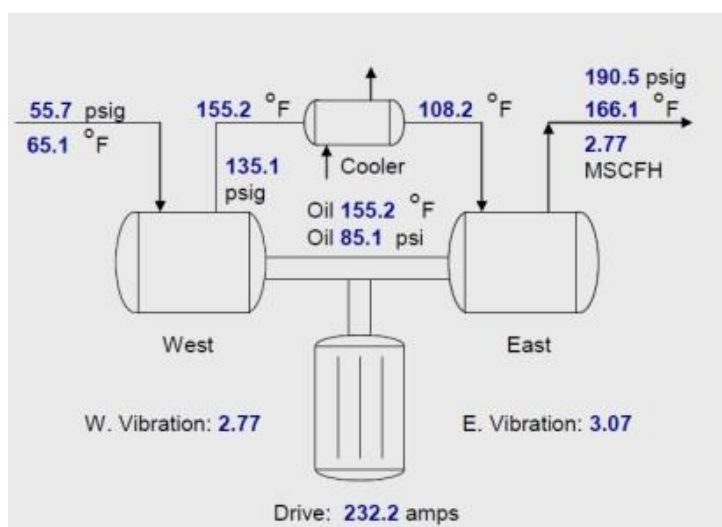


Рисунок 2.6 – Зображення даних, яке заважає сприйманню інформації [1]

В даному випадку оператор має контролювати велику кількість числових даних порівнюючи кожне число з діапазонами в пам'яті. Таким чином усе залежить від кількості ненормованих ситуацій на досвіді оператора. Навчання ж нового оператора виявляється складним та повільним процесом, а модернізація виробництва тільки посилить необізнаність людини у ситуації. Як видно з (рис.2.7), додавання діапазонів з сортуванням по рівням безпеки виправляє недолік сучасних інтерфейсів.

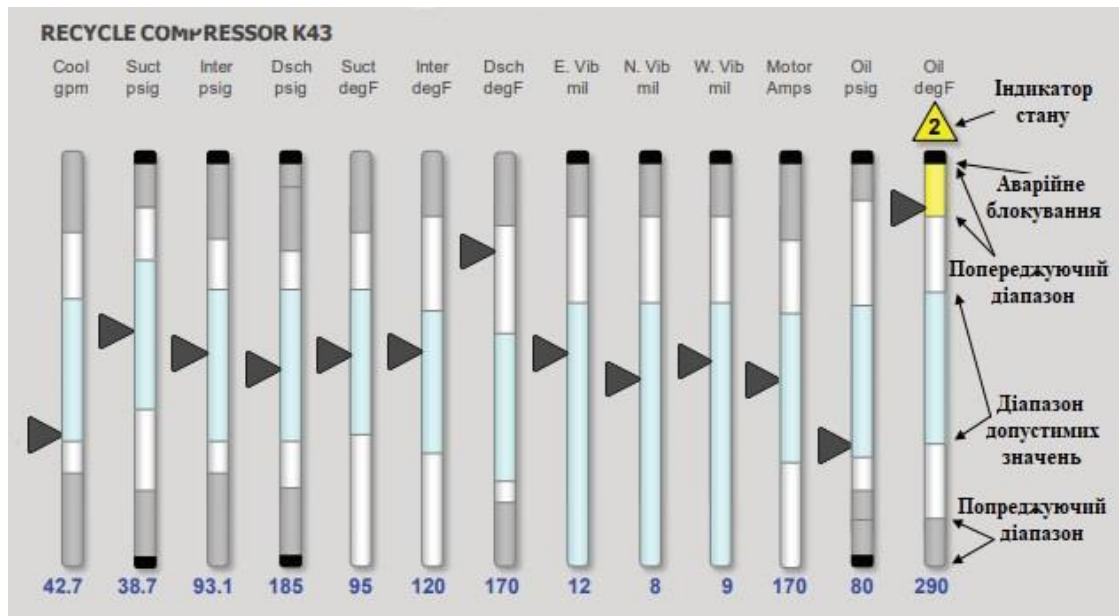


Рисунок 2.7 – Зображення даних у аналоговому вигляді [1]

Одним лише поглядом на дане зображення, навіть людина без досвіду управління поточним технологічним процесом, може дати конструктивну оцінку його стану. Процес в цілому у нормальному стані, але треба приділити більшу увагу першому зліва, середньому, передостанньому та останньому стовпцю. Як видно з (рис.2.7) показники середнього і останнього стовпця знаходяться в області високого діапазону значень, а показники першого та передостаннього стовпця можуть з часом увійти в діапазон низьких значень. Знання про те, що є нормальним, вже закладено в даний НМІ, тобто оператору не потрібно порівнювати значення з діапазонами у своїй пам'яті. В той же час, оператор має змогу спрогнозувати майбутній стан системи, якою він керує. Таким чином можна полегшити процес навчання нового спеціаліста, та полегшити виявлення ненормованої ситуації, ще до її виникнення.

2.2.3 Ієрархія екранів людино-машинного інтерфейсу

Проблемою людино-машинних інтерфейсів, які основані на мнемосхемах є їх розірвана структура, адже повна схема процесу, зазвичай, не поміщається на одному екрані. Таким чином оператор не має змоги бачити весь технологічний процес. Цю проблему вирішує ієрархія рівнів, що поділяють відображувану інформацію (рис.2.8).

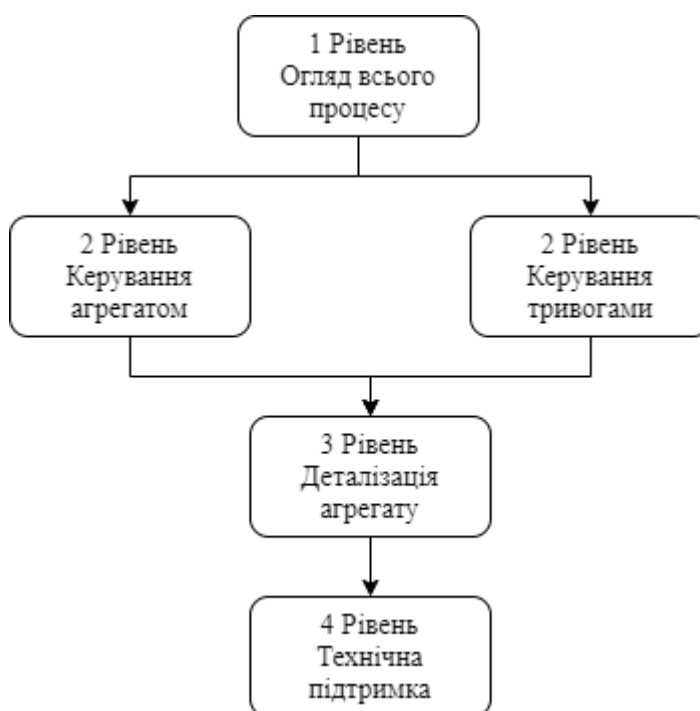


Рисунок 2.8 – Ієрархія екранів НМІ

З екрану першого рівня не виконуються управляючі дії, він містить короткий огляд системи, та повинен складатися з такої інформації :

- сигнали оповіщення;
- графіки важливих технологічних параметрів;
- аналогове зображення технологічних параметрів;
- стан основного обладнання.

Екран другого рівня використовується для моніторингу і управління конкретним агрегатом, та повинен містити:

- елементи керування;
- значення датчиків і параметрів;
- стан датчиків та агрегату;

Деталізоване зображення і детальна інформація агрегату відноситься до третього рівня ієрархії, він може містити:

- мнемосхеми;
- окремі частини обладнання;
- зображення інформації, що не є критичною у часі.

Для складних систем може вводитися четвертий рівень – підтримка агрегату. Тут розміщують методичні вказівки, документацію і керівництво з діагностики несправності.

3 РОЗРОБКА ПРОГРАМИ

3.1 Розробка структури автоматизованої системи управління технологічним процесом

Стандарт ISO 9241 [3], що розглянутий в монографії [23] містить керівництво з людино-орієнтованого проєктування комп'ютерних інтерактивних систем, що дозволяє уникнути негативного впливу ряду факторів, пов'язаних з фізичними та когнітивними особливостями людей, які взаємодіють з системою управління.

Характеристики користувачів і завдань (табл.3.1), а також організаційна, технічна і фізична середа визначають умови використання майбутньої системи. Корисно збирати й аналізувати інформацію про поточні умови, щоб зрозуміти, а потім визначити умови, в яких буде використана майбутня система [3].

Таблиця 3.1 – Завдання користувачів системи управління

№	Завдання	Користувач
1	Моніторинг ходу технологічного процесу	Оператор-технолог
2	Налаштування параметрів технологічного процесу	Оператор-технолог
3	Керування пуском і зупинкою технологічного процесу	Оператор-технолог
4	Аналіз динаміки технологічного процесу	Оператор-технолог
5	Налаштування датчиків і обладнання	Оператор-технолог
6	Налаштування датчиків і обладнання	Інженер з автоматизації
7	Контроль кабельної системи	Інженер з автоматизації
8	Виявлення характеру несправності та її локалізація	Інженер з автоматизації

Технологічний процес, що автоматизується є частиною виробничого комплексу підприємства з виготовлення без тротильових вибухових речовин. Об'єктом автоматизації є лінія виготовлення емульсійного розчину, яка поділяється на три ділянки: вузол прийому і зберігання рідких окислювачів; вузол приготування розчину окислювача; вузол охолодження емульсійної композиції. В свою чергу ділянки мають відповідне технологічне забезпечення (рис.3.1,3.2).

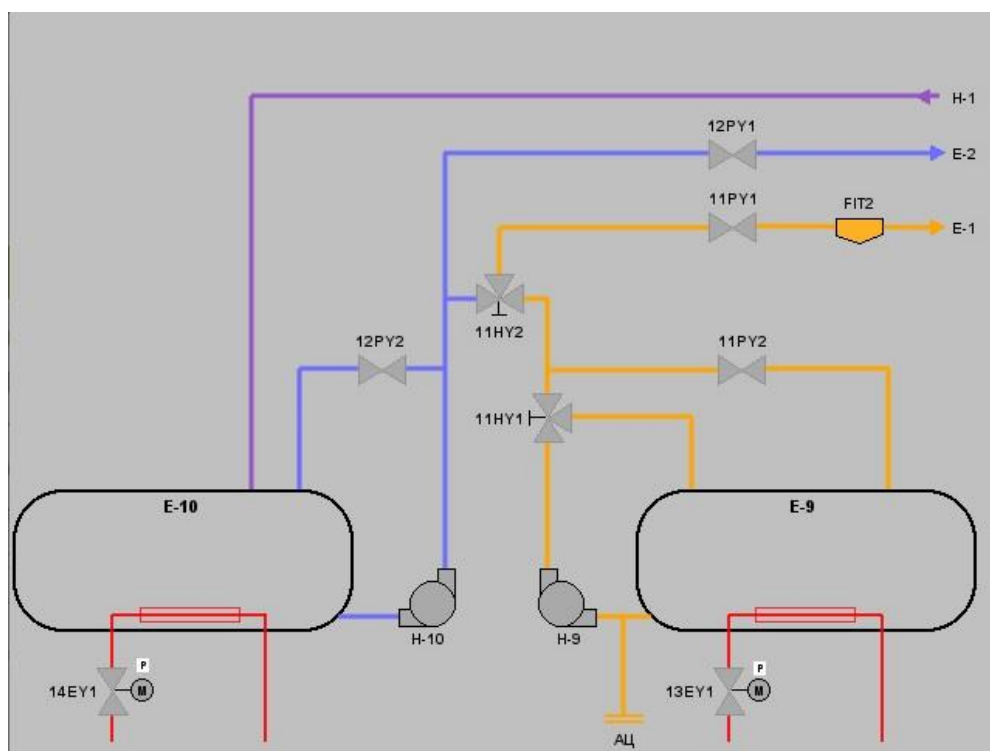


Рисунок 3.1 – Вузол прийому і зберігання рідких окислювачів

Вузол прийому і зберігання рідких окислювачів має в своєму складі дві ємності з обігрівом – E-10, E-9. Температура в ємностях регулюється двома електропозиційними клапанами – 14EY1, 13EY1. Для подачі рідких окислювачів у вузол приготування розчину використовуються насоси – H-10 і H-9. FIT2 – витратомір, що використовується для дозування рідини. Пневмоклапани 12PY2 і 11PY2 застосовуються для циркуляції рідини в ємностях. Трипозиційні ручні клапани 11HY1 та 11HY2 задають напрямок потоку рідини, а пневмоклапани 12PY1 і 11PY1 є обмежувачами.

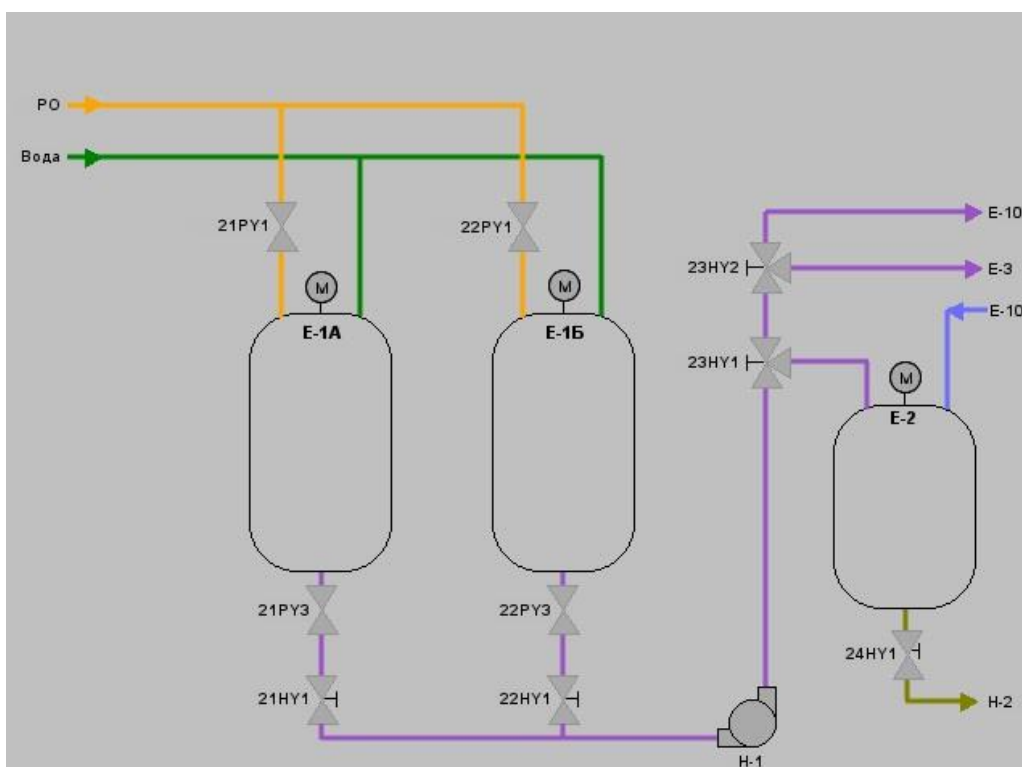


Рисунок 3.2 – Вузол приготування розчину окислювача

Рідкий окислювач подається в ємності через пневмоклапани 21PY1 і 22PY1. Приготування розчину виконується за допомогою ємностей з вертикальним міксерами – E-1A, E-1B, E-2. Переміщення рідини здійснюється за допомогою насосу – H-1, та керується двома донними пневмоклапанами – 21PY3, 22PY3, і двопозиційними та трипозиційними ручними клапанами 21HY1, 22HY1, 23HY1, 23HY2, 24HY1, відповідно.

Технологічне забезпечення (рис.3.3) вузла охолодження емульсійної композиції складається з градирні, де відбувається охолодження води за рахунок направленого потоку повітря, ємності для збору води – E-8, і теплообмінника – ТО-1, який охолоджує емульсійну композицію. Для циркуляції води в системі використовується насос – H-8. Двопозиційні клапани 31HY1, 32HY1, 32HY2 – обмежуючі.

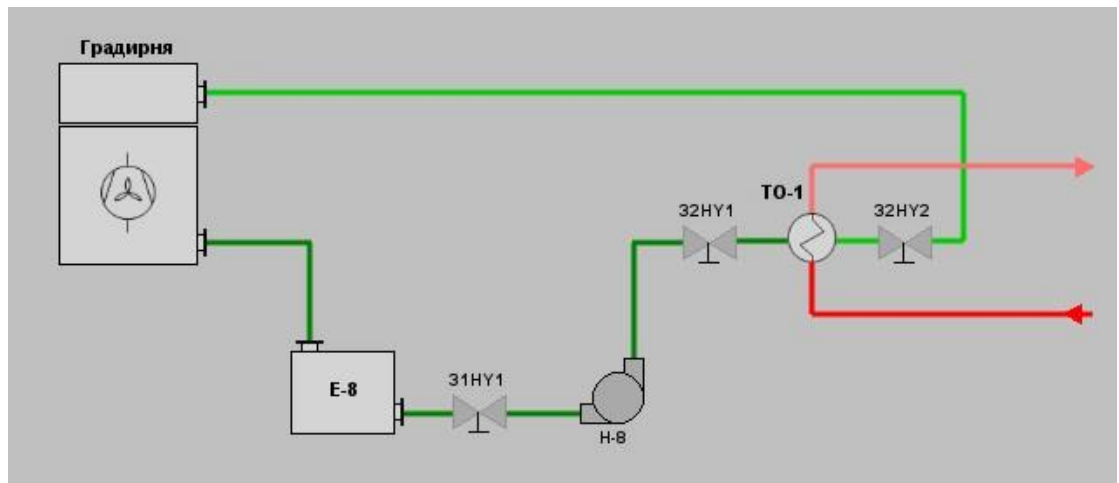


Рисунок 3.3 – Вузол охолодження емульсійної композиції

На основі опису умов використання, сучасних досягнень в предметній області, вимог стандартів і керівництв з проектування була спроектована АСУ ТП з трирівневою структурою, де на середньому та високому рівні розташовуються: шафа управління (рис.3.4) з панелью оператора, що має функцію ПЛК, та панельний ПК з SCADA-системою, відповідно. За підсумками основні параметри системи становлять : 120 вхідних цифрових сигналів; 45 вихідних цифрових сигналів; 16 вхідних аналогових сигналів.

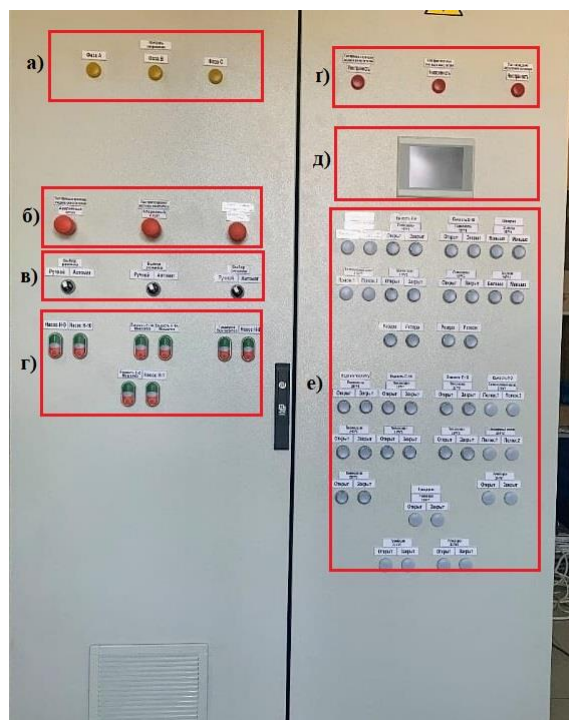


Рисунок 3.4 – Шафа управління

На рис.3.4 зображена шафа управління, де: а – індикація мережі; б - аварійні кнопки розподілені за вузлами; в – перемикачі режимів автоматичний/ручний розподілені за вузлами; г – управління двигунами насосів, міксерів і градирні; г – індикація несправностей розподілена за вузлами; д – панель оператора з функцією ПЛК; е – управління клапанами. Шафа управління виконана згідно [2].

3.2 Розробка функціональної схеми автоматизованої системи управління технологічним процесом

Для керування технологічним процесом використовується програмований логічний контролер XV-102 з функцією панелі оператора (рис.3.4д) та модулі вводу/виводу від компанії Eaton (Moeller). ПЛК зв'язується з модулями вводу/виводу за допомогою мережевого промислового протоколу верхнього рівня CANopen, що використовує стандарт CAN в якості мережевого та транспортного рівнів. В свою чергу модулі вводу/виводу мають в своєму складі аналогові та дискретні входи і виходи.

Функціональні схеми системи управління розроблені в середовищі «CoDeSys» на мові сходових діаграм (LD) з використанням мови структурного тексту (ST) для оголошення екземплярів функціональних блоків і ініціалізації. CoDeSys – це сучасний інструмент для програмування контролерів (назва CoDeSys похідна від Controllers Development System). Середовище розробки надає інструменти для програмування на мовах стандарту МЭК 61131-3. Використовувані редактори та відладочні засоби базуються на широко відомих принципах, які добре себе зарекомендували та знайомі за іншими популярними середовищами професійного програмування (наприклад, Visual C++). Мінімальні системні вимоги для використання середовища розробки CoDeSys наведені в табл.3.1.

Таблиця 3.1 – Системні вимоги CoDeSys [24]

Операційна система	Процесор	Оперативна пам'ять	Жорсткий диск
Windows 7 (SP1)	Intel Atom 2,2 ГГц	4 Гб	HDD 2 Гб

Система, що автоматизується має велику кількість зворотних зв'язків, в цьому випадку мова сходових діаграм має перевагу над більшістю інших, як вже зазначалося вище. Мова LD дає змогу швидше орієнтуватись у алгоритмі при його безпосередньому виконанні та дозволяє ефективніше вести пусконаладжувальні роботи.

Перед початком роботи з вхідними та вихідними сигналами, їх потрібно поставити у відповідність змінним, що будуть використовуватись при побудові функціональної схеми, як це показано на рис.3.5.

Змінна	Соотнесение	Канал	Адрес	Тип	Значение
		ModuleDiag : A22_AI	%IW9	WORD	
Tank_9_Level		InputChannel1 : A22_AI	%IW10	WORD	
Tank_9_Temperature		InputChannel2 : A22_AI	%IW11	WORD	
Pump_H9_Pressure		InputChannel3 : A22_AI	%IW12	WORD	
AI_reserve4		InputChannel4 : A22_AI	%IW13	WORD	
Tank_10_Level		InputChannel5 : A22_AI	%IW14	WORD	
Tank_10_Temperature		InputChannel6 : A22_AI	%IW15	WORD	
Pump_H10_Pressure		InputChannel7 : A22_AI	%IW16	WORD	
AI_reserve8		InputChannel8 : A22_AI	%IW17	WORD	
		WireBreakDiag : A22_AI	%IW18	WORD	

Рисунок 3.5 – Співвідношення змінних до аналогових сигналів на вході

3.2.1 Обробка аналогових сигналів

Обробка аналогових сигналів виконується за допомогою користувацьких функціональних блоків. Для початку роботи з функціональним блоком потрібно оголосити його екземпляри, які мають відповідний тип даних. На рис.3.6 це зроблено на мові ST.

```

1  PROGRAM Analog
2  VAR
3      idB_Press_Pump_9: Analog_config;
4      idB_Press_Pump_10: Analog_config;
5      idB_Press_Pump_1: Analog_config;
6      idB_Press_Pump_8: Analog_config;
7
8      idB_Analog_Tank_9: Analog_config;
9      idB_Level_Tank_9: Level_USonic_config;
10     idB_Analog_Tank_10: Analog_config;
11     idB_Level_Tank_10: Level_USonic_config;
12
13     idB_Press_Tank_1A: Analog_config;
14     idB_Level_Tank_1A: Level_Hidro_config;
15     idB_Press_Tank_1B: Analog_config;
16     idB_Level_Tank_1B: Level_Hidro_config;
17     idB_Press_Tank_2: Analog_config;
18     idB_Level_Tank_2: Level_Hidro_config;
19
20     idB_Temp_Tank_9: Analog_config;
21     idB_Temp_Tank_10: Analog_config;
22     idB_Temp_Tank_1A: Analog_config;
23     idB_Temp_Tank_1B: Analog_config;
24     idB_Temp_E8: Analog_config;
25     idB_Temp1_TO: Analog_config;
26     idB_Temp2_TO: Analog_config;
27
28     Tank_1A_Level_Min: BOOL;
29     Tank_1A_Level_Max: BOOL;
30     Tank_1B_Level_Min: BOOL;
31     Tank_1B_Level_Max: BOOL;
32
33 END VAR

```

Рисунок 3.6 – Оголошення екземплярів функціонального блоку

В даному випадку визначення функціонального блоку подібне до визначення типу даних в рядках 28-31, зліва знаходяться екземпляри, а з права ім'я функціонального блоку. Кожен екземпляр функціонального блоку отримує свій власний ідентифікатор і дані, що містять вхідні і вихідні змінні.

Реалізацію обробки аналогових сигналів розглянемо на прикладі вхідного сигналу від датчика температури, що знаходиться в ємності Е-9. На рис.3.7 зображено екземпляр «idB_Temp_Tank_9» функціонального блоку «Analog_config» і його вхідні і вихідні параметри.

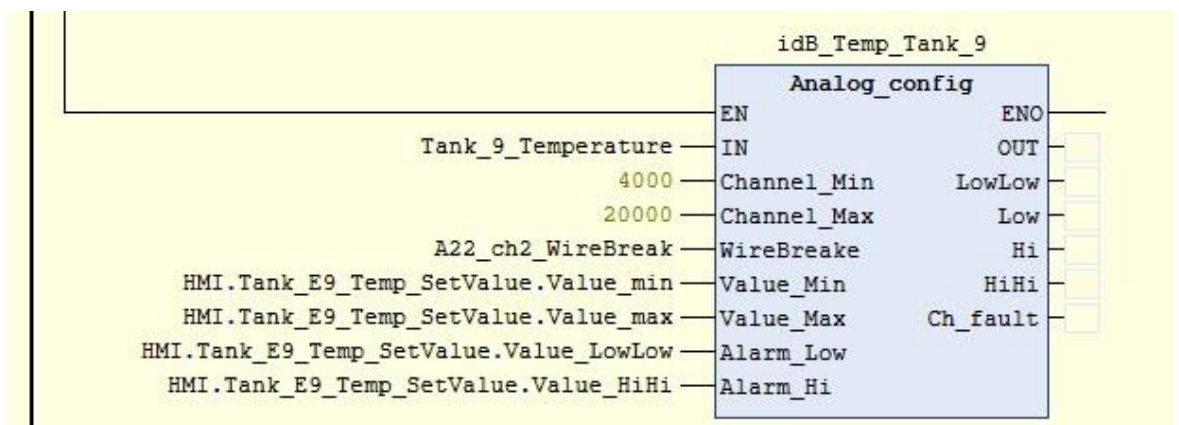


Рисунок 3.7 – Функціональний блок обробки сигналу температури

Вхідні параметри:

- «EN» - сигнал, що дозволяє роботу ФБ;
- «IN» - аналоговий сигнал;
- «Channel_Min», «Channel_Max» - межі аналогового сигналу (в мкА);
- «WireBreak» - сигнал обриву мережі;
- «Value_Min», «Value_Max», «Alarm_Low», «Alarm_Hi» - уставки значень.

В свою чергу вхідні параметри приймають такі змінні, як :

- Tank_9_Temperature (значення аналогового сигналу від датчика температури E-9);
- A22_ch2_WireBreak (сигнал стану з'єднання з двадцять другим модулем вводу);
- HMI.Tank_E9_Temp_SetValue.Value_Min (уставка мінімального значення температури);
- HMI.Tank_E9_Temp_SetValue.Value_Max (уставка максимального значення температури);
- HMI.Tank_E9_Temp_SetValue.Value_LowLow (уставка аварійно низької температури);
- HMI.Tank_E9_Temp_SetValue.Value_HiHi (уставка аварійно високої температури).

Вихідні параметри:

- «ENO» - сигнал стану ФБ;
- «OUT» - величина в відповідних одиницях вимірювання;
- «LowLow», «Low», «Hi», «HiHi» - сигнали оповіщення (попереджувачі та аварійні);
- «Ch_fault» - обрив дроту датчика.

3.2.2 Керування пневматичними клапанами

Керування клапанами здійснюється за допомогою кнопок на передній панелі шафи управління (рис.3.4е), а також за допомогою SCADA-системи. Функціональна схема управління клапанами зображена на рис.3.8а, 3.8б.

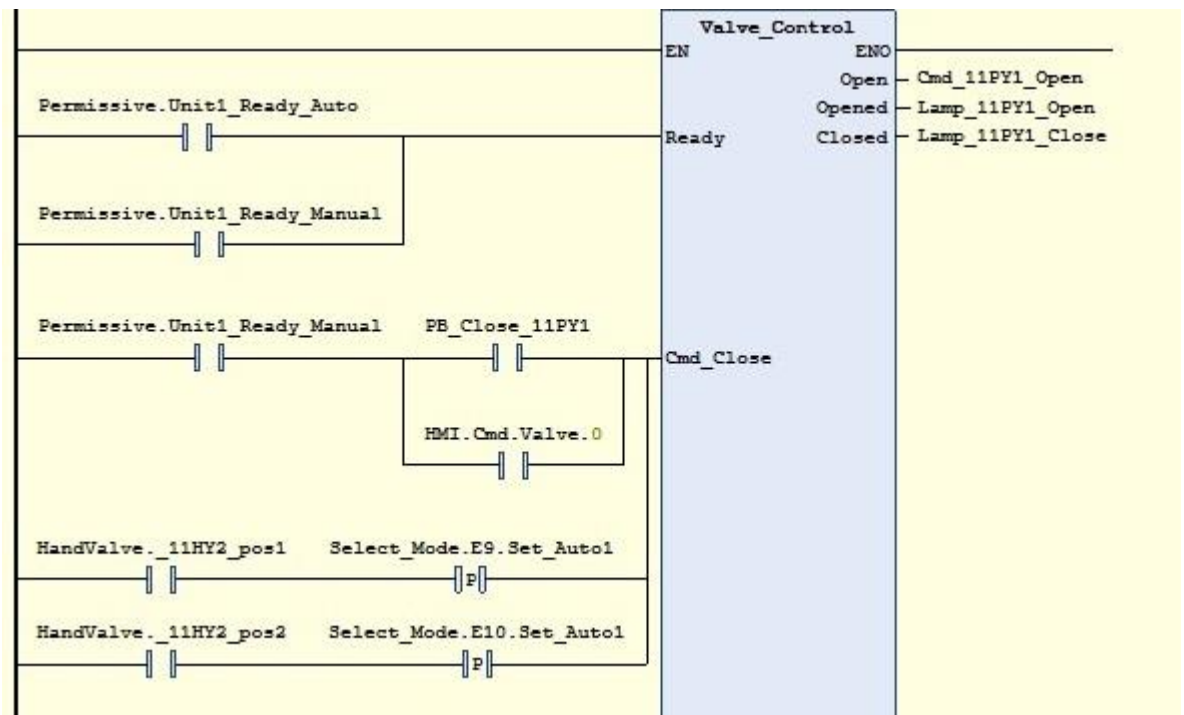


Рисунок 3.8а – Функціональна схема управління пневматичним клапаном

Розглянемо вхідні і вихідні параметри ФБ, а також ланцюги, що під'єднані до входів функціонального блоку. До входу, що визначає готовність - «Ready», під'єднані дозволяючі сигнали з умовою «АБО». Вхід, що визначає команду на закриття - «Cmd_Close» спрацьовує в наступних випадках:

- якщо була натиснута кнопка на шафі управління «АБО» в SCADA-системі «I» вибраний ручний режим;
- якщо ручний клапан 11HY2 знаходиться в положенні 1 «I» був вибраний автоматичний режим 1 для ємності E-9;
- якщо ручний клапан 11HY2 знаходиться в положенні 2 «I» був вибраний автоматичний режим 1 для ємності E-10.

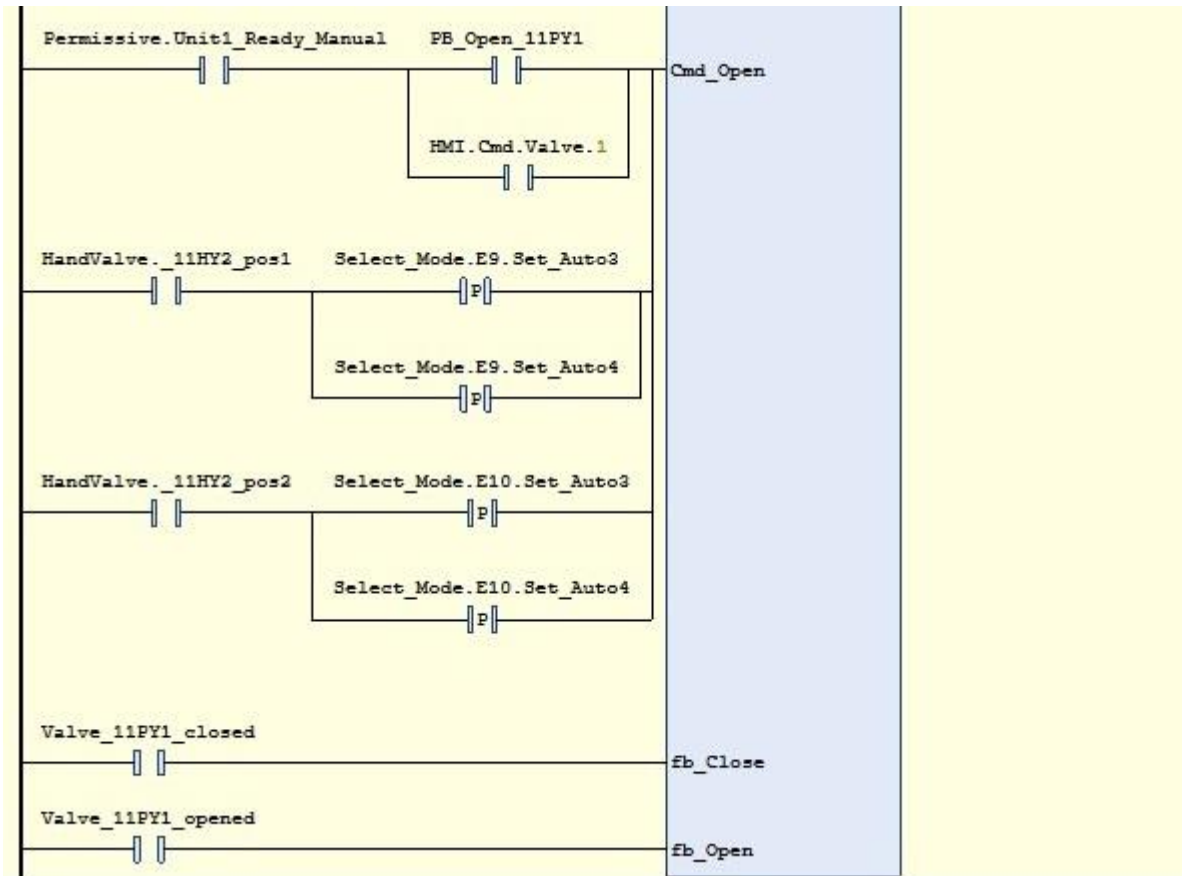


Рисунок 3.8б – Продовження функціональної схеми управління пневматичним клапаном

Вхід, що визначає команду на відкриття - «Cmd_Open» спрацьовує в наступних випадках:

- якщо вибраний ручний режим «І» була натиснута кнопка на шафі управління «АБО» в SCADA-системі;
- якщо ручний клапан 11НУ2 знаходиться в положенні 1 «І» вибраний автоматичний режим 3 «АБО» 4 для ємності Е-9;
- якщо ручний клапан 11НУ2 знаходиться в положенні 2 «І» вибраний автоматичний режим 3 «АБО» 4 для ємності Е-10.

Також функціональний блок має два входи «fb_Close», «fb_Open», що приймають сигнали про поточне положення пневматичного клапану.

До вихідних параметрів функціонального блоку відносяться:

- «ENO» - сигнал стану ФБ;

- «Open» - сигнал відкриття клапану;
- «Opened» і «Closed» - сигнал стану клапану, що в даному випадку використовуються для індикації на передній панелі шафи управління.

На рис.3.9 зображена внутрішня схема функціонального блоку «Valve_Control».

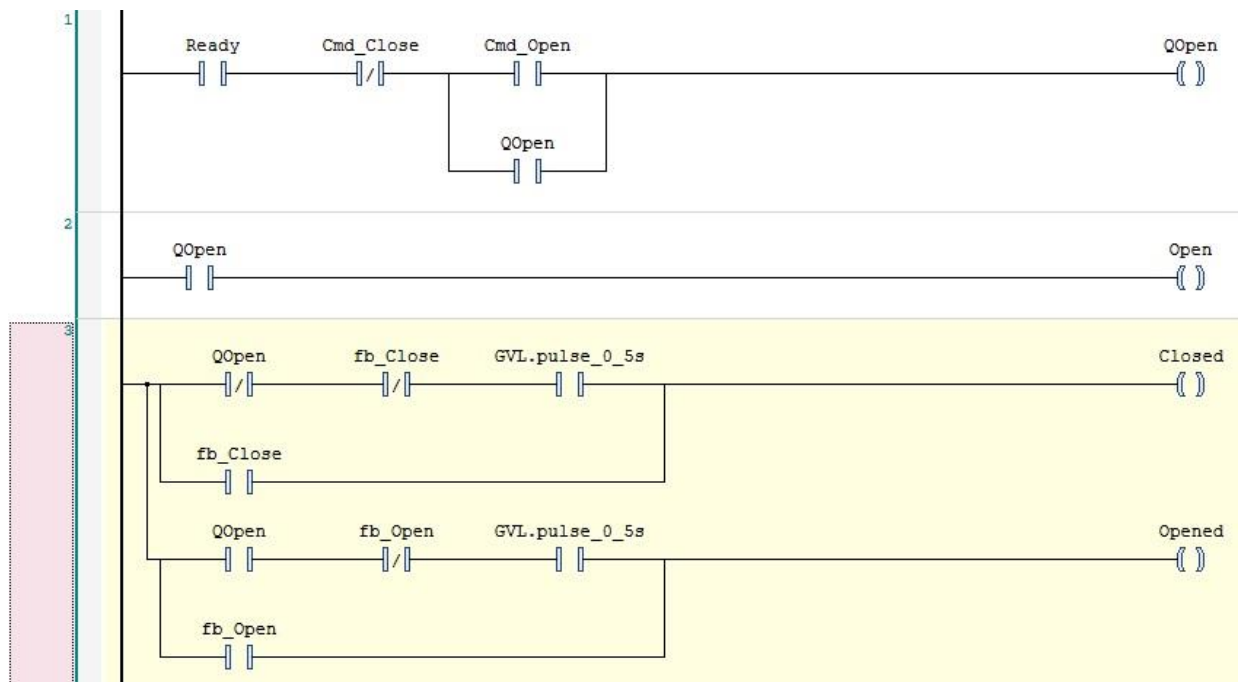


Рисунок 3.9 – Схема функціонального блоку «Valve_Control»

З рисунку, який наведений вище видно, що в першому рядку визначається умова відкриття та закриття клапану: якщо надійшов сигнал готовності - «Ready», «I» немає сигналу на закриття клапану – «Cmd_Close» «I» надійшов сигнал на відкриття клапану – «Cmd_Open», то видається сигнал на відкриття «QOpen», що «підхоплює» початковий сигнал на відкриття клапану і встановлює значення «True» вихідній змінній «Open». В іншому випадку умова не виконується і ланцюг розпадається. В третьому рядку здійснюється визначення стану клапану і видається відповідний сигнал для індикації:

- якщо не видається сигнал на відкриття «I» клапан не закритий, тоді видається сигнал з інтервалом в 0,5 секунди, «АБО», якщо клапан

закритий, то видається постійний сигнал, що сигналізує про закрите положення - «Closed»;

- якщо видається сигнал на відкриття «I» клапан не відкритий, тоді видається сигнал з інтервалом в 0,5 секунди, «АБО», якщо клапан відкритий, то видається постійний сигнал, що сигналізує про відкрите положення - «Opened»;

3.3.3 Керування насосами

Управління кожним з насосів здійснюється за однаковим алгоритмом: спочатку відбуваються визначення обраного режиму; після цього перевіряється стан насосу та елементів взаємодії (кнопок на шафі управління та кнопок SCADA-системи); згідно з обраним режимом визначається швидкість двигуна (рис.3.11); після запуску перевіряються умови роботи (рис.3.12). Отже розглянемо функціональну схему роботи насосу Н-1 (рис.3.10).

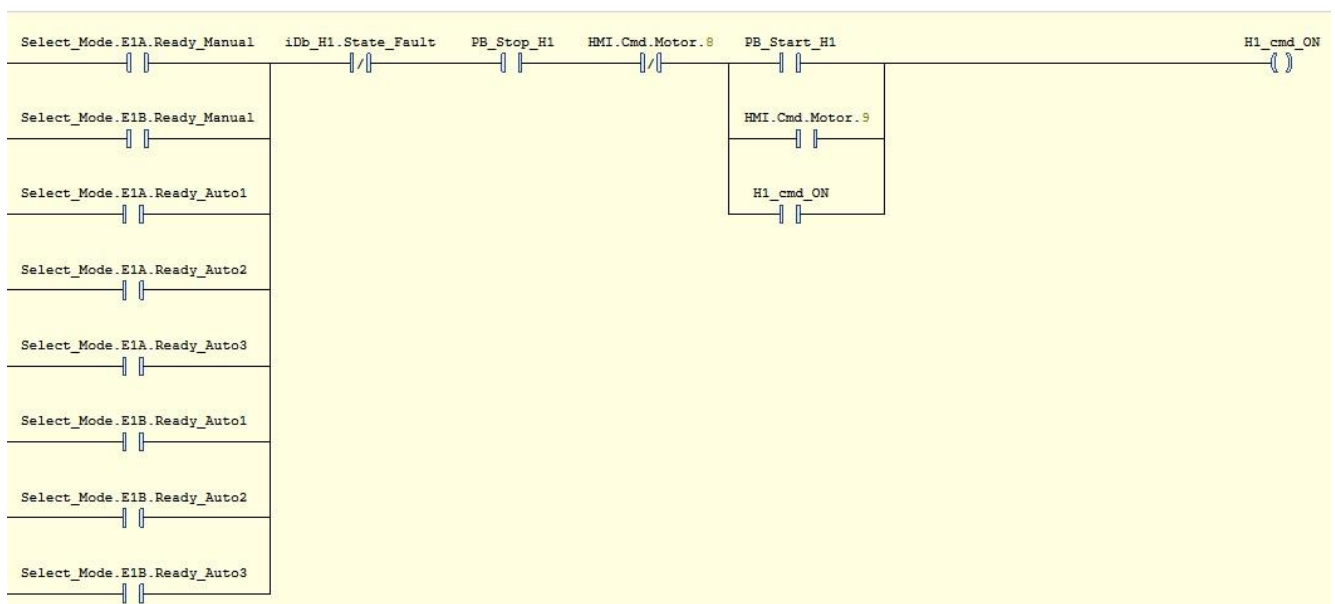


Рисунок 3.10 – Визначення обраного режиму

З рис.3.10 видно, що насос Н-1 має сім режимів роботи, які визначаються за умовою «АБО». Після визначення, перевіряється наступна умова: якщо відсутні аварії двигуна (iDb_H1.State.Fault), «I» кнопка зупинки насосу не натиснута

(PB_Stop_H1), «I» нульовий біт змінної «HMI.Cmd.Motor» дорівнює значенню False, «I» натиснута кнопка запуску насосу «АБО» дев'ятий біт змінної «HMI.Cmd.Motor» дорівнює значенню True, тоді видається команда на запуск насосу (H1_cmd_ON = True) (рис.3.11).



Рисунок 3.11 – Визначення швидкості роботи згідно обраного режиму

У випадку зображеному на рис.3.11 використовується функціональний блок «MOVE». Цей функціональний блок слугує для присвоєння значення змінної на другому вході, змінній, що знаходиться на другому виході. Згідно цього розглянемо, яким чином визначається швидкість в залежності від обраного режиму:

- якщо немає команди на запуск (H1_cmd_ON = False), то швидкість роботи двигуна дорівнює нулю;
- якщо надійшла команда на запуск двигуна «I» немає аварій (iDb_H1.State.Fault = False), то згідно обраного режиму буде визначено швидкість, яку користувач задав у відповідному вікні HMI.

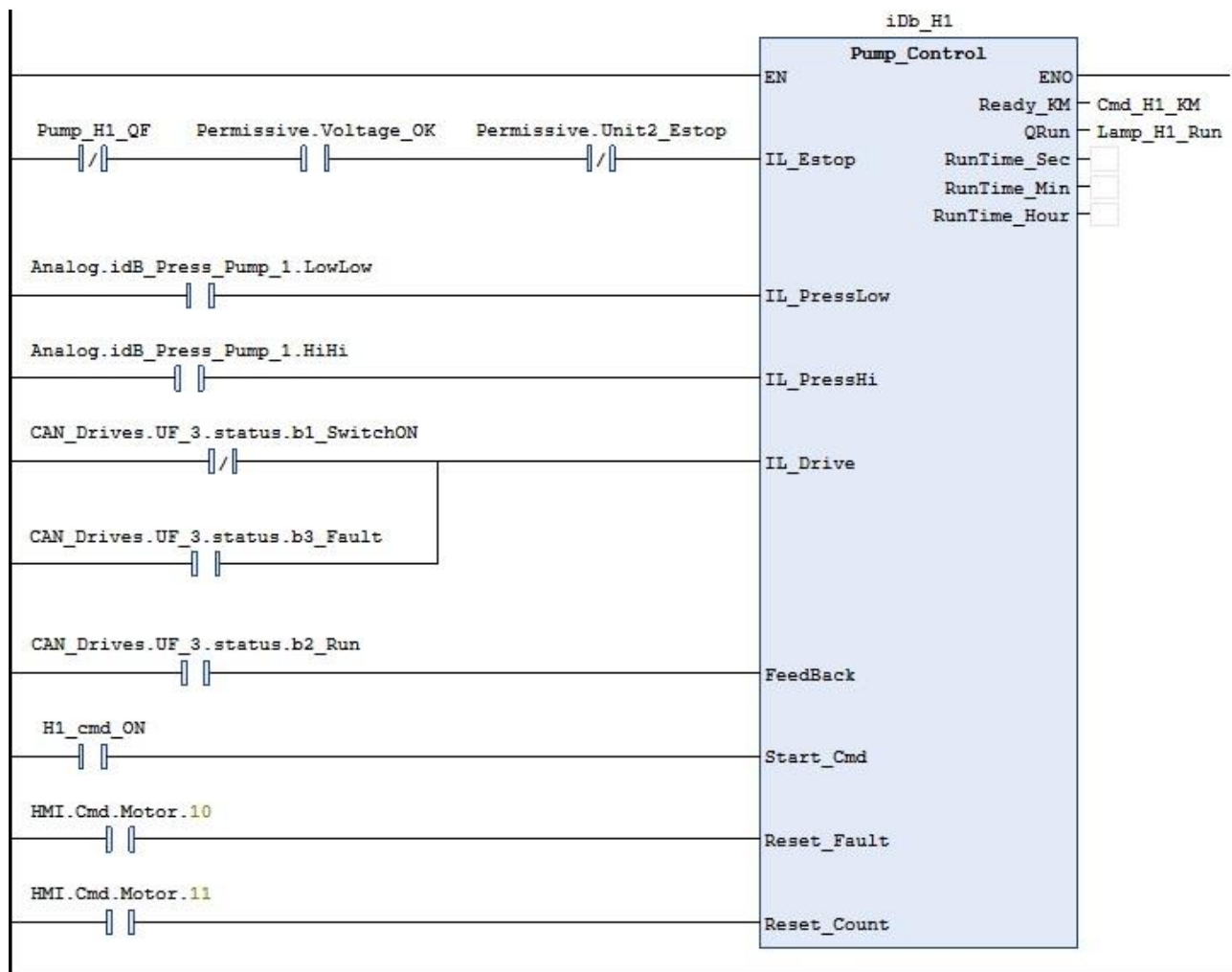


Рисунок 3.12 – ФБ керування насосом

Розглянемо вхідні та вихідні параметри функціонального блоку, що зображений на рис.3.12, а також вхідні та вихідні ланцюги змінних. До вхідних параметрів відносяться:

- «EN» вхід, що дозволяє роботу ФБ;
- «IL_Estop» вхід стану електричної схеми;
- «IL_PressLow» вхід, для сигналу низького тиску;
- «IL_PressHi» вхід, для сигналу високого тиску;
- «IL_Drive» вхід стану частотного перетворювача;
- «FeedBack» зворотній зв'язок від двигуна;
- «Start_Cmd» вхід, для сигналу на запуск двигуна;
- «Reset_Fault» вхід, для скидання аварійних повідомлень;
- «Reset_Count» вхід, для скидання часу двигуна.

Виходи функціонального блоку складаються з таких, параметрів:

- «ENO» стан ФБ;
- «Ready_KM» готовність електричної схеми;
- «QRun» стан двигуна;
- «Runtime_Sec», «Runtime_Min», «Runtime_Hour» час двигуна.

Розглянемо ланцюги змінних. До входу «IL_Estop» під'єднані три змінні з умовою: якщо автоматичний вимикач не увімкнений (Pump_H1_QF = False) «I» напруга мережі в порядку (Permissive.Voltage_OK) «I» не натиснута аварійна кнопка вузла (Permissive.Unit2_Estop), то електрична схема зібрана (IL_Estop = True). Дві змінні, що під'єднані з умовою «АБО», до входу «IL_Drive» визначають стан частотного перетворювача (включений «АБО» аварійний стан).

Визначимо змінні, що під'єднані без ланцюгів:

- аварійно низьке значення тиску (Analog.idB_Press_Pump_1.LowLow);
- аварійно високе значення тиску (Analog.idB_Press_Pump_1.HiHi);
- стан роботи частотного перетворювача (CAN_Drives.UF_3.status.b2_Run);
- сигнал запуску насосу Н-1 (H1_cmd_ON);
- стан десятого біту змінної HMI.Cmd.Motor;
- стан одинадцятого біту змінної HMI.Cmd.Motor.

3.3 Розробка людино-машинного інтерфейсу

Основним критерієм при виборі середовища розробки людино-машинного інтерфейсу є його сумісність з програмованим логічним контролером. В даному випадку була обрана SCADA «Galileo», що розроблена компанією Eaton, яка є виробником ПЛК. Інтуїтивно зрозумілий інтерфейс, що зображений на рис.3.13, доповнюється можливістю використання графіків, менеджера користувачів, менеджера аварійних повідомлень, базових графічних об'єктів (регулятори, стовпцеві регулятори, перемикачі).

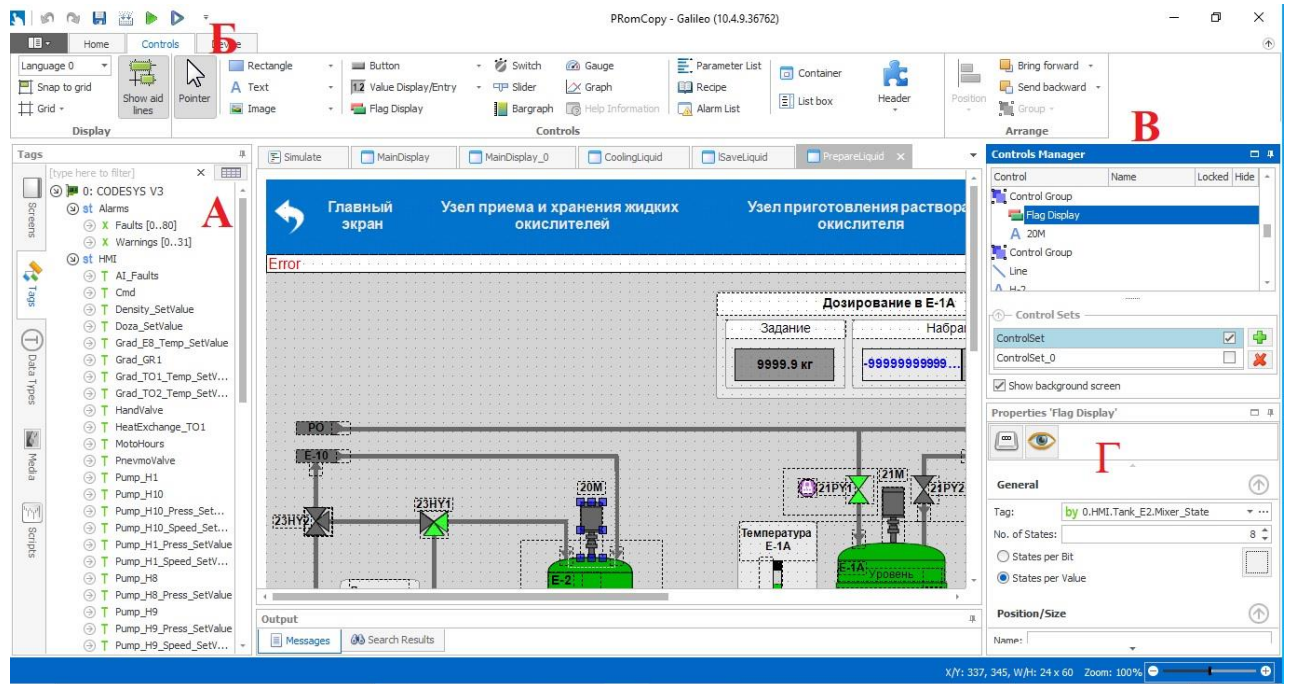


Рисунок 3.13 – Интерфейс среды разработки

З рис.3.13 видно, що інтерфейс складається з таких вікон: а – вікно керування гілками проекту ; б – вікно графічних об'єктів; в – вікно керування об'єктами робочої області; г – вікно властивостей об'єкту.

Розробники даної SCADA-системи надихалися такими середовищами розробки, як «Delphi» та «C# Windows Forms», тому «Galileo» дуже схоже на них з точки зору надання потенційному розробнику готових рішень по інтерфейсу. В той же час, суттєвими недоліками середовища розробки є відсутність зображень об'єктів промисловості та неможливість зміни алгоритму або додавання властивостей до базових графічних об'єктів, що змушує розробника в першому випадку самостійно створювати зображення цих об'єктів, а в другому випадку розробник не має гнучких елементів для створення майбутньої системи моніторингу і управління. Мінімальні вимоги до системи розробника наведено в табл.3.2.

Таблиця 3.2 – Системні вимоги Galileo [16]

Операційна система	Процесор	Оперативна пам'ять	Вільне місце на жорсткому диску
Windows 7 (SP1)	1 ГГц	1 Гб	300 Мб

Графічний інтерфейс розробленої SCADA-системи складається з шести екранів, які розподілені за ієрархією рівнів (рис.3.14). Зовнішній вигляд екранів представлений на (рис.А.1, А.2, А.3, А.4, А.5).

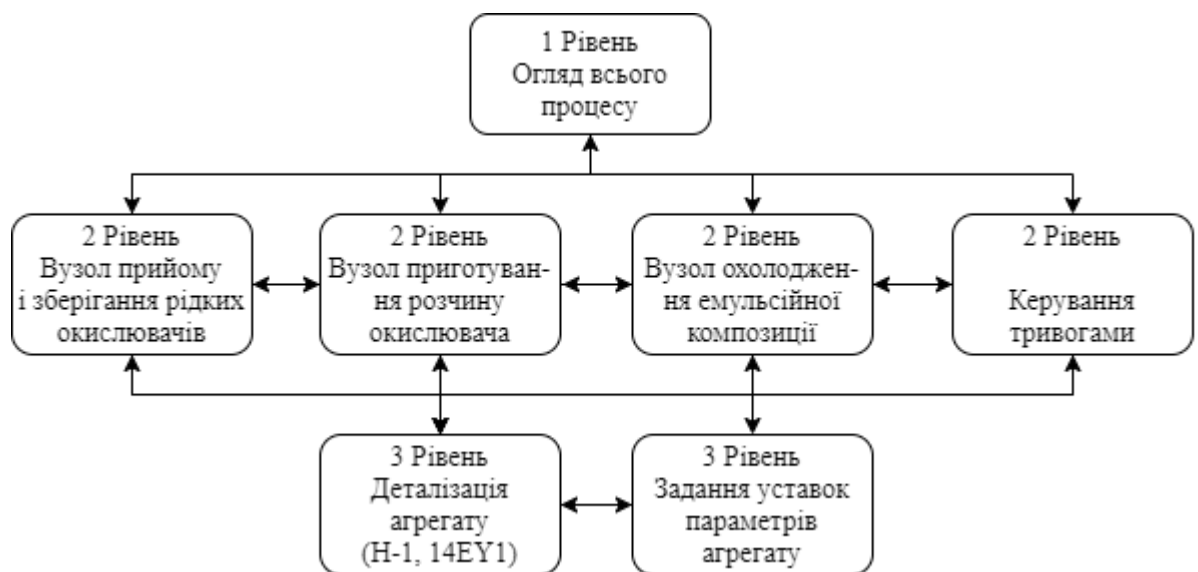


Рисунок 3.14 – Схема рівнів розроблюваного НМІ

Як вже було зазначено раніше, ієрархія і організація екранів повинна бути створена для забезпечення поступового розкриття інформації. Головний екран, що знаходиться на першому рівні тільки відображає інформацію про найважливіші параметри системи та дає змогу перейти до більш складних і деталізованих екранів на яких виконується керування процесом. Навігація по екранам здійснюється вільним чином, що дає змогу оператору перейти з будь-якого рівня на цільовий і тим самим пришвидшує його роботу. Зміст екранів людино-машинного інтерфейсу та задачі, які виконуються оператором, наведено в табл.3.3.

Таблиця 3.3 – Зміст екранів ЛМІ та вирішувані задачі

Рівень	Екран	Інформація	Зміст	Задачі
1	2	3	4	5
1	Головний екран (рис.А.1)	Основні параметри ТП	Температура, рівень, тиск, позиція клапанів обігріву, режими роботи	1. Оцінка і прогнозування динаміки ТП 2. Оцінка відповідності параметрів до регламенту
		Сигналізація основних параметрів та обладнання ТП	Сигнали оповіщення про стан основних параметрів та обладнання ТП	
		Графіки основних параметрів ТП	Температура в ємностях приготування і теплообміннику	
2	Вузол прийому і зберігання рідких окислювачів (рис.А.2)	Сигналізація	Сигнали оповіщення про стан параметрів та індикація стану обладнання ТП	1. Оцінка і прогнозування динаміки ТП 2. Завдання параметрів ТП 3. Оцінка готовності до режиму роботи 4. Керування запуском і зупинкою двигунів 5. Оцінка виконання режиму роботи 6. Оцінка виконання обігріву
		Параметри ємностей та витратоміру	Температура, рівень, маса, об'єм, густина	
		Позиції клапанів	Позиція ручних клапанів (циркуляція, на Е-1А,Б), позиція пневматичних клапанів(відкрито, закрито), позиція електропозиційних клапанів (відкрито і закрито у %)	
		Параметри насосів	Режим роботи з можливістю вибору, завдання швидкості роботи	

Продовження таблиці 3.3

1	2	3	4	5
2	Вузол приготування розчину окислювача (рис.А.3)	Сигналізація	Сигнали оповіщення про стан параметрів та індикація стану обладнання ТП	1. Оцінка і прогнозування динаміки ТП 2. Завдання параметрів ТП 3. Керування запуском і зупинкою двигунів 4. Оцінка готовності до режиму роботи 5. Оцінка виконання режиму роботи 6. Оцінка виконання дозування
		Параметри ємностей	Температура, рівень, маса, густина	
		Позиції клапанів	Позиція ручних клапанів (на Н-1, на Е-2, на Е-3, на Е-10), позиція пневматичних клапанів (відкрито, закрито)	
		Параметри насосу	Тиск, режим роботи з можливістю вибору, завдання швидкості роботи	
2	Вузол охолодження емульсійної композиції (рис.А.4)	Сигналізація	Сигнали оповіщення про стан параметрів та індикація стану обладнання ТП	1. Оцінка і прогнозування динаміки ТП 2. Завдання параметрів ТП 3. Керування запуском і зупинкою двигунів 4. Оцінка готовності до режиму роботи 5. Оцінка виконання режиму роботи
		Параметри ємності	Температура	
		Параметри градирні	Режим роботи	
		Позиції клапанів	Позиція ручних клапанів (на Н-8, на Т-1, на Т-2)	
		Параметри насосу	Режим роботи, тиск	
		Параметри теплообмінника	Температура	
2	Панель керування аварійними повідомленнями	Параметри повідомлення	Текст повідомлення з значенням параметру ТП, час виникнення події	1. Діагностика події, що виникла 2. Аналіз попередніх подій

Продовження таблиці 3.3

1	2	3	4	5
3	Завдання уставки параметрів ТП (рис.А.5)	Ємність Е-10	Уставка температури, уставка рівня, уставка коефіцієнтів регулятора обігріву	1.Завдання уставок параметрів ТП
		Ємність Е-9	Уставка температури, уставка рівня, уставка коефіцієнтів регулятора обігріву	
		Ємність Е-1А	Уставка температури, уставка рівня	
		Ємність Е-1Б	Уставка температури, уставка рівня	1.Завдання уставок параметрів ТП
		Ємність Е-2	Уставка рівня	
		Градирня	Уставки температури емульсії, охолоджуючої води, ємності Е-8	
		Насоси	Уставки тиску Н-1, Н-8	
3	Насос	Параметри	Швидкість, струм, момент, потужність, напруга, години, показники стану	1. Оцінка стану насосу 2. Діагностика несправностей
3	Електро-позиційний клапан	Параметри	Вибір режиму роботи, позиція, стан, завдання позиції, повний хід (в секундах), імпульс (в секундах)	1. Оцінка стану клапану 2. Керування клапаном

Продовження таблиці 3.3

1	2	3	4	5
3	Пневматичний клапан	Позиція клапану	Відкрито, закрито	1. Оцінка стану клапану 2. Керування клапаном

Для розробки графічних засобів, які розміщуються на вищезгаданих екранах використовувалися стандартні інструменти обраного середовища розробки. Кожен з графічних засобів складається з об'єктів візуалізації, які в свою чергу поділяються на статичні і динамічні. Статичні об'єкти візуалізації розташовуються на задньому плані екрану і не змінюються під час роботи. Динамічні об'єкти візуалізації, навпаки розташовані на передньому плані екрана, і їх зовнішній вигляд може змінюватися в результаті зміни даних у ПЛК.

Об'єкти статичної візуалізації, які також називаються "статичними елементами керування", складаються з простих елементів малювання, які містяться практично в кожному відповідному додатку: лінія, прямокутник, еліпс, багатокутник, текст і зображення.

Об'єкти динамічної візуалізації - це елементи керування візуалізацією, які використовуються для виведення чи введення на екрані пристрою. Вони мають подібні властивості і можуть бути класифіковані наступним чином залежно від їх функції:

- об'єкти, що використовуються для відображення стану (Flag display, Value display, Bargraph);
- об'єкти керування змінними ПЛК (Button, Switch, Value Entry, Slider);

- об'єкти запуску спеціальних функцій (Function Key, Screen Change, Help Window, Time/Date).

В свою чергу кожен об'єкт візуалізації має типові властивості:

- будь-який об'єкт можна скрити за певної умови (invisible if bit = True or False) ;
- будь-який динамічний об'єкт можна заблокувати з метою запобігання помилкового натиснення (inaccessible if bit = True or False);
- позиція та розмір об'єктів візуалізації редагується за координатами (вісь X / вісь Y) та шириною і висотою відповідно.

Виходячи з зображення (рис.3.15) розглянемо розробку засобу графіки, який найчастіше зустрічається на всіх екранах НМІ та є найскладнішим з точки зору кількості використаних об'єктів візуалізації.

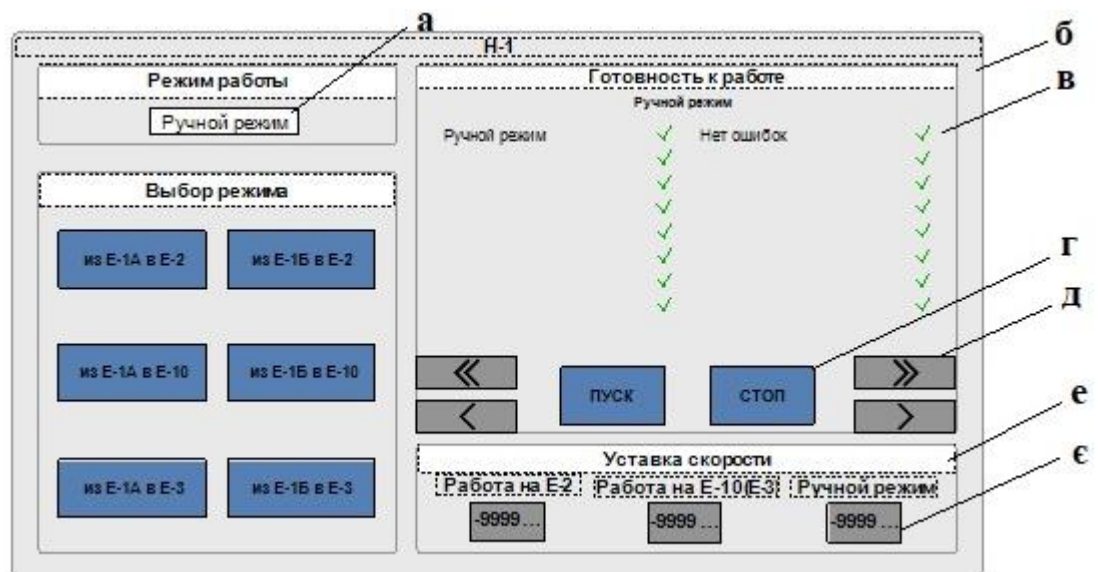


Рисунок 3.15 – Вікно управління насосом

В даному випадку на рис.3.15 зображено вікно керування насосом та його об'єктний склад, де: а – відображення у вигляді прапору (Flag display); б - геометрична фігура (Geometric shape); в – контейнер (Container); г – кнопка (Button); д – функціональна кнопка (Function key) ; е - текст (Text); є - ввід значень

(Value entry). Для виокремлення та послідовного подання інформації використовуються такі статичні об'єкти візуалізації, як текст та прямокутники.

Як видно з фрагменту (рис.3.15а) повідомлення про режим роботи насоса реалізоване за допомогою динамічного об'єкту «Flag display», що базується на правилах булевої алгебри. Властивості об'єкту зображені на рис.3.16.

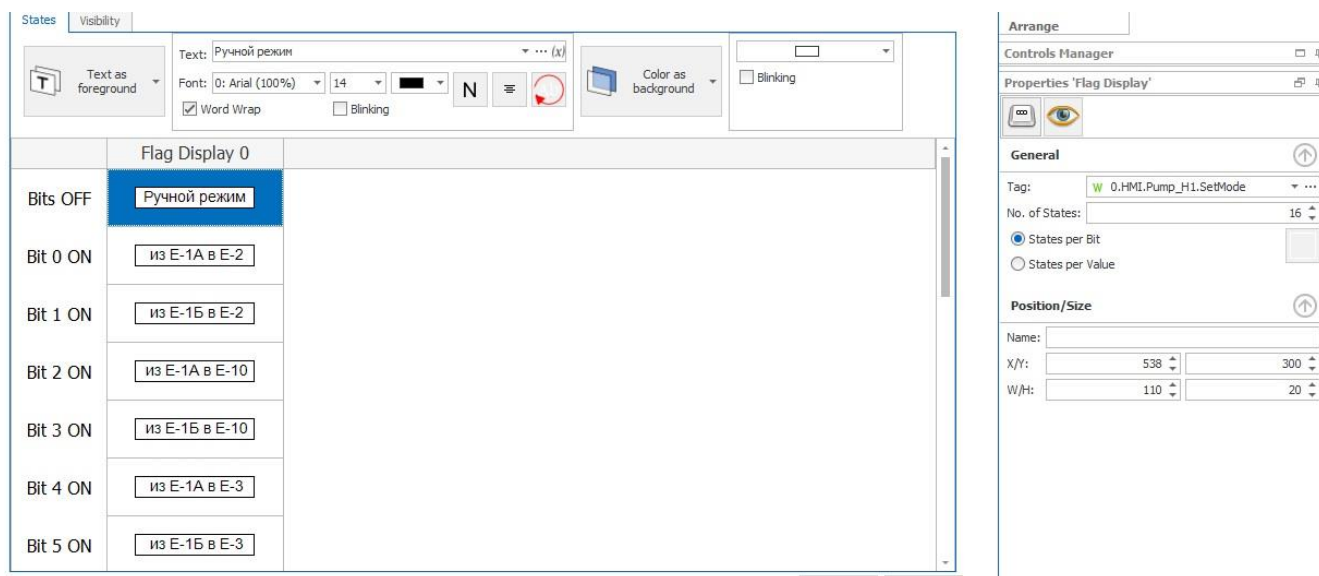


Рисунок 3.16 – Властивості об'єкту «Flag display»

Згідно рис.3.16 для роботи з даними у ПЛК, потрібно вказати відповідну змінну в полі «Tag» та спосіб визначення станів (побітово або за величиною змінної), а також їх кількість. В даному випадку обрана змінна «0.HMI.Pump_H1.SetMode» з типом даних «WORD», яка містить в кожному біті інформацію про поточний режим роботи. Насос Н-1 має сім режимів роботи, отже потрібно визначити стан, як мінімум семи бітів та одного, що передбачає аварійний сигнал (інші біти відповідатимуть за правильність сигналу від ПЛК). Налаштування, що знаходяться зліва на рис.3.15, є розширеними та слугують для редагування гарнітури тексту, а також кольору визначених станів.

На фрагменті рис.3.15в зображений динамічний об'єкт візуалізації – контейнер, що має наступні властивості (рис.3.17).

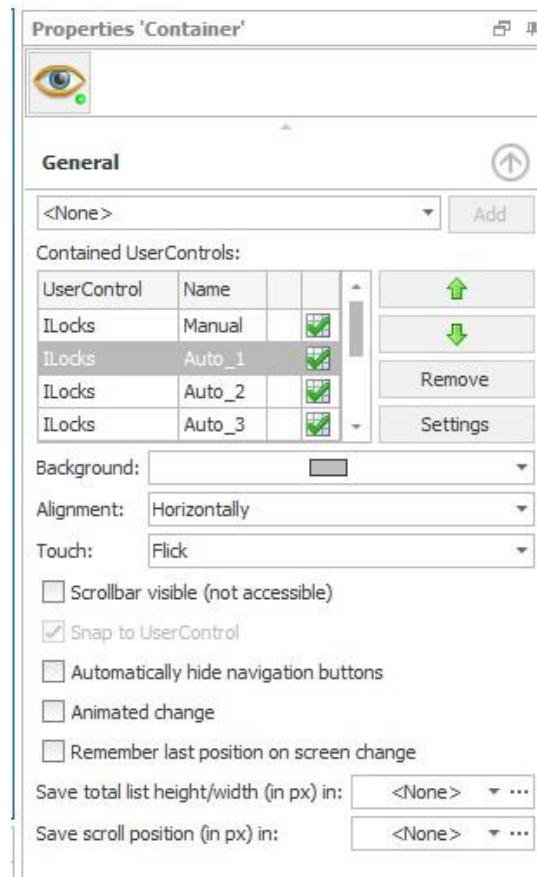


Рисунок 3.17 – Властивості контейнеру

З рис.3.17 видно, що контейнер містить в собі об'єкти «User controls» та функції для керування ними. Прокрутка об'єктів в контейнері (Alignment) відбувається горизонтально за допомогою жестів (Touch – flick) або функціональних кнопок. Користувацькі екрани (User controls) будуються на основі одного шаблону з параметрами, що зображений на рис.3.18. В даному випадку користувацькі екрани мають параметри з відомостями про готовність до режиму роботи та заповнені відповідною інформацією за допомогою вікна «Settings» (рис.3.19), де кожному параметру ставиться у відповідність потрібна змінна. Наприклад, параметру «Name» відповідає текст «Ручний режим», а параметру «NOп_1», що відповідає за відображення галки, відповідає бітова змінна «0.HMI.Pump_H1.ReadyManual», яка сигналізує про готовність до ручного режиму роботи, наступні користувацькі екрани будуються за таким же принципом.

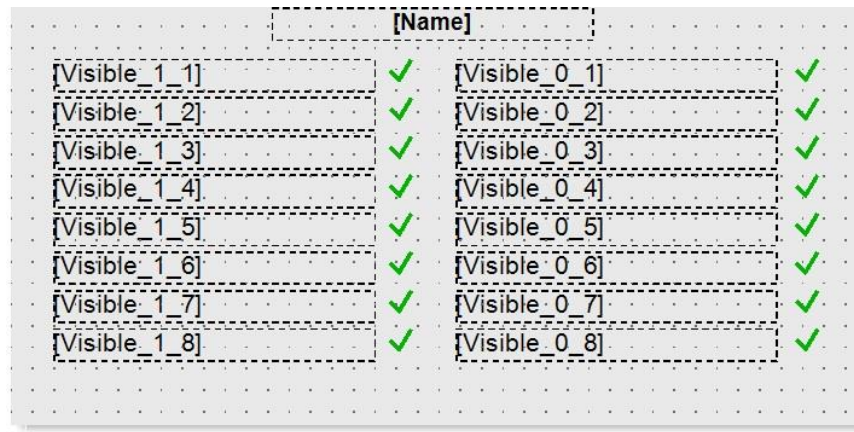


Рисунок 3.18 – Користувачський екран з параметрами

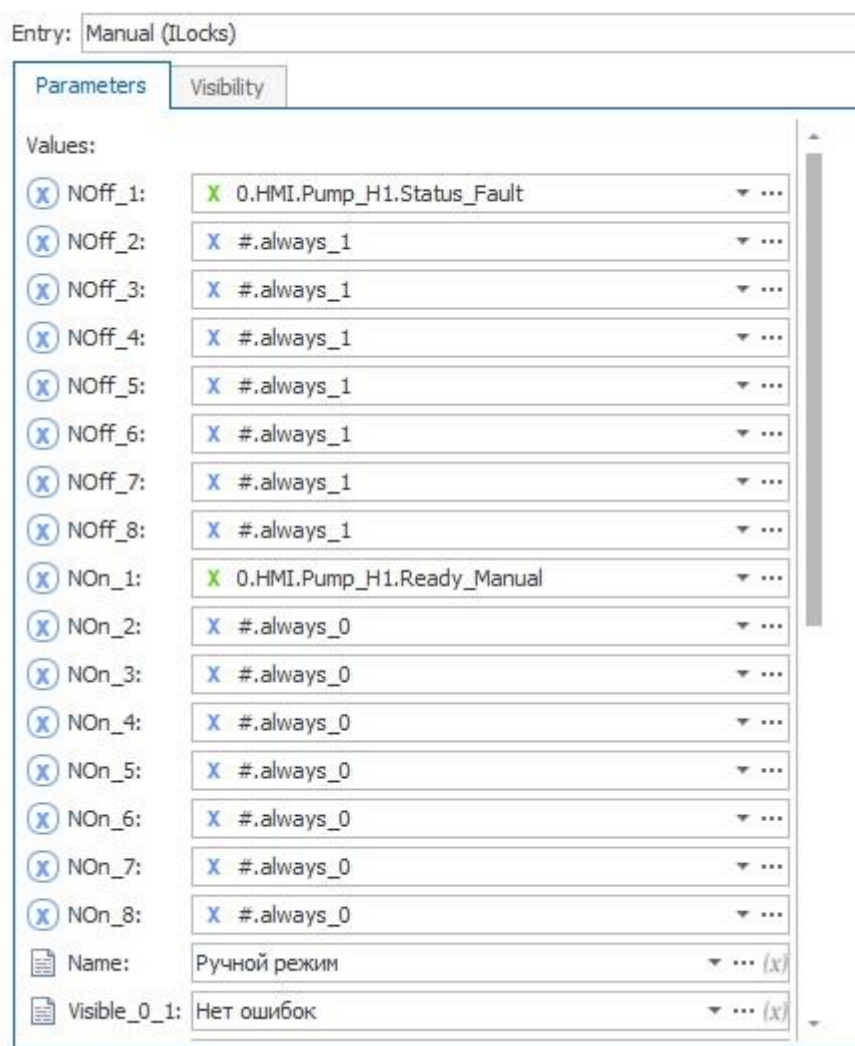


Рисунок 3.19 – Вікно заповнення параметрів відповідними даними

Вибір режимів та керування запуском і зупинкою насосу відбувається за допомогою об'єктів «Button», властивості одного з них зображено на рис.3.20.

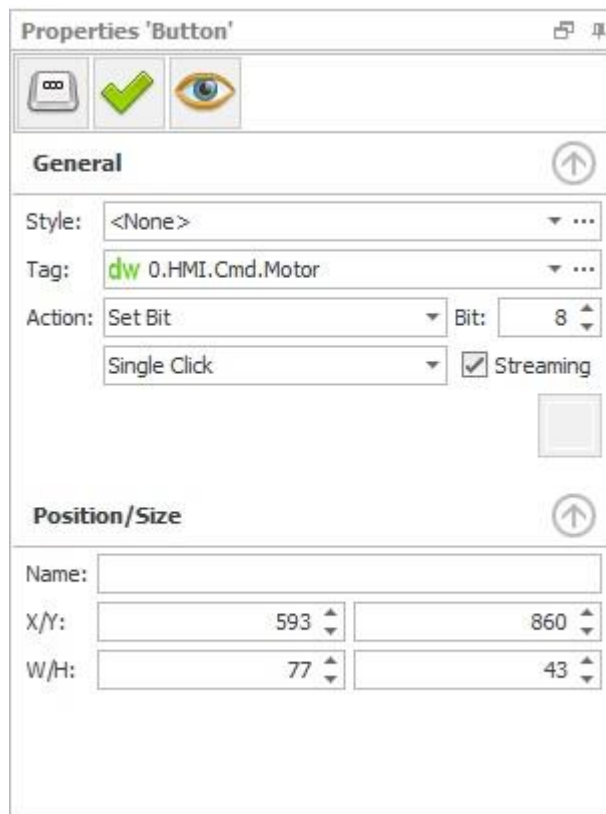


Рисунок 3.20 – Властивості об'єкту «Button»

Взаємодія з об'єктами «Button» має однотипний характер та може розрізнятися лише за такими властивостями:

- за способом дії (встановити / скинути біт);
- за номером біту (від 0 до 32) ;
- за натисканням (однократне / подвійне);
- за часом дії (дія відбувається поки натиснута кнопка, якщо Streaming=True, або дія відбувається до наступного натискання на кнопку).

В зображеному на рис.3.20 випадку, однократне натискання кнопки зупиняє насос Н-1, за допомогою встановлення восьмого біту в змінній «0.HMI.Cmd.Motor», що відповідає за управління усіма двигунами поточного технологічного процесу.

Функціональні кнопки (рис.3.15д) мають схожі властивості з об'єктами «Button», за винятком того, що вони застосовуються до іншого об'єкту візуалізації, а не до змінної ПЛК. В даному випадку функціональні кнопки застосовуються до

контейнеру, щоб здійснювати переміщення по користувацьким екранам, для цього використовуються такі функції:

- останній користувацький екран (Last entry);
- перший користувацький екран (First entry);
- наступний користувацький екран (Move next entry);
- попередній користувацький екран (Move previous entry).

Для вводу швидкості роботи насосу використовується об'єкт вводу значень (Value entry), властивості якого зображені на рис.3.21.

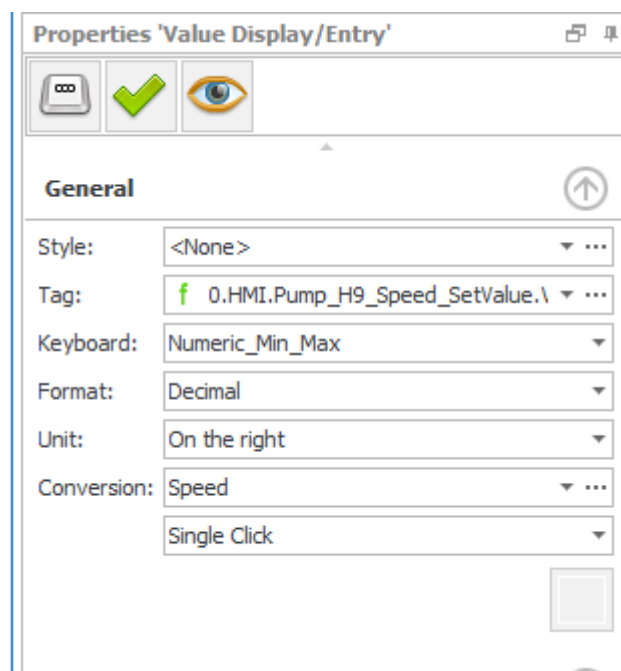


Рисунок 3.21 – Властивості об'єкту вводу

З рис.3.21 видно, що для вводу значень треба вказати змінну ПЛК в полі «Tag» та обрати екранну клавіатуру «Keyboard», а також формат даних та місце розташування одиниць вимірювання (за наявності).

4 КЕРІВНИЦТВО ОПЕРАТОРА

4.1 Призначення програми

Програма призначена для моніторингу і управління лінією виробництва емульсійного розчину.

4.2 Умови виконання програми

Цільова система повинна відповідати вимогам наведеним в табл.3.4. Програма встановлена в автозавантаження, тому запускається при завантаженні операційної системи.

Таблиця 3.4 – Системні вимоги до цільової системи

Операційна система	Процесор	Оперативна пам'ять	Вільне місце на жорсткому диску
Windows 7 (SP1)	800 МГц	512 Мб	20 Мб

4.3 Виконання програми

На рис.А.1 зображено головне вікно SCADA-системи. Інформація представлена у вигляді аналогових діапазонів, графіків і сповіщувального тексту, а також поділена за вузлами технологічного процесу. В даному випадку представлена ситуація, коли у вузлі охолодження процес проходить нормально, а в двох інших вузлах аварійні і попереджувальні повідомлення:

- аварія насосу Н-10;
- аварійний сигнал рівня в ємності Е-10;
- аварійний сигнал температури в ємності Е-9;
- сигнал попередження про тиск насосу Н-1;
- сигнал попередження про рівень в ємності Е-1Б.

Згідно цих сигналів оператор має прийняти рішення про зупинку або продовження технологічного процесу в двох вузлах. Щоб перейти до керування вузлами, достатньо натиснути на область інформації вузла або перейти до нього застосовуючи панель навігації (зверху). Для більш детальної інформації про сигнали, оператор може звернутися до панелі аварійних повідомлень, що знаходиться під панеллю навігації (рис.4.1) або з екрану вузла натисканням на обладнання відкрити вікно детальної інформації, в даному випадку інформації про насос (рис.4.2).

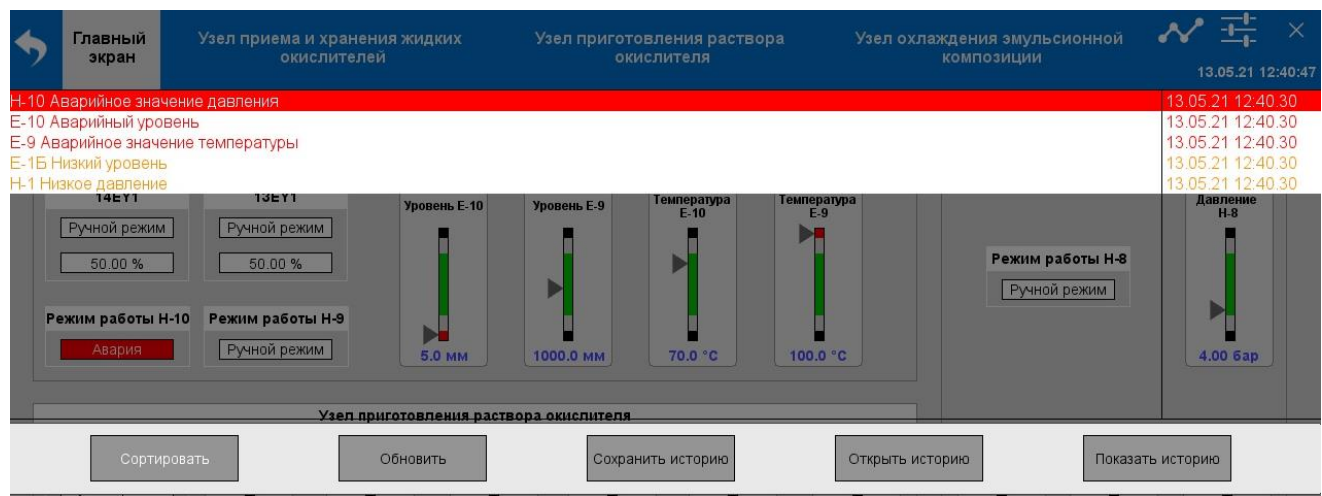


Рисунок 4.1 – Панель аварійних повідомлень



Рисунок 4.2 – Інформаційне вікно насосу Н-10

Розглянемо вікно вузла прийому і зберігання окислювачів (рис.А.2). По положенням клапанів 11НУ2 і 11РУ1 видно, що йшов процес перекачування розчину в ємності Е-1А і Е-1Б, але коли рівень розчину в ємності Е-10 став аварійним, виникла аварія Н-10 через низький тиск в насосі, отже оператору слід зупинити роботу насосу Н-10 за допомогою області управління зображеної на рис.4.3.

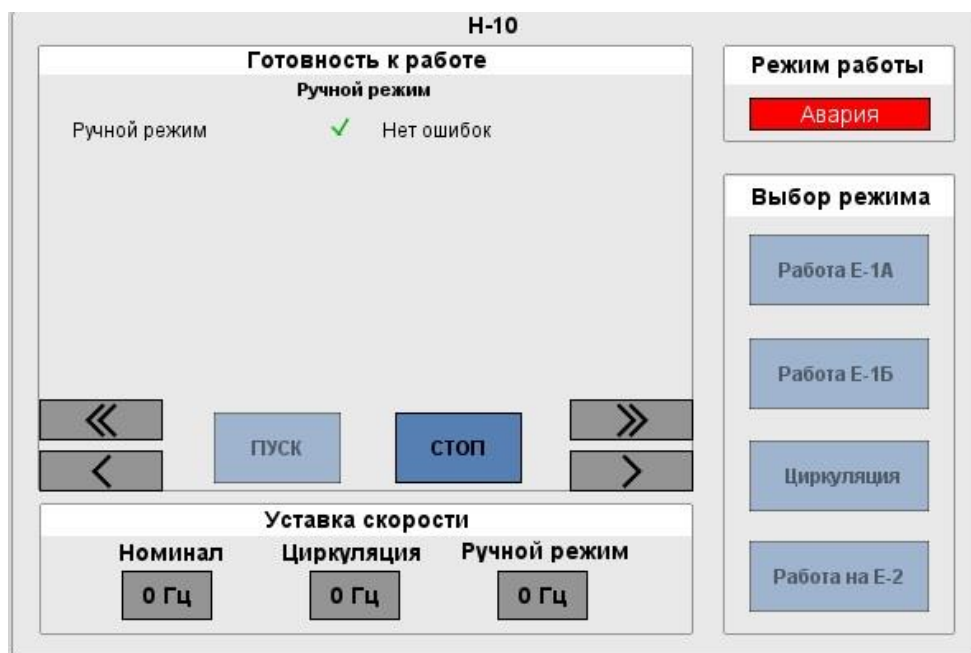


Рисунок 4.3 - Область керування Н-10

Область керування поділена на 4 частини : готовність до роботи – оцінка готовності до режиму, пуск/зупинка двигуна; режим роботи; вибір режиму роботи; уставка швидкості – всіх режимів(номінал), циркуляції, ручного режиму.

З рис.А.2 можна побачити, що температура в ємності Е-10 може вийти за рамки робочої, як це сталося в ємності Е-9. Оператор може змінити позицію електропозиційного клапану 14ЕУ1 в менший бік, та закрити 13ЕУ1 звернувшись до відповідних вікон (рис.4.4а,б).

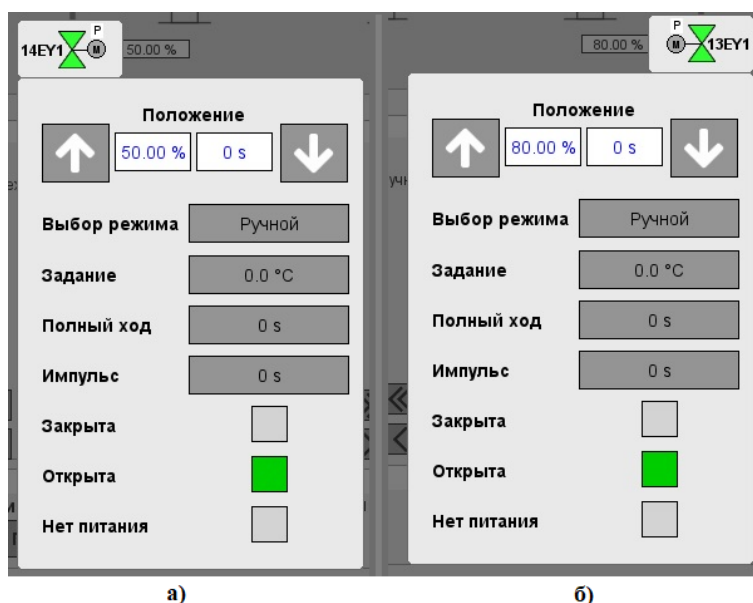


Рисунок 4.4 – Вікна електропозиційних клапанів: а) 14EY1; б) 13EY1.

На рис.4.4а,б зображені вікна для керування електропозиційними клапанами, де: стрілка вниз – кнопка видачі сигналу на закриття; стрілка вгору – кнопка видачі сигналу на відкриття; вибір режиму – ручний/автоматичний; завдання – завдання температури для автоматичного режиму; повний хід – час повного проходження від кінця до кінця; імпульс – час видачі сигналу після натискання кнопки.

Вигляд вікна вузла приготування розчину окислювача зображено на рис.А.3. Ситуація ідентична ситуації у вузлі прийому і зберігання, але має попереджуючий характер. Оператор має прийняти рішення про зупинку або продовження роботи двигуна. Якщо буде прийнято рішення про завершення процесу перекачування з Е-1Б в Е-2, то потрібно закрити клапани 22НУ1, 22РУ3. Керування пневматичними клапанами відбувається за допомогою вікна, приклад якого зображено на рис.4.5.

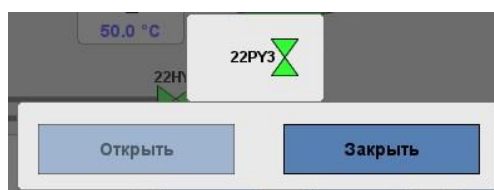


Рисунок 4.5 – Вікно керування клапаном 22РУ3

Екрани другого рівня схожі за наповненням інформацією, і вікно вузла охолодження емульсійної композиції (рис.А.4) не є винятком. На всіх екранах присутні, такі графічні елементи, як:

- технологічні лінії з клапанами;
- сигнали сповіщення;
- параметри технологічного процесу;
- область керування насосом;
- ввід параметрів технологічного процесу;

Більшість з переліку існуючих графічних елементів та принципів подання інформації не були розглянуті у другому розділі та створені на користь вимог потенційного користувача, так як згідно [3], для більшої частини проєктів ключовим моментом є визначення вимог користувачів, а також функціональних і інших вимог до продукту або системи. У даному випадку людино-орієнтовного проєктування було сформовано та скориговано вимоги причасних сторін, до елементів та принципів побудови НМІ. До результатів виконання вимог, можна віднести:

- стани обладнання закодовані кольорами згідно [21];
- екрани другого рівня складаються з мнемосхеми відповідних вузлів;
- значення рівню розчину зображується заповненням ємності;
- керування насосами здійснюється з другого рівня ієрархії.

ВИСНОВКИ

З метою покращення ситуаційної обізнаності оператора АСУ ТП, в роботі було досліджено та вирішено проблему сучасної людино-машинної взаємодії в промисловості.

Під час розгляду структури та складу сучасних автоматизованих систем управління була виділена трирівнева структура, що на відміну від інших більш гнучка до модернізації і відказу керуючого обладнання. Гнучкість забезпечується, за рахунок використання в своєму складі: панелей оператора, які можуть замінити верхній рівень; ПЛК, що замінюють релейні схеми .

В результаті огляду існуючих людино-машинних інтерфейсів, було визначено їх основні недоліки:

- відображення даних у вигляді чисел;
- низька кількість графіків або вони відсутні;
- анімація технологічного забезпечення промислових об'єктів;
- яскраве кольорове тривимірне обладнання;
- деталізовані зображення агрегатів;
- не послідовне використання кольорів для індикації;
- обмежена навігація по екранам;
- відсутність ієрархії відображення;

Також були обрані шляхи покращення людино-машинних інтерфейсів:

- використання кольору спеціально і послідовно;
- сірий фон для мінімізації проблем з відблисками та відображенням;
- відсутність анімацій, за винятком аварійних сигналів;
- графіки важливих параметрів;
- аналогове представлення даних, з визначеними межами нормальних, попереджувальних і аварійних умов;
- прості та зрозумілі двовимірні зображення;
- логічні та послідовні методи навігації;

- мінімізація помилок введення даних;
- заходи перевірки та безпеки;

На основі вищерозглянутої теоретичної інформації за допомогою методу людино-орієнтовного проектування, була спроектована та реалізована трирівнева АСУ ТП вибухонебезпечного хімічного виробництва, з SCADA-системою на верхньому рівні. Для розробки системи збору даних і управління, було використане середовище розробки Galileo, що сумісне з обраним ПЛК. Розроблений людино-машинний інтерфейс SCADA-системи відповідає розглянутим принципам :

- всі екрани побудовані на основі вимог потенційного користувача;
- інформація представлена для поняття ситуації;
- інформація представлена для прогнозування ситуацій;
- представлена інформація про загальний стан технологічного процесу;
- між формуванням і поданням даних дотримується компроміс;
- представлені критичні показники сигналів;
- основні параметри відображаються за допомогою трендів;
- інформація представлена за пріоритетом.

Результати роботи рекомендується використовувати при модернізації існуючих та розробці нових людино-машинних інтерфейсів в промисловості.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Hollifield B. The high performance HMI handbook / B. Hollifield, D. Oliver, I. Nimmo, E. Habibi.– USA, Kalamazoo: Digital Books, 2008. – 206 p.
2. Интерфейс человеко-машинный. Принципы приведения в действие : МЭК 60447-93.- [Действующий от 2000-12-08].- М.: ГОСТ Р МЭК 60447, 2000. – 349 с.
3. Эргономика взаимодействия человек—система. Часть 210. Человеко-ориентированное проектирование интерактивных систем : ISO 9241-210. – [Действующий от 2016-11-02]. – М.: Стандратинформ, 2016. – 35 с.
4. Втюрин В. А. Основы АСУ ТП [Текст] / В. А. Втюрин. – СПб.: Санкт-Петербургская Государственная Лесотехническая Академия имени С.М. Кирова, 2006. - 152 с.
5. Ивлев, П. Панель оператора как средство создания высокоэффективного HMI / П. Ивлев // Control engineering Россия. – 2018. – № 2. – С.1–5.
6. Автоматизация производств Oil Systems [Электрон. ресурс]. - Режим доступа: <https://oil-systems.ru/arm-scada/>.
7. Митин Г. П., Хазанова О. В. Системы автоматизации с использованием программируемых логических контроллеров: Учебное пособие / Г. П. Митин , О. В. Хазанова - М.: ИЦ МГТУ «Станкин», 2005. – 136 с.
8. Programmable controllers — Part 3: Programming languages : IEC 61131-3. – [Effective from 2013-02-20]. - INTERNATIONAL ELECTROTECHNICAL COMMISSION, 2013. – 460 p.
9. Системы автоматического управления на основе программируемых логических контроллеров // Schneider Electric. – 2006. – № 16. – С.21–22.
10. Деменков Н. П., К. А. Пупкова, Языки программирования промышленных контроллеров: Учебное пособие / Н. П. Деменков, К. А. Пупкова - М.: МГТУ им. Н.Э. Баумана, 2004. – 172 с.

11. Языки программирования промышленных контроллеров [Электрон. ресурс]. - Режим доступа: <http://xn-----8kcbtc0a2addoucyh7cl9e.xn--p1ai/assets/files/presentation/jci/sfc.pdf>.
12. Пользовательский интерфейс, SCADA-пакеты [Электрон. ресурс]. - Режим доступа: https://www.bookasutp.ru/chapter9_4.aspx.
13. Citect SCADA [Электрон. ресурс]. - Режим доступа: <http://scada.ru/software/citectscada/>.
14. Master SCADA [Электрон. ресурс]. - Режим доступа: <https://insat.ru/products/?category=9>.
15. Trace Mode [Электрон. ресурс]. - Режим доступа: https://ru.wikipedia.org/wiki/Trace_mode.
16. Galileo [Электрон. ресурс]. - Режим доступа: <https://www.eaton.com/content/dam/eaton/products/industrialcontrols-drives-automation-sensors/en-globalprime/product-overview-for-machinery.pdf>.
17. ASM Consortium's Solution Framework [Electronic resource]. – Access mode: <https://www.asmconsortium.net/Documents/ASM%20Consortium%20Solution%20Framework%20Creates%20Business%20Value%20for%20Adopters.pdf>.
18. Комната управления [Электрон. ресурс]. - Режим доступа: <https://back-in-ussr.com/2019/05/strogost-i-krasota-komnat-upravleniya-sovetskoy-epohi.html>.
19. ЧМИ 80-х [Электрон. ресурс]. - Режим доступа: <http://amatrosov.blogspot.com/2011/03/scada.html>.
20. Плоский дизайн [Электрон. ресурс]. - Режим доступа: <https://www.dreamstime.com/infographic-dashboard-template-flat-design-graphs-chart-infographic-dashboard-template-flat-design-graphs-charts-image129188591>.
21. Интерфейс человеко-машинный. Маркировка и обозначения органов управления и контрольных устройств. Правила кодирования информации: МЭК 60037-96.- [Действующий от 2000-12-08]. – 2000. – 348 с.
22. Endsley, M. R., Garland, D. J. Situation Awareness Analysis and Measurement / M. R. Endsly, D. J. Garland. – Lawrence Erlbaum Associates Publishers, 2000. – 418 p.

23. Сердюк С. Ергономічні питання проектування людино-машинних систем [Текст] / С. Сердюк.— З.: ЗНТУ, 2014. – 334 с.

24. Системні вимоги CoDeSys [Електрон. ресурс]. - Режим доступу: https://store.codesys.com/engineering/codesys.html?__store=en.

ДОДАТОК А

Зображення людино-машинного інтерфейсу

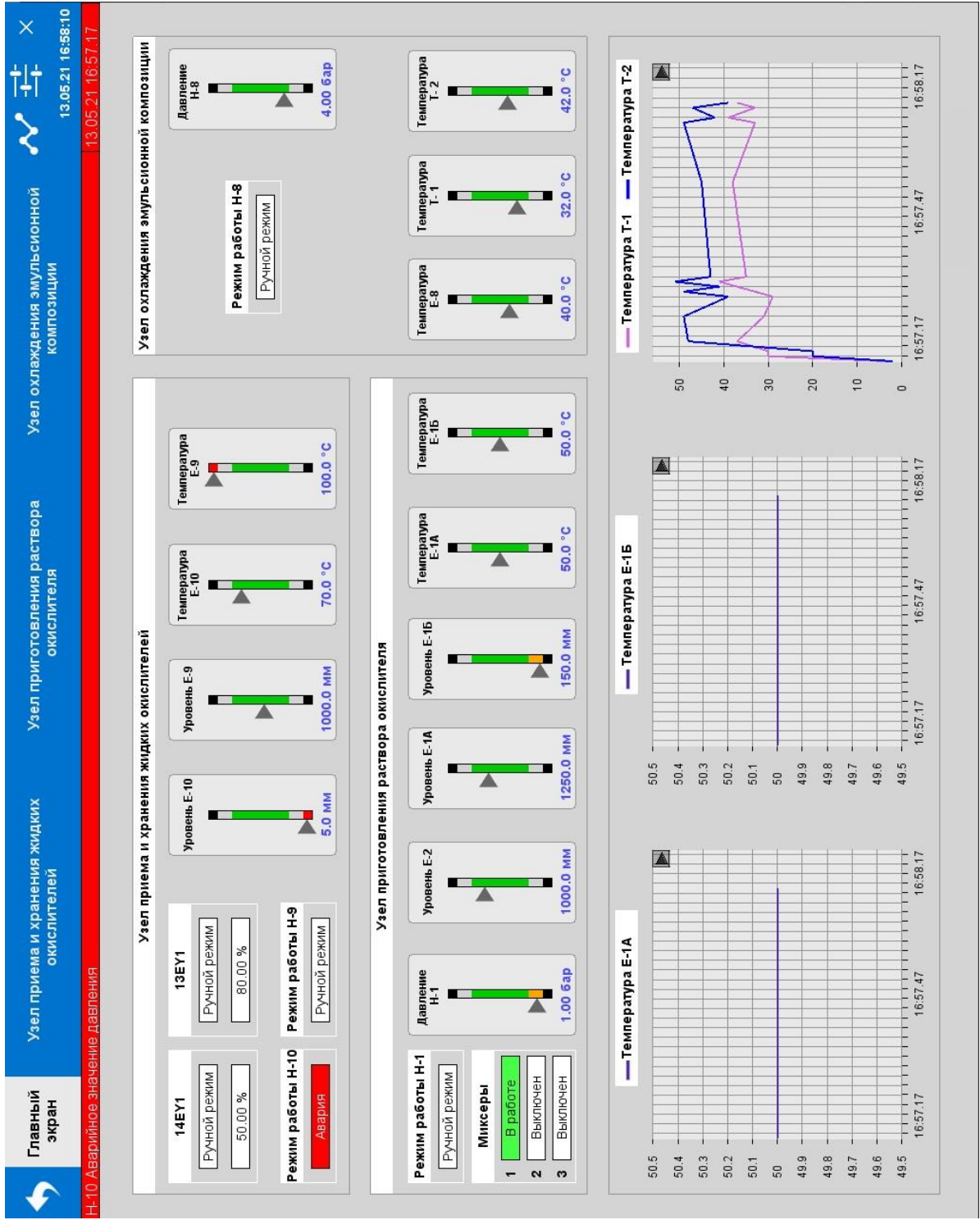


Рисунок А.1 – Зображення вікна головного екрану

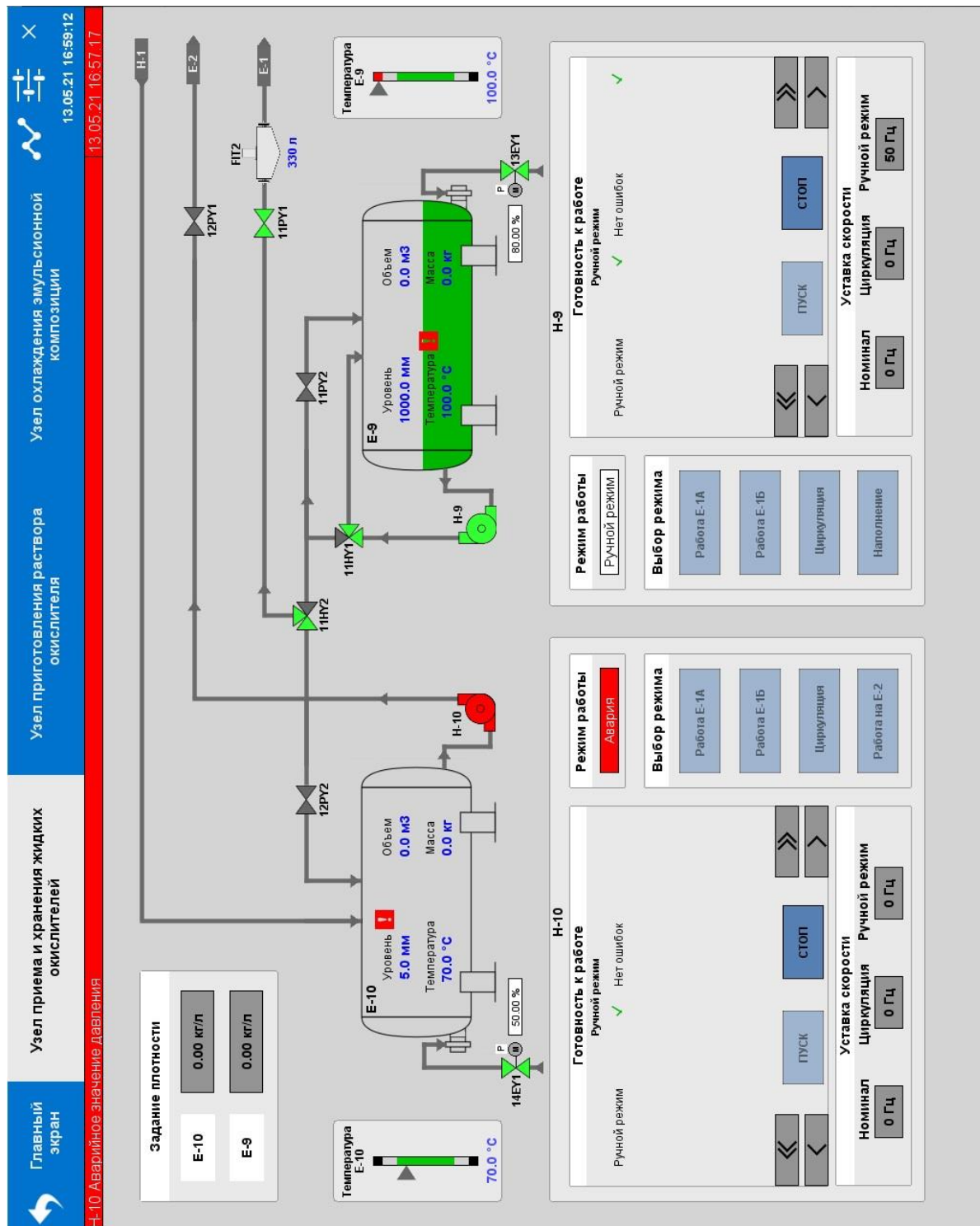
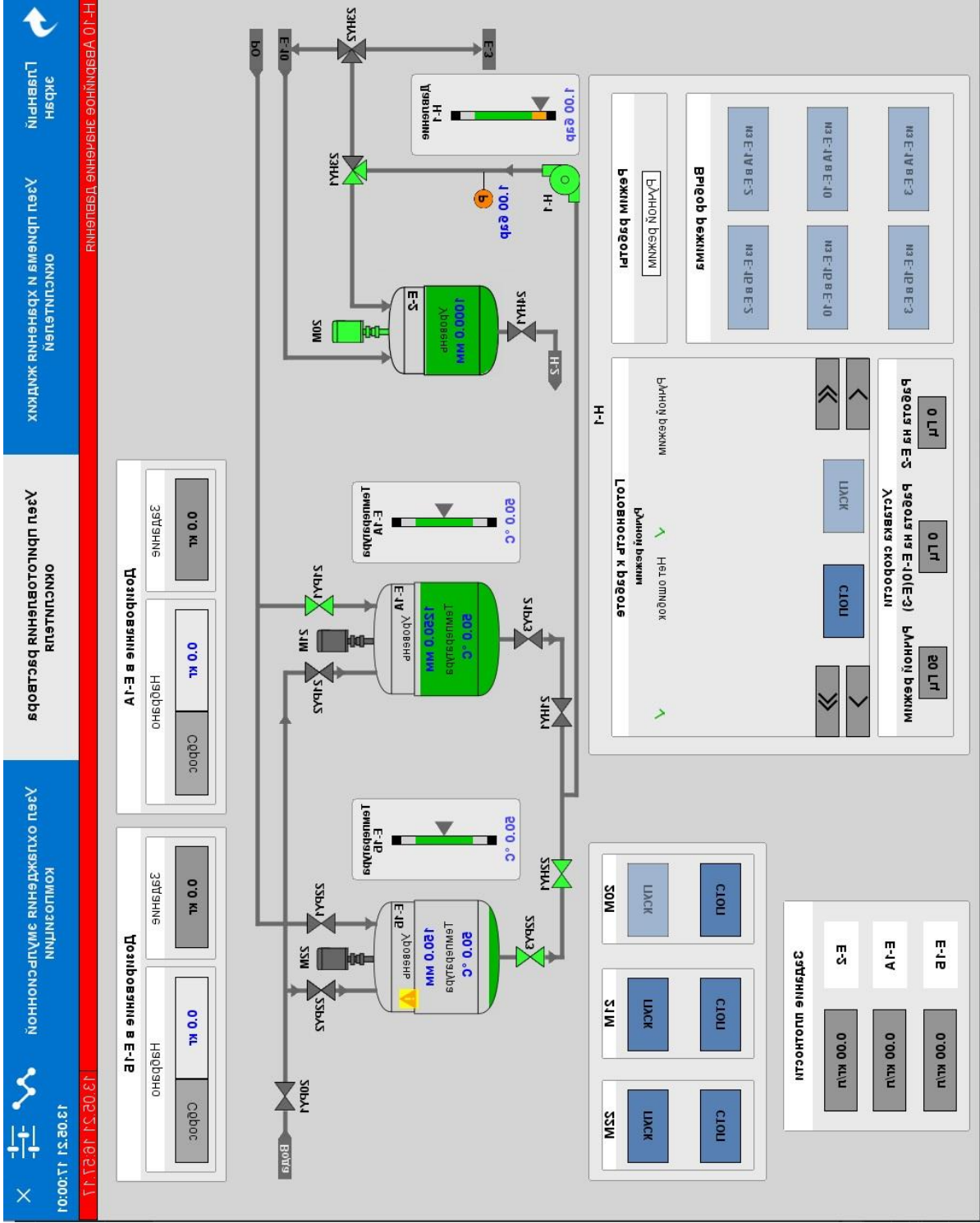


Рисунок А.2 – Зображення вікна вузла прийому та зберігання рідких окислювачів

Рисунок А.3 – Зображення вікна вузла приготування розчину окислювача



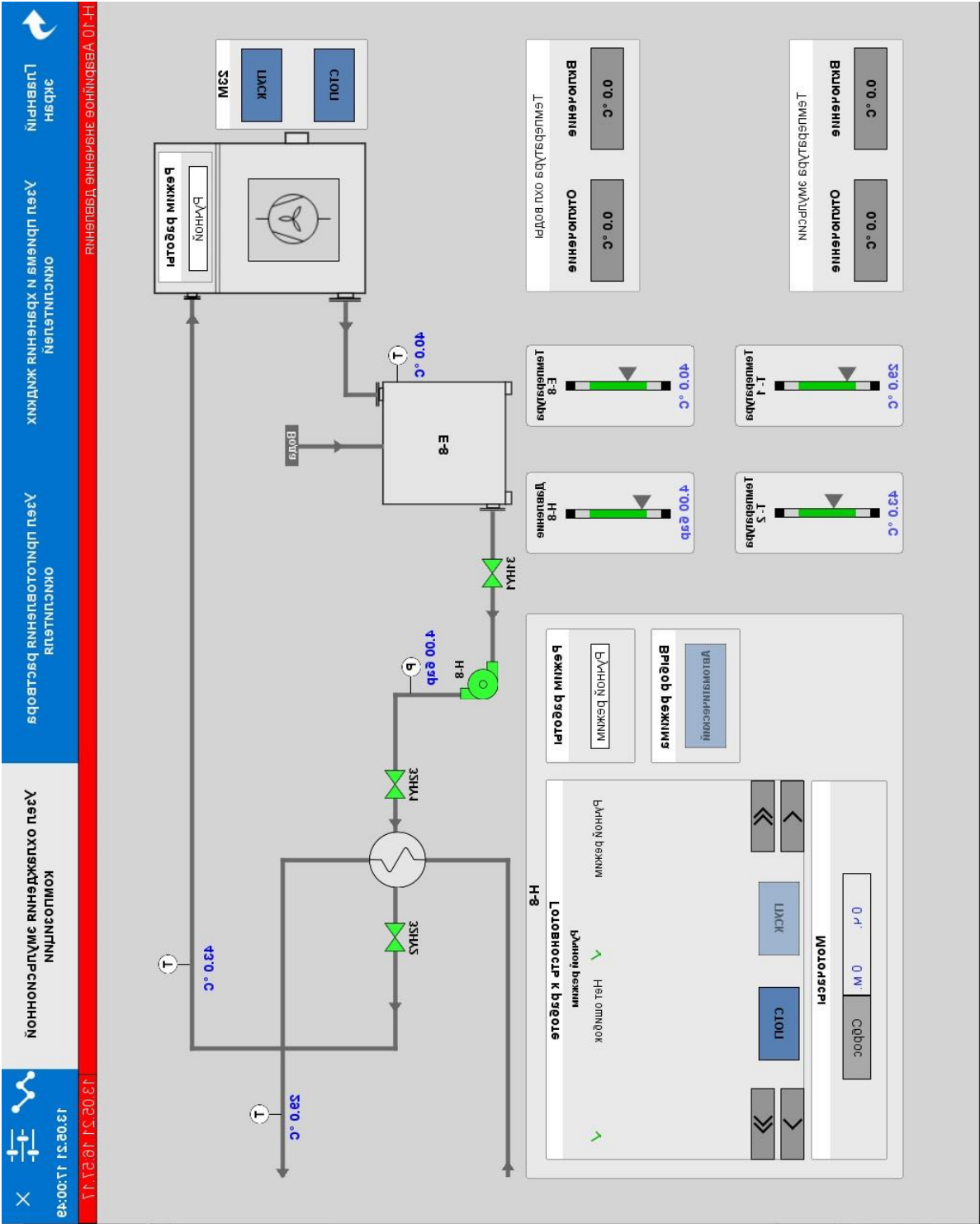


Рисунок А.4 – Зображення вікна вузла охолодження емульсійної композиції

Низька

Діагностична

Емісія E-3

Емісія E-12

Емісія E-14

Емісія E-9

Емісія E-10

Увага - Низька температура

Увага - Висока температура

Діагностична

Установка температури E-10

Увага - Низька температура

Увага - Висока температура

Діагностична

Установка температури E-10

Диф. сист. LV

Низька сист. LV

Висока сист. KV

Висока сист. KV

Вихід

0.0

0.0

0.0

0.0

0.0 %

Екран

Діагностична

Увага - Низька температура

Увага - Висока температура

Діагностична

Установка температури E-10

0.0

0.0

0.0

0.0

0.0 %

Рисунок А.5 – Зображення вікна задання уставок

ДОДАТОК Б
Текст програми

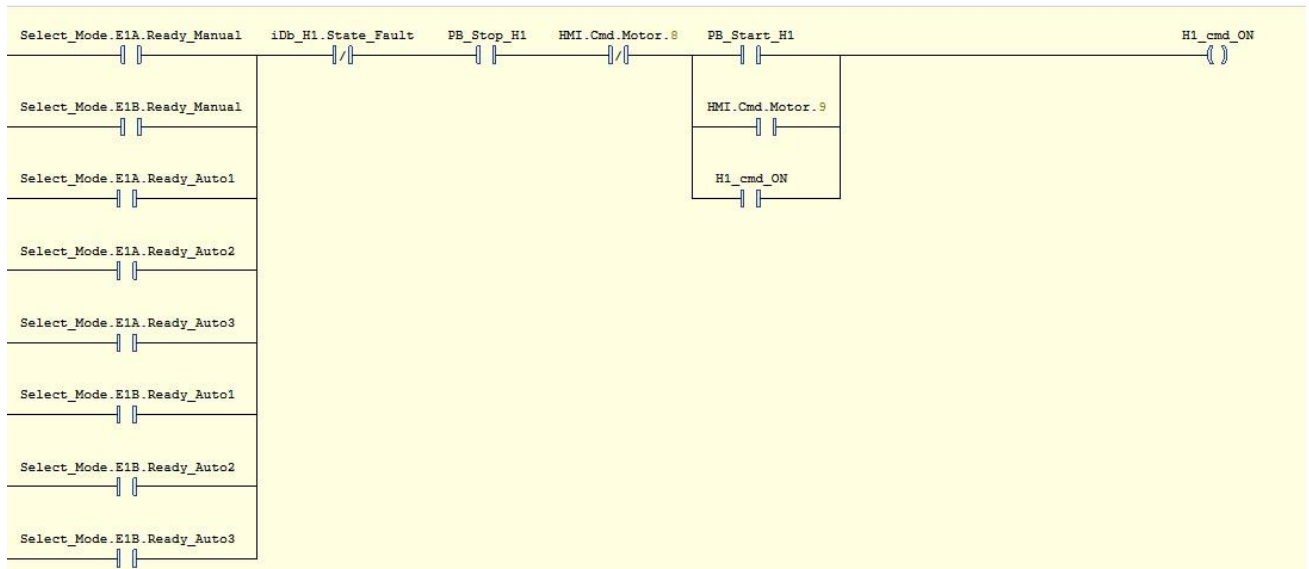


Рисунок Б.1 – Визначення готовності до запуску двигуна



Рисунок Б.2 – Визначення швидкості роботи двигуна

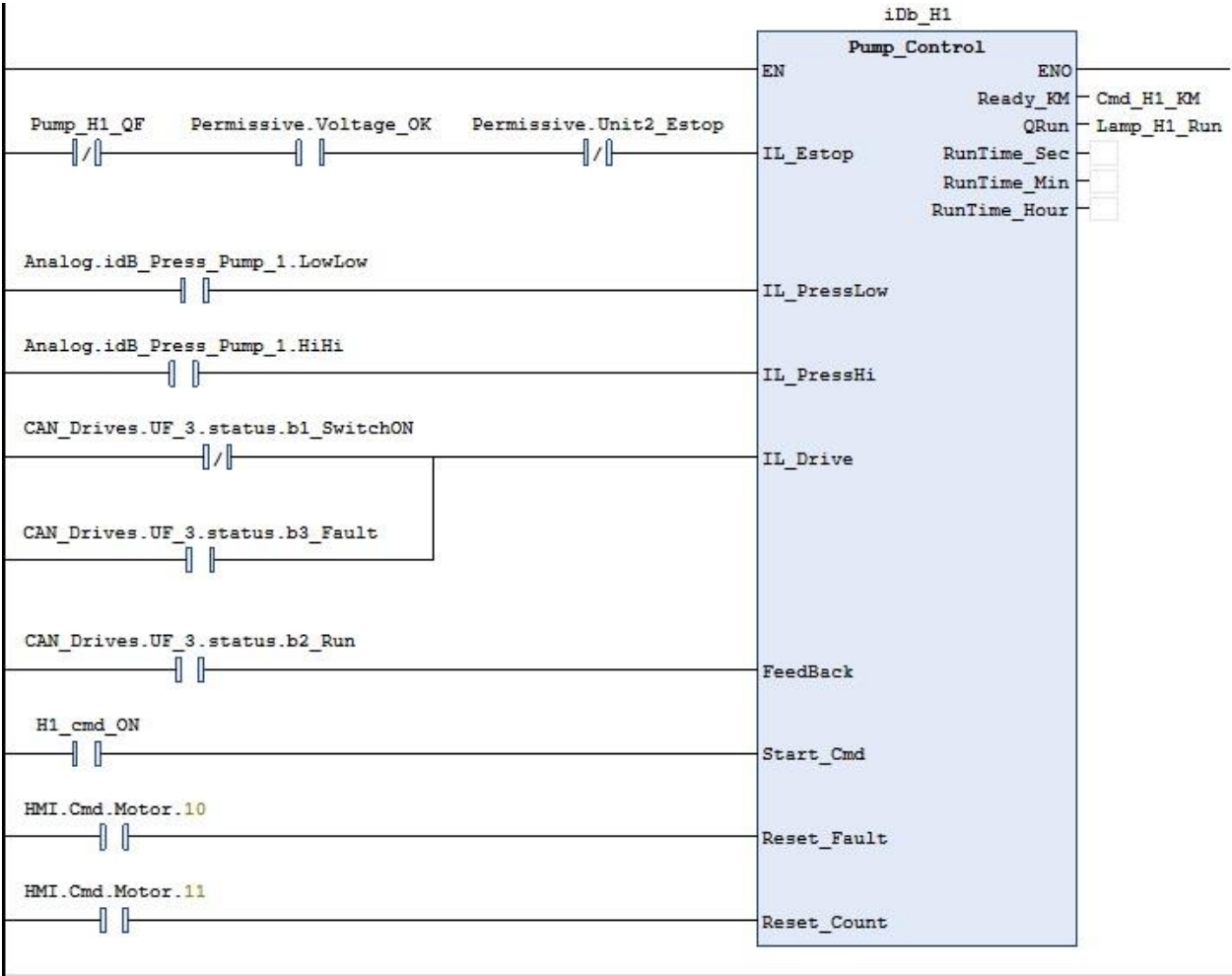


Рисунок Б.3 – Визначення стану двигуна

ДОДАТОК В
Слайди презентації

Національний університет Запорізька політехніка
Кафедра програмних засобів

Дипломна робота
на тему:
“Розробка людино-машинного інтерфейсу
оператора хімічної промисловості”

Виконав:
ст.гр. КНТз -117
Керівник:
к.т.н, доцент

А.О. Червоненко
С.М. Сердюк

2021

1

Рисунок В.1 – Слайд 1

Мета, об’єкт та предмет дослідження

Мета роботи – розробка людино-машинного інтерфейсу оператора АСУ ТП хімічної промисловості на основі методів, що підвищують ситуаційну обізнаність оператора в промисловому процесі.

Об’єкт дослідження – процес автоматизації діяльності оператора АСУ ТП хімічної промисловості.

Предмет дослідження – методи розробки людино-машинного інтерфейсу оператора АСУ ТП хімічної промисловості.

2

Рисунок В.2 – Слайд 2

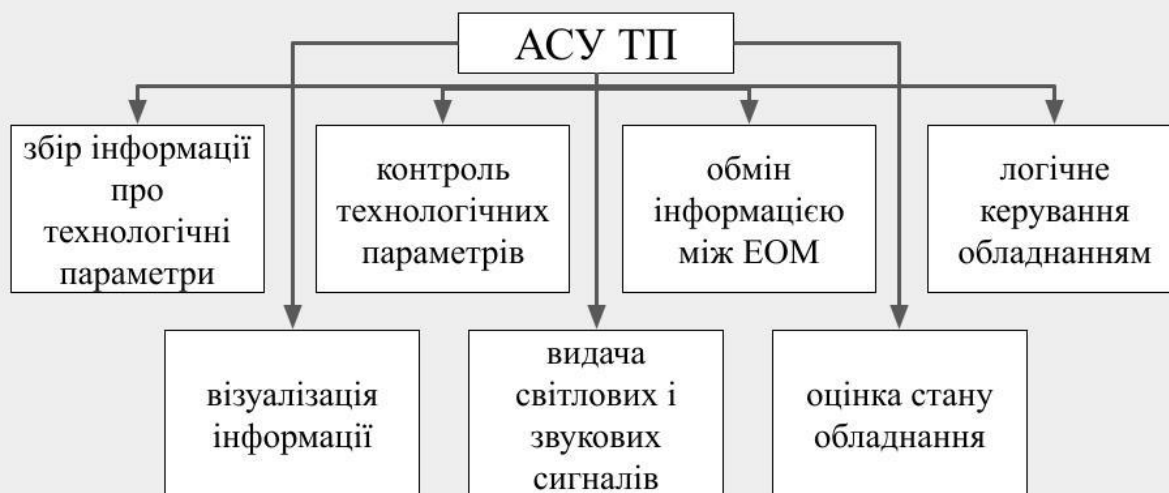
Аналіз предметної області

Автоматизована система управління (АСУ) – комплекс апаратних і програмних засобів для управління технологічним процесом. АСУ застосовуються в різноманітних галузях промисловості, енергетики, транспорті тощо. Термін «автоматизована» на противагу терміну «автоматична», зберігає роль управління процесом за людиною. Автоматизовані системи розрізняють за об'єктами управління: технологічними процесами – АСУ ТП; підприємством – АСУП [4].

3

Рисунок В.3 – Слайд 3

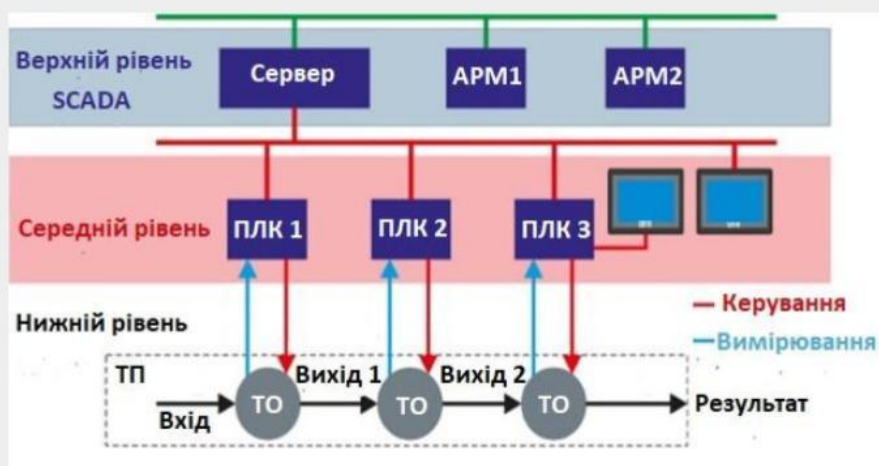
Основні функції автоматизованих систем управління технологічним процесом



4

Рисунок В.4 – Слайд 4

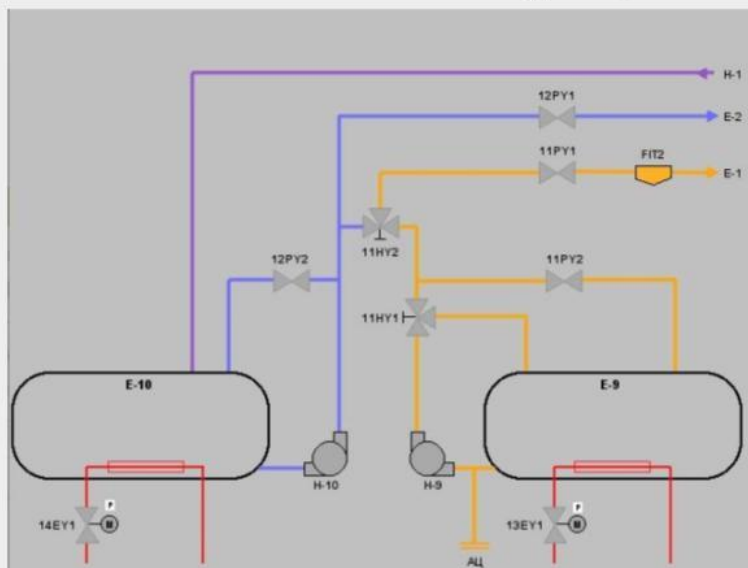
Структура автоматизованих систем управління технологічним процесом



5

Рисунок В.5 – Слайд 5

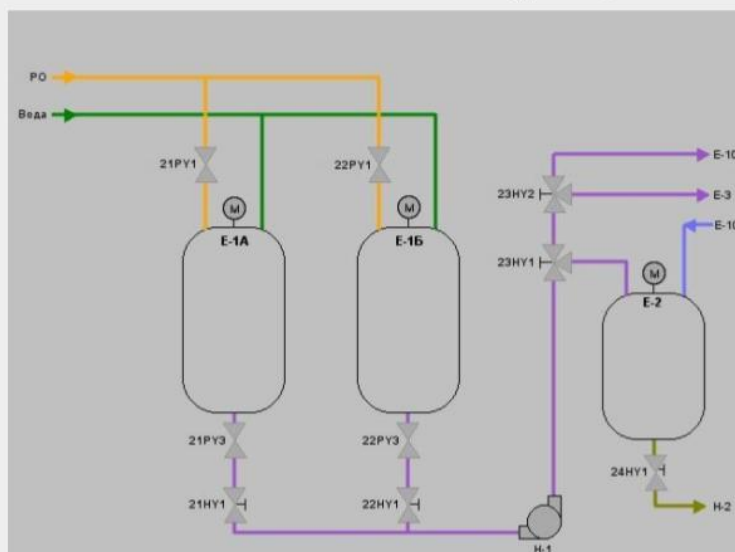
Технологічний процес



6

Рисунок В.6 – Слайд 6

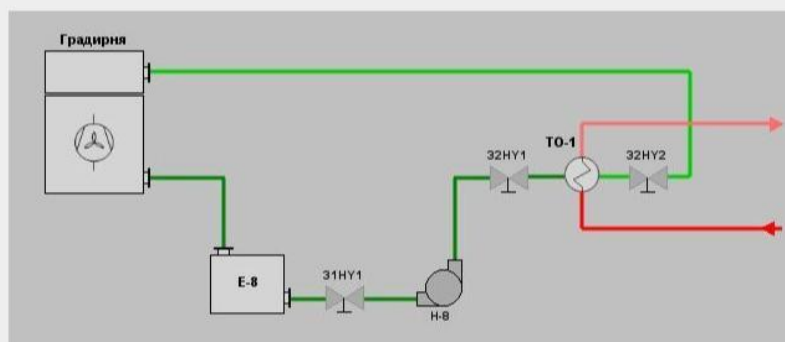
Технологічний процес



7

Рисунок В.7 – Слайд 7

Технологічний процес



8

Рисунок В.8 – Слайд 8

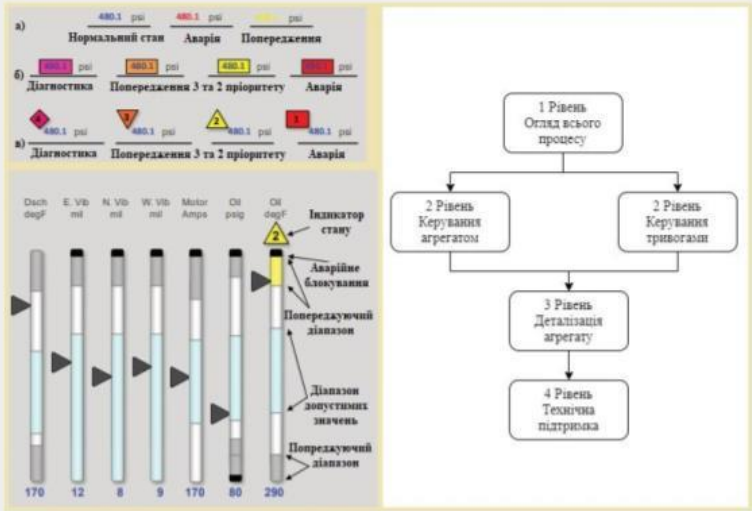
Завдання оперативного персоналу

№	Завдання	Користувач
1	Моніторинг ходу технологічного процесу	Оператор-технолог
2	Налаштування параметрів технологічного процесу	Оператор-технолог
3	Керування пуском і зупинкою технологічного процесу	Оператор-технолог
4	Аналіз динаміки технологічного процесу	Оператор-технолог
5	Налаштування датчиків і обладнання	Оператор-технолог
6	Налаштування датчиків і обладнання	Інженер з автоматизації
7	Контроль кабельної системи	Інженер з автоматизації
8	Виявлення характеру несправності та її локалізація	Інженер з автоматизації

9

Рисунок В.9 – Слайд 9

Вимоги до розробки людино-машинного інтерфейсу



- використання кольору спеціально і послідовно;
- відсутність анімацій, за винятком аварійних сигналів;
- аналогове представлення даних, з визначеними межами нормальних, попереджувальних і аварійних умов;
- логічні та послідовні методи навігації;
- заходи перевірки та безпеки;

10

Рисунок В.10 – Слайд 10

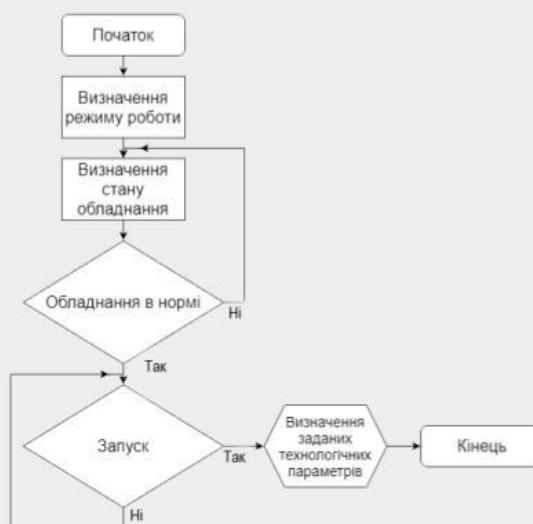
Вибір середовища розробки

Назва	Виробник	Єдине середовище проекту АСУ ТП	Гнучкість інструментів
CitectSCADA	Schneider Electric	Ні	Так
Master SCADA	InSAT	Так	Так
Trace Mode	AdAstra	Так	Так
Galileo	Eaton	Ні	Ні

11

Рисунок В.11 – Слайд 11

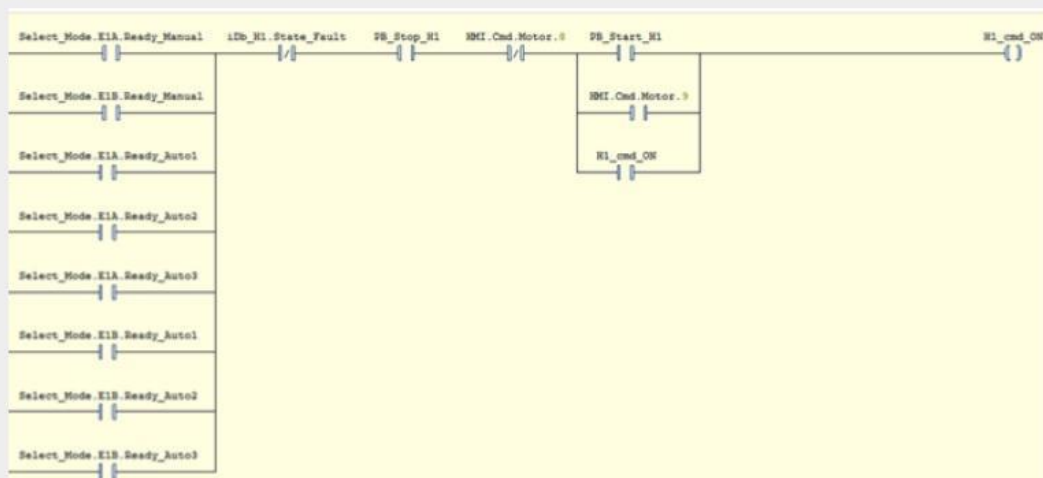
Алгоритм програми контролера



12

Рисунок В.12 – Слайд 12

Визначення режиму запуску



13

Рисунок В.13 – Слайд 13

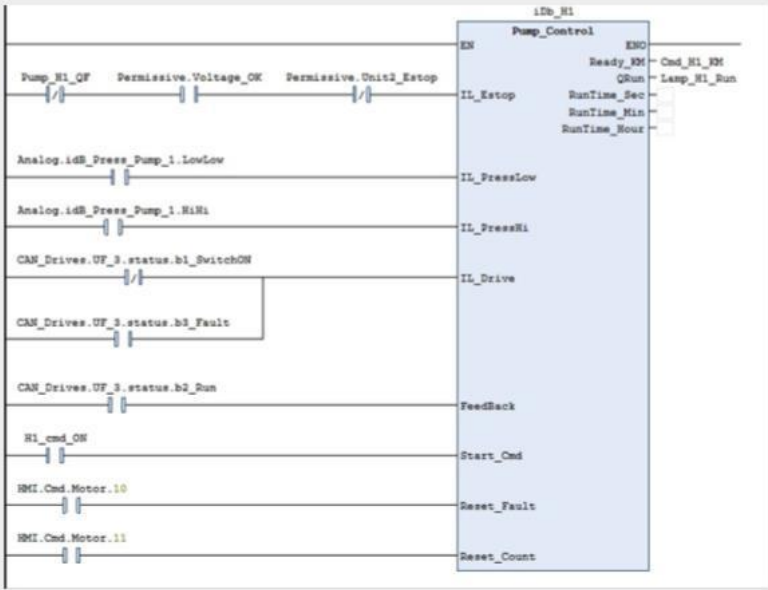
Визначення технологічних параметрів



14

Рисунок В.14 – Слайд 14

Визначення стану



15

Рисунок В.15 – Слайд 15

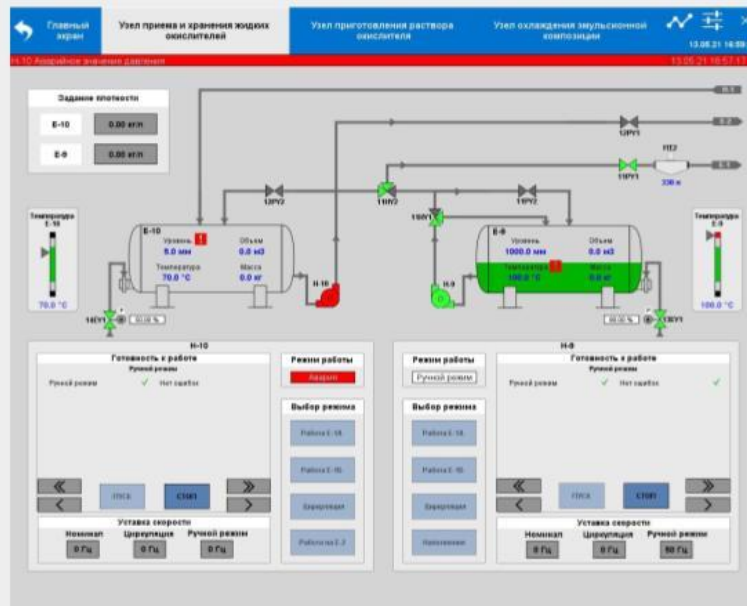
Розроблений інтерфейс



16

Рисунок В.16 – Слайд 16

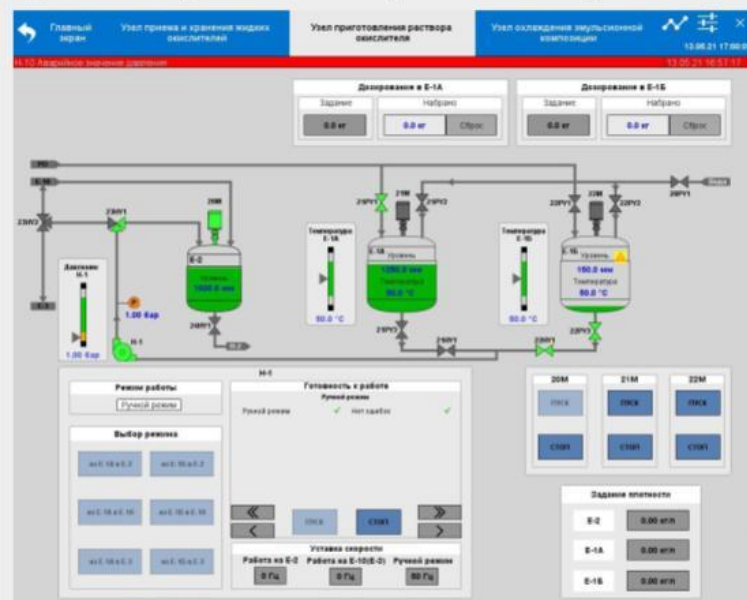
Вузол прийому і зберігання окислювачів



17

Рисунок В.17 – Слайд 17

Вузол приготування розчину окислювача



18

Рисунок В.18 – Слайд 18

