

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2025 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему:

**Програмне забезпечення мікропроцесорної системи
управління насосною станцією водозабору смт. Бобринець**

КБПЗ-2025

Виконав здобувач вищої освіти
IV курсу, групи КІ-21-2
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Іванченко О.В.
« ____ » _____ 2025 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Кислун О.А.
« ____ » _____ 2025 р.
Рецензент _____

Центральноукраїнський національний технічний університет

Факультет Механіко-технологічний

Кафедра Кібербезпеки та програмного забезпечення

Рівень вищої освіти бакалавр

Галузь знань 12 "Інформаційні технології"

Спеціальність 123 "Комп'ютерна інженерія"

Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

" " 2025 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Іванченко Олександр Володимирович

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи мікропроцесорної системи управління насосною станцією водозабору смт. Бобринець

2. Керівник роботи Кислун Олег Андрійович, канд. техн. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від "17" січня 2025 року №47-02

3. Строк подання роботи до захисту 22.05.2025 р.

4. Мета та завдання випускної кваліфікаційної роботи. *Метою роботи є розробка програмного забезпечення системи управління водозабором смт.Бобринець*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

2.Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію

6. Висновки.

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

7. Дата видачі завдання «__» _____ 20__ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем керування	10.03.2025	
2.	Постановка задачі, оформлення ТЗ	15.03.2025	
3.	Розробка моделі компонента	20.03.2025	
4.	Розробка структур даних	25.03.2025	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2025	
6.	Програмування алгоритмів	10.04.2025	
7.	Оформлення ПЗ	17.05.2025	
8.	Попередній захист роботи	22.05.2025	

Дата видачі завдання
«__» _____ 2025р.

Підпис керівника

_____ (прізвище та ініціали)

Завдання прийнято до виконання
«__» _____ 2025р.

Підпис здобувача

_____ (прізвище та ініціали)

АНОТАЦІЯ

Іванченко О.В. Програмне забезпечення мікропроцесорної системи управління насосною станцією водозабору смт. Бобринець.

123 Комп'ютерна Інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній кваліфікаційній бакалаврській роботі розроблено програмне забезпечення, яке призначено для управління насосною станцією.

Метою розробки є програмне забезпечення системи керування водозабором насосної станції смт.Бобринець.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows XP/Vista/7/8/10.

Програму розроблено в середовищі Visual C#.

Ключові слова: комп'ютерна інженерія, мікроконтролер, дані, керування, алгоритми.

ANNOTATION

Ivanchenko O.V. Software of the microprocessor control system of the water intake pumping station of the town of Bobrynets.

123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.

In this qualifying bachelor's thesis, software designed for the control of the pumping station was developed.

The purpose of the development is the software of the water intake control system for the supply system of the "Ingulska".

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on personal computers of IBM PC architecture with Windows XP/Vista/7/8/10.

The program was developed in the Visual C# environment.

Keywords: computer engineering, microcontroller, data, resources.

ВСТУП

Актуальність теми. Групи насосів різного призначення та насосні станції є одним із найпоширеніших типів обладнання у виробничих системах, а також у системах холодного та гарячого водопостачання, теплопунктах, котельнях, системах пожежогасіння та каналізації.

Розробники апаратури для автоматичного керування насосними групами дедалі активніше використовують сучасні електронні технології, зокрема мікропроцесорні блоки управління та електронні перетворювачі частоти (ПЧ). Завдяки розвитку обчислювальної техніки акцент у проектуванні зміщується з апаратного забезпечення на програмне, а гнучкість мікроконтролерів дозволяє впроваджувати більш складні та ефективні алгоритми управління.

Досвід розвинених країн свідчить, що на сучасному етапі інвестиції в удосконалення систем автоматизації управління є більш ефективними, ніж розробка нових видів обладнання та технологій. Це пояснюється стрімким розвитком електроніки, комп'ютерної техніки та інформаційних технологій, які підвищують інтелектуальний рівень керувальних алгоритмів і сприяють максимальному використанню можливостей наявного обладнання та технологічних процесів.

Водночас, попри високу вартість розробки систем управління (зокрема, їх алгоритмічного та прикладного програмного забезпечення) та дефіцит кваліфікованих спеціалістів, витрати на впровадження та експлуатацію таких систем поступово знижуються.

Це зумовлено зниженням вартості засобів автоматизації інтелектуального ядра, значним підвищенням їхньої надійності та спрощенням монтажно-налагоджувальних робіт. Основними джерелами економічної ефективності автоматизації технологічних процесів є: запобігання втратам через аварійні ситуації та їхні наслідки; зменшення витрат, зокрема скорочення витрат сировини

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

та енергетичних ресурсів; а також оптимізація чисельності персоналу підприємства.

Мета і завдання дослідження. Автоматизація управління технологічними процесами на базі комп'ютерних технологій може стати ключовим фактором у підвищенні ефективності підприємств. Для цього необхідно, по-перше, відійти від усталених стереотипів, які зводять комп'ютеризовану систему управління технологічними процесами (КСУ ТП) до суто централізованої системи, що виконує лише традиційні функції диспетчерського автоматизованого управління (ДАУ). По-друге, слід відмовитися від практики впровадження таких систем одночасно і в повному обсязі.

Альтернативним підходом є поетапне створення високотехнологічних підсистем КСУ ТП, які мають реалізовувати передусім нові, найбільш значущі для підприємства функції. Ці підсистеми повинні поступово формувати цілісну, функціонально структуровану систему управління технологічними процесами, наприклад, у вигляді автоматизованих робочих місць (АРМ). Такий підхід дозволить максимально ефективно використовувати обчислювальні можливості сучасних контролерів і комп'ютерів.

Практична цінність отриманих результатів полягає в тому, що розроблене програмне забезпечення та алгоритми сприятимуть удосконаленню системи керування насосною станцією, підвищенню якості наданих послуг та покращенню економічних показників. Отже, розробка програмного забезпечення для мікропроцесорної системи управління насосними станціями є актуальним завданням, яке потребує вирішення в межах цієї кваліфікаційної бакалаврської роботи.

Практична цінність отриманих результатів полягає в тому що розроблене ПЗ та алгоритми дозволять покращити систему управління насосною станцією, надавати якісні послуги та підвищити економічну ефективність. Таким чином, виходячи з вищеперерахованого, розробка програмного забезпечення для

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

мікропроцесорної системи управління насосними станціями, є актуальною задачею, яка потребує вирішення у даній кваліфікаційній бакалаврській роботі.

КБПЗ_2025

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Розроблювана автоматизована система призначена для моніторингу та регулювання технологічних параметрів водопідйомної свердловини. Вона контролює такі показники, як напруга живлення двигуна насоса, струм навантаження, тиск у трубопроводі, рівень води, «сухий хід» насоса, а також стан вимикачів і систем керування насосним обладнанням.

Система розрахована на експлуатацію в широкому діапазоні температур – від $-40\text{ }^{\circ}\text{C}$ до $+85\text{ }^{\circ}\text{C}$, при відносній вологості до 98 % (при $25\text{ }^{\circ}\text{C}$) та атмосферному тиску від 80 до 106 кПа (630–800 мм рт. ст.). Вона може бути інтегрована в обладнання для автоматизації водопідйомних станцій, забезпечуючи диспетчеризацію та централізоване управління кількома віддаленими системами. При розробці автоматизованої системи управління водозабором були передбачені наступні контрольовані параметри:

- При досягненні максимального рівня води в водонапірній башті (ВНБ) насос на свердловині повинен відключатися, при досягненні мінімального рівня води - включатися. Сигнали на включення і відключення глибинного насоса повинні передаватися за сигналами датчиків рівня.

Система повинна автоматично відключати глибинний насос при:

- зниженні рівня води в свердловині нижче допустимого (захист від «сухого ходу»);
- несправності насоса;
- несправність електрообладнання;
- автоматичне повторне включення (АПВ) насосного агрегату після появи раніше зниклої живлячої напруги.

Електричний захист:

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

- від неповнофазного режиму;
- можливість управління насосним агрегатом - ручне та автоматичне;
- ступінь захисту обладнання IP 54.

Індикація:

- стан насосного агрегату (робочий, позаробочий);
- стан електрифікованих засувок (закрыта, відкрита) , якщо є;
- індикація АПВ ;
- наявність напруги в мережі (усіх фаз);
- наявність води в свердловині;
- відображення аварійних ситуацій, їх архівування на протязі року;
- рівень води в резервуарі чистої води(РЧВ) або ВНБ;
- інформація про перелив РЧВ (ВНБ);
- струм електродвигуна працюючого насоса;
- про несанкціоноване проникнення в будівлі;
- про пониження температури повітря в будівлях нижче, ніж 5 С.

У систему входять:

- блок логіки;
- блок силовий;
- зовнішні датчики.

Блок силовий – це сукупність виконавчих пристроїв, які отримують команди від контролера та керують насосом водопідйомної станції, а також іншими силовими елементами системи.

Зовнішні датчики:

- датчик мінімуму - максимуму (манометр);
- датчик «сухого ходу» (манометр);
- додатково можуть бути встановлені датчики температури, датчики

охорони приміщення і.т.ін.

Можливості

Режими управління:

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

- автоматичний;
- ручний.

В автоматичному режимі, система контролює всі параметри і відпрацьовує повний цикл, без втручання людини. Всі дії і несправності (аварійні ситуації) відображаються на рідкокристалічному дисплеї, який знаходиться на передній панелі пристрою. Всі дані відображаються з прив'язкою за часом. При використанні централізованого контролю та управління, всі дії і несправності транслюються в диспетчерський пункт і відображаються на екрані персонального комп'ютера. Зв'язок з комп'ютером може бути як дротовий, так і бездротовий Wi - Fi. По одній парі проводів, послідовно може бути підключено до 255 пристроїв.

У ручному режимі всі операції проводяться за допомогою органів управління, що знаходяться на передній панелі.

За бажанням система може комплектуватися пристроєм плавного пуску двигуна, що значно збільшує термін служби насоса.

1.2 Область застосування

Мета кваліфікаційної роботи

Основною метою даної кваліфікаційної роботи є розробка автоматизованої системи управління насосною станцією для забезпечення таких функцій:

- автоматична підтримка заданого тиску води в колекторі відповідно до заданих параметрів;
- оптимізація режиму роботи насосів за допомогою частотного регулювання (ЧРП) з метою підвищення енергоефективності;
- оперативний диспетчерський контроль за основними параметрами технологічного процесу;
- виявлення аварійних ситуацій та несправностей технологічного обладнання з подальшим виведенням аварійно-попереджувальної сигналізації та фіксацією подій у журналі;

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

- обробка аналогових і дискретних сигналів відповідно до заданого алгоритму з формуванням необхідних команд управління;
- передача інформації про поточний стан обладнання та параметри процесу на верхній рівень керування (у разі роботи в складі автоматизованої системи управління технологічними процесами (АСУ ТП) підприємства).

Актуальність розробки. Розробка програмного забезпечення для мікропроцесорної системи управління насосною станцією є актуальним завданням, що потребує вирішення в межах даної кваліфікаційної бакалаврської роботи. Впровадження такої системи дозволить підвищити ефективність роботи насосного обладнання, знизити експлуатаційні витрати та покращити надійність управління водопостачанням.

КБПЗ - 2025

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським рівнем вищої освіти)

Сучасні автоматизовані системи управління технологічними процесами (АСУ ТП)

АСУ ТП є багаторівневою людино-машинною системою управління, що забезпечує ефективне керування складними технологічними процесами. Її створення здійснюється із застосуванням автоматизованих інформаційних систем збору даних та обчислювальних комплексів, які постійно вдосконалюються відповідно до розвитку технічних засобів і програмного забезпечення.

АСУ ТП і диспетчерське управління

Історичний розвиток АСУ ТП можна умовно поділити на три основні етапи, кожен з яких характеризується появою нових наукових ідей та технологічних можливостей.

Перший етап. Впровадження систем автоматичного регулювання (САР). На цьому етапі автоматизація охоплювала окремі параметри, установки та агрегати. Відбулося передавання функцій стабілізації, програмного управління та стеження від оператора до автоматичних регуляторів. Людина зосереджувалась на налаштуванні параметрів і розрахунку оптимальних режимів роботи.

Другий етап. Автоматизація технологічних процесів. Об'єктами управління стали розподілені у просторі технологічні системи. У цей період активно розвивались системи автоматичного управління (САУ), що дозволяли реалізовувати складні алгоритми керування, адаптивне управління та оптимізацію технологічних процесів. Особливістю цього етапу стало використання телемеханіки для дистанційного контролю та управління, що дозволило оператору

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

віддалятися від об'єкта, отримуючи інформацію через багаторівневі вимірювальні системи, виконавчі механізми та мнемосхеми.

Третій етап. Впровадження обчислювальної техніки в АСУ ТП. Розвиток мікропроцесорної техніки дозволив розширити можливості керування технологічними процесами. Спочатку це було впровадження мікропроцесорів у окремі підсистеми, а згодом – створення інтегрованих людино-машинних систем управління. На цьому етапі активно розвивались:

- методи інженерної психології та моделювання операцій;
- диспетчерські системи управління на основі автоматизованого збору даних;
- обчислювальні комплекси для централізованого контролю та оптимізації технологічних процесів.

Сучасні АСУ ТП поєднують потужні обчислювальні ресурси, автоматизовані системи збору даних і передові технології диспетчерського управління. Вони забезпечують високу ефективність управління, мінімізують вплив людського фактору та сприяють підвищенню продуктивності виробництва.

Роль диспетчера в багаторівневих АСУ ТП та проблема технологічного ризику. З розвитком автоматизованих систем управління технологічними процесами (АСУ ТП) змінювалися і функції диспетчера, який забезпечує регламентне функціонування технологічного процесу. Розширилося коло завдань, що вирішуються на рівні управління, доповнюючись новими викликами, які раніше вважались допоміжними або належали до інших рівнів керування.

Роль диспетчера в багаторівневих АСУ ТП та проблема технологічного ризику. З розвитком автоматизованих систем управління технологічними процесами (АСУ ТП) змінювалися і функції диспетчера, який забезпечує регламентне функціонування технологічного процесу. Розширилося коло завдань, що вирішуються на рівні управління, доповнюючись новими викликами, які раніше вважались допоміжними або належали до інших рівнів керування.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Функції диспетчера в сучасних АСУ ТП

Сучасний диспетчер працює у багаторівневій системі, отримуючи інформацію через монітори ЕОМ або електронні системи відображення даних. Він здійснює управління об'єктами, що можуть знаходитися на значній відстані, використовуючи телекомунікаційні технології, контролери та інтелектуальні виконавчі механізми.

Головними аспектами диспетчерського управління є:

- збір та обробка інформації – диспетчер повинен оперативно отримувати та аналізувати дані про стан об'єкта управління;
- прийняття рішень в режимі реального часу – особливо важливе в позаштатних та аварійних ситуаціях;
- взаємодія з автоматизованими системами – управління технологічним процесом в діалозі з ЕОМ;
- оцінка ризиків та їх мінімізація – критичний аспект для запобігання аварійним ситуаціям.

Проблема технологічного ризику. У таких галузях, як енергетика, нафтогазова промисловість та важка індустрія, технологічні процеси є потенційно небезпечними. Аварії можуть призводити до людських жертв, значних матеріальних збитків та екологічних катастроф.

Статистичні дослідження свідчать, що кількість аварій подвоюється приблизно кожні 10 років. Аналіз показав, що у 1960-х роках людський фактор був причиною аварій у 20% випадків, тоді як наприкінці 1980-х цей показник зріс до 80%.

Основні причини такої тенденції:

- недостатня увага до людино-машинного інтерфейсу (ЛМІ);
- надмірна орієнтація на технічні та технологічні досягнення без врахування зручності роботи оператора;
- відсутність ефективної підготовки диспетчерів до роботи в умовах підвищеного ризику.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Новий підхід до розробки АСУ ТП: орієнтація на диспетчера. Зростання ролі людського фактору в аварійних ситуаціях зумовило необхідність нового підходу до побудови диспетчерських систем управління. Основними напрямками удосконалення є:

- розвиток людино-машинного інтерфейсу – забезпечення інтуїтивно зрозумілого та адаптивного інтерфейсу, що мінімізує можливість помилок оператора;
- інтелектуальна підтримка рішень – використання систем аналізу даних, штучного інтелекту та експертних систем для прогнозування аварійних ситуацій;
- підготовка та навчання диспетчерів – впровадження тренажерів, імітаційних систем та навчальних модулів для відпрацювання дій у надзвичайних ситуаціях;
- автоматизація рутинних процесів – зменшення навантаження на диспетчера, що дозволяє зосередитися на критичних завданнях.

Ефективність сучасних АСУ ТП залежить не лише від передових технологій, а й від інтеграції оператора у систему управління. Орієнтація на диспетчера як ключову ланку у прийнятті рішень сприятиме зниженню аварійності, підвищенню надійності та безпеки технологічних процесів.

Концепція SCADA та її роль у сучасних системах управління. Розвиток SCADA (Supervisory Control And Data Acquisition – диспетчерське управління і збір даних) є закономірним результатом еволюції автоматизованих систем управління (АСУ). SCADA-технології забезпечують високий рівень автоматизації у сфері керування технологічними процесами, збору, обробки, передачі, зберігання та відображення даних.

Однією з ключових переваг SCADA є її потужний людино-машинний інтерфейс (HMI/MMI). Завдяки цьому диспетчер отримує:

- зручний і зрозумілий інтерфейс – дані візуалізуються у вигляді мнемосхем, графіків та таблиць, що полегшує моніторинг процесів;

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

- оперативне керування – доступ до механізмів керування у режимі реального часу;
- інтерактивні підказки та довідкові системи – зниження ймовірності помилок диспетчера;

Крім покращення взаємодії оператора з системою, SCADA також дозволяє:

- скорочувати терміни розробки проектів автоматизації – завдяки готовим бібліотекам компонентів та можливості швидкої конфігурації;
- оптимізувати фінансові витрати – впровадження SCADA значно дешевше, ніж створення спеціалізованих систем управління "з нуля".

Розвиток SCADA у світі та в Україні. Впровадження SCADA-систем у промисловості розпочалося у 1980-х роках у передових країнах Заходу. Вони активно використовуються у таких галузях, як:

- енергетика (електро- і водопостачання);
- хімічна та нафтохімічна промисловість;
- нафтопереробка та транспортування ресурсів;
- залізничний транспорт.

В Україні довгий час диспетчерське управління ґрунтувалося переважно на досвіді персоналу. Перехід до SCADA-систем розпочався пізніше, що було зумовлено низкою факторів:

- відсутність експлуатаційного досвіду – фахівці мали адаптуватися до нових методів роботи;
- недостатня інформованість про SCADA-рішення – кожна система є власною розробкою компанії-виробника, що обмежує доступ до детальної документації.

SCADA-системи є перспективним інструментом для автоматизованого управління складними процесами. Незважаючи на певні труднощі впровадження, їх використання сприяє підвищенню ефективності виробництва, зниженню аварійності та покращенню контролю над технологічними об'єктами. Подальше

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

розширення SCADA в Україні залежить від розвитку технічної бази, навчання персоналу та адаптації до світових стандартів автоматизації.

Ключові аспекти впровадження SCADA-систем

Ефективне впровадження SCADA-систем потребує вирішення двох основних завдань.

В Україні диспетчерське управління технологічними процесами спиралося, головним чином, на досвід оперативно-диспетчерського персоналу. Тому перехід до управління на основі SCADA-систем став здійснюватися кілька пізніше. До труднощів освоєння нової інформаційної технології, якими є SCADA-системи, відноситься як відсутність експлуатаційного досвіду, так і недолік інформації про різні SCADA-системах. У світі налічується не один десяток компаній, що активно займаються розробкою і впровадженням SCADA-систем. Кожна SCADA-система - це "know-how" компанії і тому дані по тій чи іншій системі не настільки великі.

Велике значення при впровадженні сучасних систем диспетчерського управління має рішення наступних завдань:

- а) вибору SCADA-системи (виходячи з вимог і особливостей технологічного процесу);
- б) кадрового супроводу.

Вибір SCADA-системи являє собою досить важке завдання, аналогічну прийняттю рішень в умовах багатокритеріальності, ускладнену неможливістю кількісної оцінки низки критеріїв через брак інформації.

Підготовка фахівців з розробки і експлуатації систем управління на базі програмного забезпечення SCADA здійснюється на спеціалізованих курсах різних фірм, курсах підвищення кваліфікації. В даний час в навчальні плани ряду технічних університетів почали вводитися дисципліни, пов'язані з вивченням SCADA-систем. Однак спеціальна література по SCADA-системам відсутня а є лише окремі статті і рекламні проспекти.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Компоненти систем контролю і управління та їх призначення

Узагальнена схема реалізації автоматизованих систем контролю і управління (СКУ).

Розробка автоматизованих систем контролю і управління (СКУ) для різних галузей промисловості ґрунтується на спільних принципах та структурних підходах. Узагальнену схему їх реалізації, представлену на рисунку. 2.1.

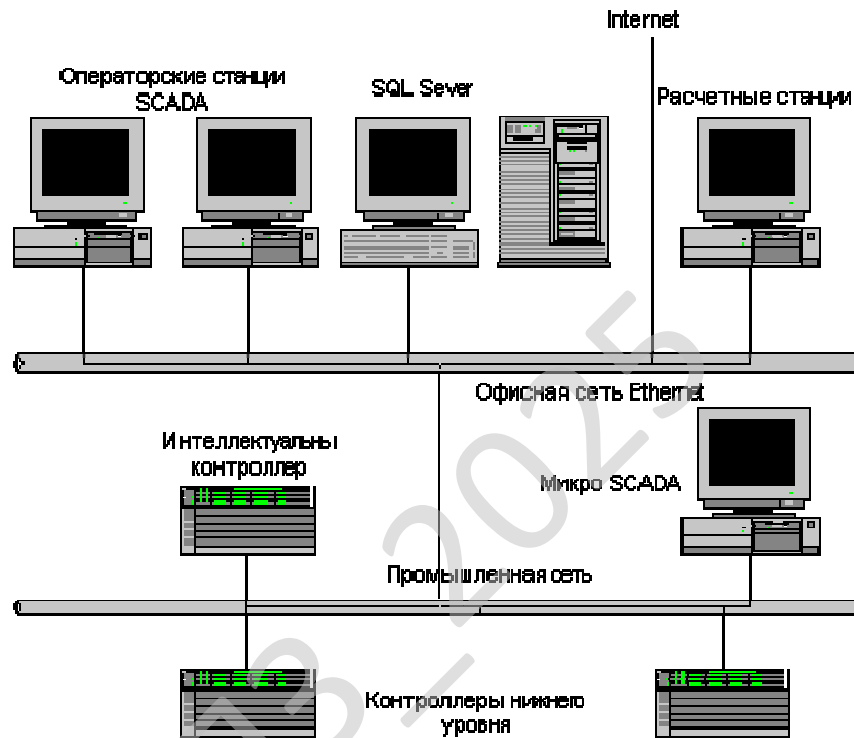


Рисунок 2.1 - Узагальнена схема системи контролю та управління

Як правило, це дворівневі системи, так як саме на цих рівнях реалізується безпосереднє управління технологічними процесами. Специфіка кожної конкретної системи управління визначається на кожному рівні програмно - апаратної платформи.

Нижній рівень - рівень об'єкта (контролерна) - включає різні датчики для збору інформації про хід технологічного процесу, електроприводи і виконавчі механізми для реалізації регулюючих і керуючих впливів. Датчики поставляють інформацію локальним програмованим логічним контролерам (PLC - Programming Logical Controoller), які можуть виконувати такі функції:

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

- збір та обробка інформації про параметри технологічного процесу;
- управління електроприводами і іншими виконавчими механізмами;
- рішення задач автоматичного логічного управління і ін.

Так як інформація в контролерах попередньо обробляється і частково використовується на місці, істотно знижуються вимоги до пропускну здатності каналів зв'язку.

Як локальних PLC в системах контролю і управління різними технологічними процесами в даний час застосовуються контролери як вітчизняних виробників, так і зарубіжних. На ринку представлені багато десятків і навіть сотні типів контролерів, здатних обробляти від декількох змінних до кількох сотень.

До апаратно-програмних засобів контролерного рівня управління висуваються жорсткі вимоги щодо надійності, часу реакції на виконавчі пристрої, датчики і т.ін. Програмовані логічні контролери повинні гарантовано відгукуватися на зовнішні події, що надходять від об'єкта, за час, визначений для кожної події.

Для критичних з цієї точки зору об'єктів рекомендується використовувати контролери з операційними системами реального часу (ОСРВ). Контролери під керуванням ОСРВ функціонують в режимі жорсткого реального часу.

Розробка, налагодження та виконання програм управління локальними контролерами здійснюється за допомогою спеціалізованого програмного забезпечення, широко представленого на ринку.

До цього класу інструментального ПО відносяться пакети типу ISaGRAF (CJ International France), InConrol (Wonderware, USA), Paradym 31 (Intellution, USA), що мають відкриту архітектуру.

Інформація з локальних контролерів може направлятися в мережу диспетчерського пункту безпосередньо, а також через контролери верхнього рівня. Залежно від поставленого завдання контролери верхнього рівня

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

(концентратори, інтелектуальні або комунікаційні контролери) реалізують різні функції. Деякі з них перераховані нижче:

- збір даних з локальних контролерів;
- обробка даних, включаючи масштабування;
- підтримку єдиного часу в системі;
- синхронізація роботи підсистем;
- організація архівів за обраними параметрами;
- обмін інформацією між локальними контролерами і верхнім рівнем;
- робота в автономному режимі при порушеннях зв'язку з верхнім рівнем;
- резервування каналів передачі даних і ін.

Верхній рівень - диспетчерський пункт (ДП) - включає, перш за все, одну чи кілька станцій управління, що представляють собою автоматизоване робоче місце (АРМ) диспетчера / оператора. Тут же може бути розміщений сервер бази даних, робочі місця (комп'ютери) для фахівців і т. ін. Часто в якості робочих станцій використовуються ПЕОМ типу IBM PC різних конфігурацій. Станції управління призначені для відображення ходу технологічного процесу і оперативного управління. Ці завдання і покликані вирішувати SCADA - системи. SCADA - це спеціалізоване програмне забезпечення, орієнтоване на забезпечення інтерфейсу між диспетчером і системою управління, а також комунікацію із зовнішнім світом.

Спектр функціональних можливостей визначено самої роллю SCADA в системах управління і реалізований практично у всіх пакетах:

- автоматизована розробка, що дозволяє створення ПЗ системи автоматизації без реального програмування;
- засоби виконання прикладних програм;
- збір первинної інформації від пристроїв нижнього рівня;
- обробка первинної інформації;
- реєстрація алармов і архівних даних;

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

- зберігання інформації з можливістю її пост-обробки (як правило, реалізується через інтерфейси до найбільш популярним базам даних);
- візуалізація інформації у вигляді мнемосхем, графіків і т.ін.;
- можливість роботи прикладної системи з наборами параметрів, що розглядаються як "єдине ціле" ("recipe" або "установки").

Розглядаючи узагальнену структуру систем управління, слід ввести і ще одне поняття - Micro-SCADA.

Micro-SCADA використовується в локальних інфраструктурах, де немає необхідності у повномасштабній SCADA-системі, але потрібен контроль і диспетчеризація. Вона виконує основні функції, характерні для SCADA-систем верхнього рівня, але орієнтована на автоматизацію конкретних галузевих завдань, тобто є вузькоспеціалізованою. На відміну від неї, SCADA-системи верхнього рівня є універсальними.

Усі компоненти системи управління взаємодіють між собою через канали зв'язку. За інтеграцію SCADA-систем із локальними та верхньорівневими контролерами, офісними та промисловими мережами відповідає комунікаційне програмне забезпечення. Вибір відповідного ПЗ для конкретної системи управління залежить від низки факторів, зокрема типу застосовуваних контролерів та SCADA-системи.

Оскільки системи управління безперервно отримують значний обсяг даних від пристроїв введення-виведення, вони використовують бази даних (БД). Головна їхня функція — оперативне надання інформації користувачам на всіх рівнях управління. Якщо на верхніх рівнях АСУ ця задача вирішується за допомогою традиційних БД, то для рівня АСУ ТП вона була проблемною. Раніше запис інформації в реальному часі здійснювався за допомогою програмного забезпечення інтелектуальних контролерів та SCADA-систем. Проте останнім часом з'явилися нові рішення, що забезпечують високошвидкісний запис і збереження даних у базах.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Швидкий розвиток Інтернету привернув увагу розробників SCADA-систем до його можливостей. Виникає питання: чи можуть інтернет-технології застосовуватися в управлінні технологічними процесами? Якщо так, то які рішення пропонують виробники?

При створенні спеціалізованого прикладного програмного забезпечення (ППЗ) для автоматизованих систем контролю та управління системний інтегратор або кінцевий користувач обирає один із двох шляхів:

- програмування на традиційних мовах із використанням стандартних засобів налагодження;
- застосування готових інструментальних COTS-рішень (Commercial Off-The-Shelf), спеціально розроблених для відповідних задач.

Другий підхід є очевидно вигіднішим, оскільки дозволяє спростити процес розробки ПЗ, скоротити витрати часу і коштів, а також залучити до процесу автоматизації не лише програмістів, а й фахівців-технологів.

Для складних розподілених систем розробка власного ППЗ з нуля є надто тривалою та витратною. Програмування виправдане лише у випадку простих систем або їхніх окремих фрагментів, для яких немає стандартних рішень (наприклад, відсутній потрібний драйвер) або існуючі варіанти не відповідають вимогам.

Найбільш популярні з них наведено в таблиці 2.1

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Таблиця 2.1 – Розповсюдженні системи SCADA

- Назва	- Розробник	- Країна
- InTouch	- Wonderware	- США
- Citect	- CI Technology	- Австралія
- FIX	- Intellution	- США
- Genesis	- Iconics Co	- США
- RealFlex	- BJ Software Systems	- США
- Sitex	- Jade Software	- ВБ
- TraceMode	- AdAstrA	- россія
- Cimplicity	- GE Fanuc	- США
- САРГОН	- НВТ - Автоматика	- россія

Широкий вибір SCADA-продуктів на ринку неминує ставить перед користувачем питання щодо вибору оптимального рішення. Визначення відповідної SCADA-системи є складним завданням, що нагадує процес прийняття рішень у багатокритеріальному середовищі.

Нижче представлено приблизний перелік критеріїв оцінки SCADA-систем, які мають ключове значення для користувачів. Цей список давно відомий і активно обговорюється в спеціалізованій літературі та профільних виданнях.

Виділяють три основні групи показників:

- технічні характеристики;
- вартісні показники;
- експлуатаційні особливості.

Програмно-апаратні платформи для SCADA-систем

Аналіз доступних платформ є важливим етапом, оскільки саме від нього залежить, чи можливо впровадити ту чи іншу SCADA-систему на наявному обладнанні, а також оцінити витрати на її подальшу експлуатацію. Зокрема, якщо прикладна програма розроблена в певному операційному середовищі, її

виконання може бути можливим і в інших середовищах, які підтримуються обраним SCADA-продуктом.

У різних SCADA-системах питання сумісності вирішується по-різному. Наприклад, FactoryLink підтримує широкий перелік програмно-апаратних платформ, що розширює можливості його застосування.

Програмно-апаратні платформи для SCADA-систем

Аналіз доступних платформ є важливим етапом, оскільки саме від нього залежить, чи можливо впровадити ту чи іншу SCADA-систему на наявному обладнанні, а також оцінити витрати на її подальшу експлуатацію. Зокрема, якщо прикладна програма розроблена в певному операційному середовищі, її виконання може бути можливим і в інших середовищах, які підтримуються обраним SCADA-продуктом.

У різних SCADA-системах питання сумісності вирішується по-різному. Наприклад, FactoryLink підтримує широкий перелік програмно-апаратних платформ, що розширює можливості його застосування.

Таблиця 2.2 – Список підтримуваних програмно-апаратних платформ

Операційна система	Комп'ютерна платформа
- DOS/MS Windows	- IBM PC
- OS/2	- IBM PC
- SCO UNIX	- IBM PC
- VMS	- VAX
- AIX	- RS6000
- HP-UX	- HP 9000
- MS Windows/NT	- Windows/NT на PC-платформі.

Деякі SCADA-системи, такі як RealFlex і Sitex, принципово базуються на єдиній операційній системі реального часу – QNX.

Водночас переважна більшість SCADA-рішень розроблені для платформ MS Windows. Саме ці системи забезпечують найповніші та легко розширювані інтерфейси взаємодії з користувачем (ММІ). Враховуючи домінуючі позиції Microsoft на ринку операційних систем, навіть такі компанії, як United States DATA Co (розробник FactoryLink), роблять ставку на подальший розвиток своїх SCADA-систем саме на базі Windows NT. Деякі виробники, які раніше використовували ОС реального часу (ОСРВ), також поступово переходять на платформу Windows NT.

ОСРВ продовжують широко застосовуватися у вбудованих системах, де їхні характеристики є особливо корисними. Проте основним напрямом розвитку SCADA-систем сьогодні залишається платформа Windows NT/2000, тоді як використання застарілих ОС, таких як MS DOS і ранні версії Windows, поступово скорочується.

Сучасні системи автоматизації відзначаються високим рівнем інтеграції. Вони включають різноманітні об'єкти управління, виконавчі механізми, реєстраційне та обчислювальне обладнання, операторські робочі місця, сервери баз даних тощо. Для ефективної роботи в такому неоднорідному середовищі SCADA-система має забезпечувати розвинені мережеві можливості. Оптимально, якщо вона підтримує стандартні мережеві середовища (ARCNET, ETHERNET тощо) з використанням загальноприйнятих протоколів (NETBIOS, TCP/IP тощо), а також може працювати з популярними промисловими інтерфейсами (PROFIBUS, CANBUS, LON, MODBUS тощо). Практично всі сучасні SCADA-системи відповідають цим вимогам, хоча набір підтримуваних мережевих інтерфейсів може відрізнитися залежно від конкретного рішення.

Вбудовані командні мови

Більшість SCADA-систем оснащені вбудованими мовами високого рівня, подібними до VBasic. Вони дозволяють створювати відповідні реакції на події, що виникають при зміні значень змінних, виконанні логічних умов, натисканні

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

комбінацій клавіш або ж при виконанні певних фрагментів коду із заданою періодичністю як для всього додатка, так і для окремих вікон.

Підтримувані бази даних

Однією з ключових функцій систем диспетчерського контролю та управління є обробка даних, що включає їх збір, аналіз, зберігання, стискання та передачу. Відповідно, у межах SCADA-системи необхідна наявність бази даних.

Більшість SCADA-систем, зокрема Genesis, InTouch, Citect, використовують синтаксис ANSI SQL, який є незалежним від конкретної бази даних. Це забезпечує віртуальну ізоляцію додатків, що дає змогу змінювати базу даних без значного впливу на прикладне програмне забезпечення, створювати автономні аналітичні програми та інтегрувати вже наявні рішення для обробки даних.

Графічні можливості

Як для розробників систем автоматизації, так і для технологів, які використовують ці системи, важливу роль відіграє графічний користувацький інтерфейс. Графічні можливості SCADA-систем у загальному досить схожі: кожна з них містить об'єктно-орієнтований графічний редактор з набором анімаційних функцій. Використання векторної графіки дозволяє здійснювати широкий спектр операцій з графічними об'єктами, а також забезпечує швидке оновлення зображень на екрані завдяки вбудованим анімаційним засобам.

Не менш важливою є підтримка стандартних функцій графічного інтерфейсу користувача (GUI). Оскільки більшість сучасних SCADA-систем працюють на платформі Windows, саме це визначає їхній GUI.

Відкритість систем

С SCADA-система вважається відкритою, якщо її формат даних та процедурний інтерфейс задокументовані та доступні для інтеграції сторонніх програмних компонентів.

Розробники систем автоматизації нерідко стикаються з необхідністю створення унікальних програмних модулів, не передбачених стандартним

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

функціоналом SCADA-системи. Саме тому відкритість системи є важливим фактором при її виборі. Відкритість передбачає доступ до специфікацій системних викликів SCADA, що дозволяє використовувати сервіси системи, такі як графічні функції, управління базами даних тощо.

Розробка власних програмних модулів

Розробники систем автоматизації нерідко стикаються з необхідністю створення унікальних програмних модулів, не передбачених стандартним функціоналом SCADA-системи. Саме тому відкритість системи є важливим фактором при її виборі. Відкритість передбачає доступ до специфікацій системних викликів SCADA, що дозволяє використовувати сервіси системи, такі як графічні функції, управління базами даних тощо.

Експлуатаційні характеристики

Критерії цієї групи є найбільш суб'єктивними, оскільки їхню оцінку важко здійснити лише за технічною документацією - краще один раз побачити, ніж сім разів почути.

Якщо користувач визначився з вибором SCADA-системи, наступним етапом стає розробка системи контролю та управління, що включає такі кроки:

a. Проєктування архітектури системи автоматизації – визначення функціонального призначення кожного її компонента.

b. Аналіз можливостей розподіленої архітектури – ухвалення рішень щодо впровадження вузлів із "гарячим резервуванням" та інших механізмів підвищеної надійності.

c. Розробка прикладної системи управління для кожного вузла – створення алгоритмів, що забезпечують виконання необхідних завдань автоматизації.

d. Налаштування параметрів прикладної системи – узгодження параметрів взаємодії пристроїв нижнього рівня, таких як програмовані логічні контролери (ПЛК), із зовнішнім середовищем (датчиками, виконавчими пристроями тощо).

e. Тестування та налагодження – перевірка працездатності створеної програми в режимі емуляції.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

У наступних розділах на прикладі двох відомих SCADA-систем (InTouch та Citect), що добре зарекомендували себе, розглянуто основні компоненти, функції та можливості систем диспетчерського управління та збору даних.

Scada-система Intouch компанії Wonderware

InTouch – це потужний людино-машинний інтерфейс (HMI), який використовується для промислової автоматизації, керування технологічними процесами та диспетчерського контролю. В Україні ця SCADA-система активно застосовується для створення розподілених систем керування (DCS) та інших автоматизованих систем управління (АСУ). InTouch належить до дев'ятого покоління HMI-рішень компанії Wonderware, яка є лідером у цій сфері.

Ця система широко відома завдяки своїм можливостям візуалізації та керування виробничими процесами. Вона забезпечує зручне середовище для розробки та потужний набір графічних засобів. У версії 9.5 реалізовано низку покращень, що сприяють підвищенню ефективності та продуктивності виробництва. Одним із ключових нововведень є технологія Wonderware® SmartSymbols, яка значно розширює можливості розробки та розгортання спеціалізованих автоматизаційних застосунків, що працюють у режимі реального часу.

Застосунки InTouch відзначаються гнучкістю та масштабованістю, що дозволяє їм відповідати як поточним, так і майбутнім вимогам без додаткових фінансових вкладень. Доступ до системи можна отримати через різні мобільні пристрої, мережевих клієнтів, комп'ютерні вузли та навіть через Інтернет. Крім того, відкритий та розширюваний інтерфейс InTouch забезпечує сумісність із широким спектром пристроїв промислової автоматизації.

Scada-система Citect компанії CI Technology

SCADA-система Citect є гнучким і універсальним рішенням, яке підходить як для невеликих об'єктів із десятками або сотнями параметрів, так і для масштабних проєктів із сотнями тисяч параметрів. Високий рівень масштабованості забезпечується модульною клієнт-серверною архітектурою, де

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

кожен функціональний модуль може працювати на окремому комп'ютері або розподілятися між кількома пристроями для підвищення продуктивності.

Завдяки цій архітектурі SCADA-система Citect може легко розширюватися відповідно до зростаючих потреб користувача, дозволяючи зберігати інвестиції в міру розвитку системи.

Вбудоване резервування

Багато виробничі процеси не допускають зупинки. У цьому випадку, як до апаратури, так і до програмного забезпечення, керуючому цим процесом, пред'являються підвищені вимоги по надійності. Резервування - загальновідомий спосіб підвищення надійності комп'ютерних систем, але в SCADA-системі Citect резервування є вбудованим, тобто що не вимагає написання додаткового прикладного програмного забезпечення для реалізації резервування. Citect допускає резервування будь-якого свого функціонального модуля, а також каналів зв'язку між модулями і між модулем і контролерами вводу / виводу.

Модульна, масштабована архітектура клієнт-сервер

SCADA-система Citect складається з п'яти функціональних модулів (серверів або клієнтів):

1. I / O - сервер вводу / виводу. Забезпечує передачу даних між фізичними пристроями введення / виводу і іншими модулями Citect.
2. Display - клієнт візуалізації. Забезпечує операторський інтерфейс: відображає дані, що надходять від інших модулів Citect і управляє виконанням команд оператора.
3. Alarms - сервер алармов (тривоги). Відстежує дані, порівнює їх з допустимими межами, перевіряє виконання заданих умов і відображає аларми на відповідному вузлі візуалізації.
4. Trends - сервер трендів. Збирає і реєструє трендову інформацію, дозволяючи відображати розвиток процесу в реальному масштабі часу або в ретроспективі.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

5. Reports - сервер звітів. Генерує звіти по закінченню певного часу, при виникненні певної події або за запитом оператора.

Кожен функціональний модуль Citect виповнюється як окрема задача незалежно від того, чи виконуються модулі на одному комп'ютері або на різних. Тому Citect дозволяє будувати як прості системи, коли всі модулі працюють на одному комп'ютері, так і складні, в яких функціональні модулі розподілені по окремих вузлах локальної мережі частково або повністю.

Scada-система FIX компанії Intellution

На першому етапі створення системи на основі програмного забезпечення FIX було організовано процес отримання даних від контролерів. Однією з ключових особливостей цього ПЗ є підтримка широкого спектра протоколів обміну даними, а також можливість одночасного використання кількох різних протоколів. Це дало змогу ефективно побудувати систему.

Сьогодні компанія Intellution активно просуває технологію OPC (OLE for Process Control), яка передбачає, що кожен виробник контролерів має забезпечити їхню сумісність із програмним забезпеченням для Windows. Завдяки здатності FIX працювати з різними контролерами ця система була застосована на Рубльовській і Східній водопровідних станціях, де поряд із контролерами ROC використовуються контролери Modicon, що працюють за протоколом Modbus+. Контролери Modicon забезпечують функціонування промислової мережі контролерів у межах одного або кількох розташованих поруч об'єктів.

Другий етап передбачав створення екранних форм для візуалізації отриманих даних. Однією з важливих функцій FIX є можливість відображати різні набори даних в межах однієї відеограми. Наприклад, 115 параметрів напору води в міській мережі групуються й відображаються поперемінно на одній відеоформі. Це дає змогу створити лише одну графічну схему, а система самостійно розподілить значення між групами.

Третій ключовий етап - інтеграція з іншими додатками. SCADA-система FIX підтримує стандартні для Windows механізми зв'язку через DDE і ODBC-

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

протоколи. Завдяки DDE фахівці інженерної служби можуть переглядати дані FIX у реальному часі безпосередньо в Excel. А використання ODBC дає можливість записувати інформацію в реляційні бази даних, зокрема в Access. Для розробників програмного забезпечення компанія Intellution надає спеціальний набір засобів, що спрощує доступ до баз даних та історичних файлів за допомогою мов програмування C++ і Visual Basic.

Scada-система Genesis32 компанії Iconics Co

Це комплекс програмних додатків, призначений для роботи в операційних системах Microsoft Windows 98, NT, ME, 2000, XP та 2003. Ця SCADA-система забезпечує збір даних і диспетчерське управління верхнього рівня у промисловій автоматизації. Вона включає середовище розробки та виконання сценаріїв VBA, що дає змогу створювати програмні компоненти за допомогою Microsoft Visual Basic for Applications 6.3, який входить до складу пакету MS Office 2000. Усі програмні модулі побудовані на багатопотоковій архітектурі та підтримують технологію ActiveX.

Genesis32 орієнтований на системних інтеграторів, які займаються розробкою великих проєктів у сфері автоматизованих систем керування технологічними процесами (АСУ ТП). Використання цієї SCADA-системи дозволяє гнучко налаштовувати створювані програми, забезпечуючи надійний контроль даних і підвищену продуктивність системи. Додатково для розробки користувацьких додатків передбачені такі інструменти Microsoft, як мова програмування VBA, а також скриптові мови VBScript і JScript.

Версія Genesis32 7.0 оптимізована для платформи MS Windows різних поколінь (95, 98, 2000, XP, W7/11), підтримує сучасні мережеві технології від Microsoft і складається з базових модулів: AlarmWorX32, GraphWorX32, TrendWorX32, ScriptWorX32, Security Server та Screen Manager. За необхідності система може бути розширена додатковими програмами, включаючи рішення від сторонніх розробників.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

SCADA-система Genesis32 широко застосовується у харчовій, фармацевтичній, нафтогазовій та енергетичній галузях.

Scada-система RealFlex компанії BJ Software Systems

Пакет RealFlex містить усі необхідні модулі для розробки та функціонування АСУ ТП. Він включає засоби конфігурації, процесори обробки даних у реальному часі та управління аварійними ситуаціями, а також інструменти для виконання розрахунків, обробки аналогових і дискретних сигналів, архівування історичних даних, візуалізації процесів і генерації звітів.

Додатково можуть бути включені наступні засоби: BJScan - зв'язок декількох RealFlex через глобальну мережу, RemFlex і LanFlex - утиліти для з'єднання з центральною БД RealFlex відповідно але по послідовному каналу або через локальну мережу, Control Sequence Language (CSL) - мова керуючих послідовностей, RealTalk - управ ляемая подіями мовної сигналізатор, Recipe Loader - завантажувач початкових установок, Statistical Process Control (SPC) - статистичний пакет управління процесами, TermFlex - доступ до RealFlex через алфавітноцифрової термінал, DDE Bridge - зв'язок RealFlex з додатками MS Windows 3.x. На рисунку показаний приклад конфігурації системи, побудованої на базі пакету RealFlex.

Система складається з двох незалежних підсистем, кожна з яких включає кілька операторських станцій.

Одна з підсистем містить «гарячий» резерв. Дві підсистеми обмінюються інформацією по послідовному каналу. В даний час RealFlex підтримує апаратуру багатьох відомих зарубіжних фірм-производителей контролерів і плат вводу-виводу. Список підтримуваних пристроїв постійно розширюється, тому на кличемо лише деякі з них: Allen Bradley, Honeywell серії TDC 3000, Metrabyte серії M1000 / M2000, Modicon по інтерфейсу J470, Yokogawa серій HR 2300, Simatic S5. В якості одного з найбільш перспективних засобів вводу-виводу аналогової і дискретної інформації від об'єктів АСУ ТП в рамках RealFlex використовуються комп'ютери MicroPC (Octagon Systems, США) з вбудованою в

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

ПЗУ або флешпам'ять операційною системою QNX 2.21 або QNX 4.22. Для RealFlex розроблені драйвери для вітчизняних контролерів TCM51, Ломіконт110, Реміконт110 і 130, ЕК2000, Ш711 і ін. RealFlex може поставлятися або у вигляді конфігурації для розробника, або у виконавчій конфігурації (Run time System). Крім цього, поставки RealFlex відрізняються за кількістю підтримуваних записів в БД RealFlex: 500 (MiniFlex) і 128000. RealFlex в даний час функціонує більш ніж на 5000 операторських станціях.

Scada-система Sitex компанії Jade Software

SCADA-система (Supervisory Control and Data Acquisition), розроблена англійською компанією Jade Software у 1995 році. Вона є відносно новою для російського ринку, але поєднує в собі сучасні концепції побудови SCADA-систем і багаторічний досвід роботи у сфері промислової автоматизації.

Цікаво, що Sitex була створена групою фахівців, які довгий час працювали з пакетом RealFlex компанії BJ Software Systems (BJSS, Х'юстон, США). Ключові розробники Sitex — Глин Дурбан, колишній дистриб'ютор RealFlex, та Барбара Джонсон, засновниця BJSS (чия компанія отримала назву від її ініціалів). Під час створення Sitex розробники враховували досвід впровадження RealFlex і сучасні вимоги до SCADA-систем.

Sitex був розроблений для задоволення найвищих вимог у сфері моніторингу та керування промисловими процесами. Він надає низку можливостей, яких зазвичай немає в інших SCADA-системах для ПК. При цьому його вартість є конкурентною серед рішень реального часу, що працюють на операційних системах зі стандартом POSIX.

Операційна система та архітектура

Sitex базується на операційній системі QNX, що забезпечує високу продуктивність і надійність. Завдяки мікроядерній архітектурі QNX підтримує систему абсолютних пріоритетів, яка ідеально підходить для задач реального часу. Вбудовані мережеві можливості забезпечують швидкий зв'язок, стійкість до відмов (FLEET), рівномірне навантаження та надлишковість мережевих ресурсів.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

Програми в QNX працюють у захищеному режимі, повністю використовуючи 32-розрядний код, що підвищує їхню стабільність. Це дозволяє Sitex здійснювати паралельну обробку даних: наприклад, одночасно відстежувати тренди у реальному часі та модифікувати бази даних без переривання основного процесу. Крім того, надійна файлова система QNX підвищує рівень захисту даних користувачів.

Графічний інтерфейс

Sitex використовує графічне середовище Open Look, яке базується на QNX Windows. Це забезпечує зручний інтерфейс для користувачів, дозволяючи працювати з кількома вікнами одночасно — всі вони оновлюються у реальному часі. У майбутньому планується випуск версії Sitex із підтримкою графічного середовища Photon. Працюючи в області АСУТП з різними SCADA-пакетами, я для себе умовно розділив їх на два класи. Перший клас - це пакети (наприклад RealFlex, VTC), що розробляються фахівцями в області АСУТП. У цих пакетах увага акцентується на базі даних системи, яка найбільшою мірою визначає об'єкт управління, а всі інші можливості (графіка, звіти і т. П.) Розглядаються як похідні. Sitex можна вважати типовим представником цього класу пакетів. Другий клас - це пакети (Genesis, Trace Mode), де на чільному місці стоять графічні можливості системи. Появі цих пакетів сприяв розвиток деякого графічного інструментарію і бібліотеки з елементами динамізації. Для цих пакетів, навпаки, база даних є похідною від графіки. У найбільш концентрованій формі цей підхід сформульований в гаслі "Намалюйте АСУ ТП".

Екранні форми дозволяють динамічно відображати зв'язки в реальному часі.

Scada-система TraceMode компанії AdAstra

Trace Mode - це популярна в Росії SCADA-система, призначена для розробки великих розподілених АСУТП широкого призначення. Trace Mode 5 створена в 1992 році фірмою AdAstra Research Group, Ltd (Росія) і до теперішнього часу має понад 10000 інсталяцій. Системи, розроблені на базі Trace

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

Mode, працюють в енергетиці, металургії, нафтової, газової, хімічної та інших галузях промисловості і в комунальному господарстві Росії. За кількістю впроваджень в Росії Trace Mode значно випереджає зарубіжні пакети подібного класу.

Trace Mode - заснована на інноваційних, що не мають аналогів технологіях. Серед них: розробка розподіленої АСУТП як єдиного проекту, автопостроєння, оригінальні алгоритми обробки сигналів і управління, об'ємна векторна графіка мнемосхем, єдиний мережевий час, унікальна технологія playback - графічного перегляду архівів на робочих місцях керівників. Trace Mode - це перша інтегрована SCADA- і softlogic-система, що підтримує наскрізне програмування операторських станцій і контролерів за допомогою єдиного інструменту.

Scada-система simplicity компанії GE Fabuc

Proficy SIMPLICITY — потужна, але проста у використанні SCADA-система з відкритою архітектурою та модульною структурою. Вона забезпечує доступ до даних у режимі реального часу, підтримуючи необмежену кількість точок збору інформації.

Основні переваги Proficy SIMPLICITY ПО Proficy HMI / SCADA - SIMPLICITY є найбільш ефективним і надійним рішенням на основі клієнт-серверної архітектури, яке надає всі необхідні засоби для організації постійного моніторингу, точного контролю і оперативного управління технологічними процесами, устаткуванням і ресурсами. Таким чином, на основі об'єктивних даних своєчасно вирішуються проблеми експлуатації обладнання, поліпшується якість управління, а з нею і якість продукції, підвищується рентабельність, скорочується період введення виробу в масове виробництво.

Основні характеристики Proficy HMI / SCADA - SIMPLICITY

Proficy SIMPLICITY - система диспетчерського контролю, збору і обробки даних (SCADA) людини-машинного інтерфейсу (HMI), призначена для автоматизації технологічних процесів (АСУТП). Це система модульної структури і відкритої архітектури, що забезпечує доступ до даних (з необмеженим числом

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

точок) в режимі реального часу, має виняткову міццю і в той же час надзвичайно проста у вивченні, впровадженні та експлуатації.

Proficy SIMPLICITY - розробка GeFanuc (спільного підприємства всесвітньо відомого індустріального лідера General Electric (США) і його японського партнера-компанії Fanuc). Система відома на ринку промислової автоматизації з 80-х років минулого тисячоліття. Серед користувачів СКАДА-системи SIMPLICITY - світові лідери самих різних галузей: AT & T, AlliedSignal, Chemical, BMW, Boeing, Chrysler, Dell, Duracell, Energizer, Ford, Fujitsu, Intel, Kodak, Land Rover, NASA, Nissan, Revlon, UPS, US Post Office, і ін.

SIMPLICITY має дуже привабливу вартість, порівнянну з ціною на вітчизняні СКАДА і набагато меншу, ніж у відомих аналогів зарубіжного виробництва.

Разом з апаратними засобами GE Fanuc (КПЛ контролерами) являє собою інструментарій для побудови ідеально збалансованих закінчених комплексів АСУТП. Як наслідок значне число успішних проектів і впроваджень на підприємствах росії та низки країн СНД.

Особливості

Широкий діапазон підтримуваних пристроїв і контролерів.

Збір даних можна здійснюватися з самих різних пристроїв. Підтримуються практично всі відомі моделі програмованих логічних контролерів (ПЛК), а вбудовані засоби OPC забезпечують можливість використання десятків тисяч пристроїв від різних виробників.

Дані перетворюються і візуалізуються в наочну інформацію (у вигляді чисел, тексту, сигналізації і графічних даних).

Розвинені засоби графіки та аварійної сигналізації «повідомляють» оператору і керівництву про точний стан ключових виробничих параметрів і показників на всіх значущих виробничих ділянках.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Інноваційні технології

SIMPLICITY - одна з найдосконаліших клієнт-серверних систем HMI відкритої архітектури, розроблена на основі стандартів і кодів Microsoft Win32, що дозволяє управляти найскладнішими додатками.

Ефективному управлінню виробничими процесами сприяють інноваційні технології SIMPLICITY, які включають в себе складні клієнти, прості клієнти на основі Інтернет-рішень і термінальних сервісів, підтримку роботи через портативні пристрої PDA "персональний цифровий секретар" і пейджингові функції. Застосування в SIMPLICITY MES-технологій, Web-додатків, бездротового зв'язку дає можливість приєднувати до системи все більше число користувачів.

Scada-система САРГОН компанії НВТ – Автоматика

Програмно-технічний комплекс "САРГОН" (Система Автоматизації енергетичного обладнання) - це вітчизняний комплекс для створення повнофункціональних АСУТП енергетичних об'єктів (енергоблоку, котлоагрегату, турбіни, цеху, станції).

ПТК "САРГОН" забезпечує новий рівень розробки, супроводу та експлуатації АСУТП за рахунок використання сучасних дружніх інтерфейсів, як для експлуатаційного персоналу, так і для всіх учасників процесу розробки, проектування та впровадження системи. Для досягнення такої якості ПТК "САРГОН" включає в себе вичерпний набір інструментальних програмних засобів, що автоматизує процес створення АСУТП і забезпечує дружній інтерфейс з його учасниками. При цьому дружнім вважається інтерфейс, який вибрав би фахівець даної предметної області для формального опису постановки розв'язуваної задачі. Власне рішення здійснюється на системному рівні відповідними компонентами ПТК "САРГОН".

ЗАТ "НВТ-Автоматика" розроблені бібліотеки типових алгоритмів автоматизації енергетичних установок всіх основних типів, типові структурні рішення, створені графічна і текстова оболонки для побудови автоматизованих

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

робочих місць (тка), потужна система наскрізного проектування (TkAconf) і повний набір інструментальних засобів, що забезпечують розробку, проектування і впровадження систем.

Для комплексної автоматизації ТЕС, теплових і компресорних станцій створено практично повний набір проектних рішень і програмного забезпечення, який дозволяє мінімізувати стадію розробки, зводячи її, переважно, до, адаптації та прив'язці типових рішень до конкретного об'єкта проектним шляхом.

ПТК САРГОН пройшов повторну експертизу РАО ЄЕС і отримав експертний висновок (від 30.04.02 р 159-АСУ-2002), рекомендований Департаментом стратегії розвитку і науково-технічної політики РАО ЄЕС для використання при створенні АСУТП енергетичних об'єктів.

АСУТП на базі ПТК САРГОН

АСУТП, побудована на базі ПТК "САРГОН", це сучасна система, яка:

1. реалізує повний набір інформаційних, управляючих і сервісних функцій;
2. дозволяє використовувати для створення АСУТП верхнього рівня стандартні технічні засоби і мережеві операційні системи;
3. використовує при створенні АСУТП інструментальні програмні засоби, що входять до ПТК "САРГОН";
4. забезпечує:
 - необхідні надійність і якість при оптимальній ціні;
 - напрацювання контролерів на відмову більше 100000ч;
 - висока надійність компонентів, можливість резервування;
 - відкритість і розширюваність системи;
 - відкриті протоколи і інтерфейси;
 - вбудована підтримка найбільш популярних стандартів;
 - функціонально-повний комплект інструментального ПО;
 - поставка прикладного ПО в початкових текстах.

Автоматизована система управління, створена на базі ПТК "САРГОН" є проектно-компонований і вільно-програмованої системою.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Використання SCADA-системи TRACE MODE у мікропроцесорній системі керування

У реалізації мікропроцесорної системи керування буде використана одна з найбільш розповсюджених SCADA-систем — TRACE MODE (ТРЕЙС МОУД).

Загальна інформація про TRACE MODE

SCADA-система TRACE MODE призначена для розробки великих розподілених АСКТП. Вона була створена у 1992 році компанією AdAstra Research Group Ltd і на сьогодні має понад 4500 інсталяцій.

ТРЕЙС МОУД - заснована на інноваціях, що не мають аналогів. Серед них: розробка розподіленої АСУТП як єдиного проекту, автопобудова, оригінальні алгоритми обробки сигналів і керування, об'ємна векторна графіка мнемосхем, єдиний мережевий час. ТРЕЙС МОУД - це перша інтегрована SCADA- і softlogic-система, що підтримує наскрізне програмування операторських станцій і контролерів за допомогою єдиного інструменту.

Основними функціями ТРЕЙС МОУД являються наступні:

- модульна структура — підтримка від 128 до 64000×16 I/O;
- необмежена кількість тегів;
- мінімальний цикл роботи системи — 0,001 с;
- відкритий формат драйверів для зв'язку з будь-якими УСО;
- підтримка мов програмування (Visual Basic, Visual C++ тощо);
- вбудовані бібліотеки (150+ алгоритмів обробки даних і керування);
- фільтрація;
- PID-, PDD-, нечітке та адаптивне регулювання;
- ШІМ;
- керування пристроями (клапанами, засувками, приводами);
- статистичні функції та довільні алгоритми.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

Технологія автопобудови

- автоматична генерація баз каналів операторських станцій і контролерів;
- автоконфігурація системи на основі інформації про контролери та УСО;
- підтримка контролерів Micro PC, Круїз, МФК, MIC2000, Advantech PCL тощо;
- автоматичне генерування каналів зв'язку з контролерами (Siemens, Allen Bradley, ADAM 4000/5000 тощо);
- автопобудова комунікацій між вузлами (RS-232/485, Profibus тощо).

При роботі в реальному часі технологія автопобудови відслідковує зміни бази каналів на різних вузлах розподіленої АСКТП (на операторських станціях і в контролерах) і автоматично проводить необхідні зміни. Так наприклад, якщо додати(видалити) датчик, ТРЕЙС МОУД автоматично додасть(видалить) і настроїть канали на усіх вузлах розподіленої АСУ.

Контролери та апаратне забезпечення

АСУТП реалізується на контролерах Lagoon та модулях серії I-7000 (аналоговий/дискретний ввід/вивід, комунікаційні модулі).

АСКТП реалізується на контролерах Lagoon, а також комунікаційних модулях, модулях аналогового вводу/виводу, модулях дискретного вводу/виводу серії I-7000.

Базовим процесорним модулем, що використовуються при розробці АСКТП є контролер I-7188 (I-7188 – аналог контролера Lagoon). Власне кажучи I-7188 - це маленький PC- сумісний комп'ютер. У ньому є процесор AMB 188-40МГц, 256 кбайт SRAM пам'яті (ОЗУ), електронний Flash-диск (аналог твердого диска) обсягом 512кбайт, годинник реального часу, 4 послідовних порти, тобто майже всі необхідні атрибути звичайного комп'ютера.

Програмне забезпечення написано мовою Visual C#. Ця мова обрана виходячи з наступних міркувань. Visual C# – строго типізована об'єктно-

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

орієнтована мова, призначена для розробки різноманітних безпечних і потужних застосунків, виконуваних у середовищі .NET Framework. Мовою Visual C# можна розробляти звичайні клієнтські застосунки Windows, веб-служби XML, розподілені компоненти, застосунки типу “ сервер-клієнт”, застосунки баз даних і багато яких інших. В Visual C# є розширений редактор коду, конструктори зі зручним користувальницьким інтерфейсом, вбудований відладник і багато інших засобів, покликані спростити розробку застосунків мовою Visual C# версії 5.0 і .NET Framework версії 4.5.

Синтаксис Visual C# дуже виразний, але простий у вивченні. Усі, хто знаком з мовами C, C++ або Java з легкістю визнають синтаксис із фігурними дужками, характерний для мови Visual C#. Розроблювачі, що знають кожен із цих мов, як правило, зможуть домогтися ефективної роботи з мовою Visual C# за дуже короткий час. Синтаксис Visual C# робить простіше те, що було складно в C++, і забезпечує потужні можливості, такі як типи значень Nullable, перерахування, делегати, лямбда-вираження й прямий доступ до пам'яті, чого немає в Java. Visual C# підтримує універсальні методи й типи, забезпечуючи більше високий рівень безпеки й продуктивності, а також ітератори, що дозволяють при реалізації колекцій класів визначати власне поводження ітерації, що може легко використовуватися в клієнтському коді. В Visual C# 5.0 вираження LINQ (Language-Integrated Query) роблять строго-типізований запит першокласною конструкцією мови.

Як об'єктно-орієнтована мова, Visual C# підтримує поняття інкапсуляції, спадкування й поліморфізму. Всі змінні й методи, включаючи метод Main – крапку входу застосунки – інкапсулюється у визначення класів. Клас може успадковувати безпосередньо з одного родового класу, але може реалізовувати будь-яке число інтерфейсів. Для методів, які перевизначають віртуальні методи в батьківському класі, необхідно ключове слово `override`, щоб виключити випадкове повторне визначення. У мові Visual C# структура схожа на полегшений клас: це тип, що розподіляється по стопках, що реалізує інтерфейси, але не підтримує спадкування.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

2.3 Розгорнута постановка завдання

Згідно з технічним завданням випускної кваліфікаційної дипломної роботи, необхідно розробити програмне забезпечення для мікропроцесорної системи керування насосною станцією. У межах кваліфікаційної роботи д слід виконати такі етапи:

а) проаналізувати існуючі аналогічні системи для виявлення їхніх переваг і недоліків, а результати цього аналізу врахувати під час подальшої розробки;

б) обґрунтувати вибір методики побудови автоматизованої системи контролю роботи технологічного обладнання, а також розробити її функціональну та структурну схеми;

в) створити програмне забезпечення, що відповідатиме технічним вимогам, а також розробити блок-схеми алгоритмів і підпрограм;

г) спроєктувати інтерфейс користувача, який забезпечуватиме виведення повідомлень про некоректні дії оператора та нестандартні ситуації в роботі обладнання;

д) підготувати рекомендації щодо організаційних і методичних заходів, необхідних для впровадження системи в промислову експлуатацію та її подальшого використання;

е) сформулювати висновки про виконану роботу та отримані результати.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

- автоматичне повторне включення (АПВ) насосного агрегату після появи раніше зниклої живлячої напруги.

Електричний захист:

- від неповнофазного режиму;
- можливість управління насосним агрегатом - ручне та автоматичне;
- ступінь захисту обладнання IP 54.

Індикація:

- стан насосного агрегату (робочий, позаробочий);
- стан електрифікованих засувок (закрыта, відкрита) , якщо є;
- індикація АПВ ;
- наявність напруги в мережі (усіх фаз);
- наявність води в свердловині;
- відображення аварійних ситуацій, їх архівування на протязі року;
- рівень води в резервуарі чистої води(РЧВ) або ВНБ;
- інформація про перелив РЧВ (ВНБ);
- струм електродвигуна працюючого насоса;
- про несанкціоноване проникнення в будівлі;
- про пониження температури повітря в будівлях нижче, ніж 5 С.

У систему входять:

- блок логіки;
- блок силовий;
- зовнішні датчики.

Блок силовий це сукупність виконавчих пристроїв, які отримують команди від контролера і керуючих насосом водопідйомної станції та іншими силовими елементами.

Зовнішні датчики:

- датчик мінімуму - максимуму (манометр);
- датчик «сухого ходу» (манометр);

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

- додатково можуть бути встановлені датчики температури, датчики охорони приміщення і.т.ін.

Можливості:

режими управління:

- автоматичний;
- ручний.

В автоматичному режимі, система контролює всі параметри і відпрацьовує повний цикл, без втручання людини. Всі дії і несправності (аварійні ситуації) відображаються на рідкокристалічному дисплеї, який знаходиться на передній панелі пристрою.

Всі дані відображаються з прив'язкою за часом.

При використанні централізованого контролю та управління, всі дії і несправності транслюються в диспетчерський пункт і відображаються на екрані персонального комп'ютера. Зв'язок з комп'ютером може бути як дротовий, так і бездротовий Wi Fi. По одній парі проводів, послідовно може бути підключено до 255 пристроїв.

У ручному режимі всі операції проводяться за допомогою органів управління, що знаходяться на передній панелі.

За бажанням система може комплектуватися пристроєм плавного пуску двигуна, що значно збільшує термін служби насоса.

3.2 Розробка структурної схеми

Структурна схема системи являє собою сукупність об'єктів, їхніх компонентів і взаємозв'язків між ними. Вона слугує для наочного відображення загальної організації системи, її основних модулів, функціональних блоків та ключових зв'язків між ними.

Структурну схему мікропроцесорної системи представлено на рисунку 3.1.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

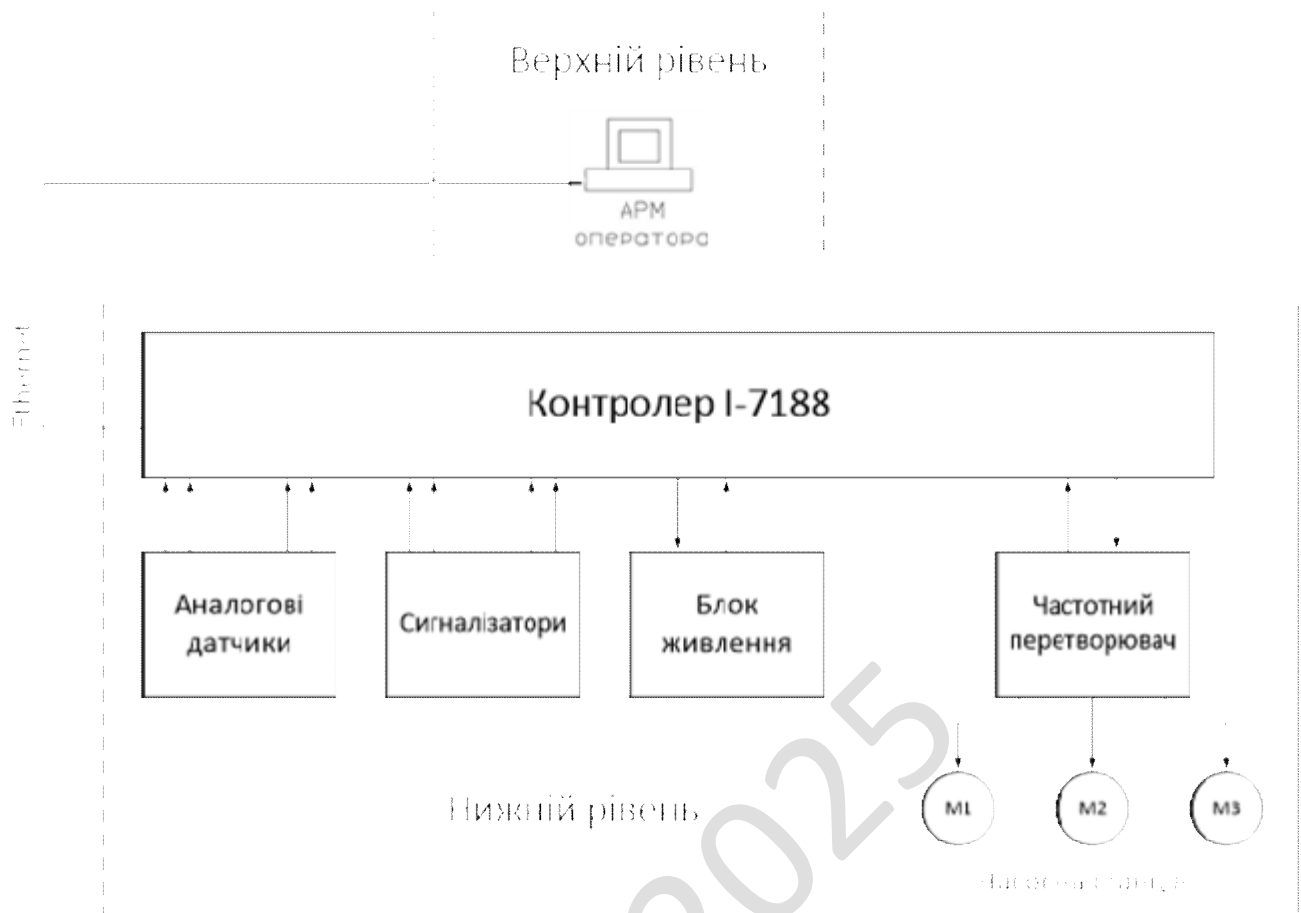


Рисунок 3.1 - Структурна схема системи керування

Зі схеми видно, що система має два рівні: верхній та нижній. Нижній рівень включає основні компоненти, такі як датчики, насоси, частотний перетворювач і контролер, який керує обладнанням нижнього рівня та забезпечує зв'язок із верхнім рівнем. Верхній рівень представлений персональним комп'ютером (ПК) із встановленим спеціалізованим програмним забезпеченням для моніторингу та управління системою.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

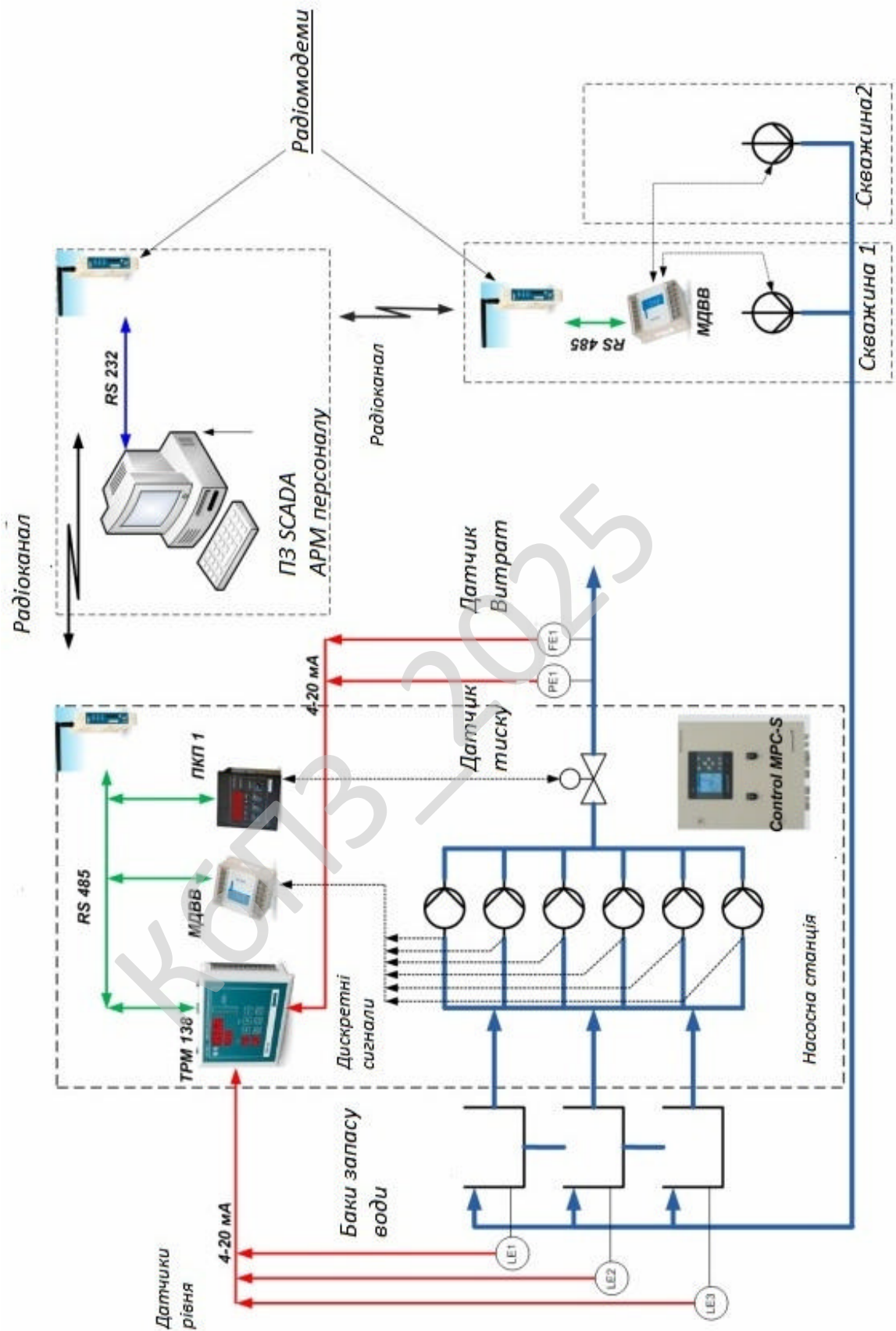


Рисунок 3.2 – Функціональна схема системи

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРБ-123.25.0032.00.00.ПЗ

Арк.

45

Автоматизована система управління водопостачанням (АСУ) є апаратно-програмним комплексом (АПК), призначеним для забезпечення безперебійної подачі води споживачам.

АПК складається з програмної частини - SCADA-пакета MasterSCADA фірми ІНСАТ, що встановлюється на ПК в приміщенні, чергового персоналу, і апаратної частини, яка розроблена на базі мікропроцесорних пристроїв ОВЕН.

Автоматизована система водопостачання здійснює:

- автоматичне управління 2-ма віддаленими свердловинами;
- дистанційне та автоматичне керування 6-ю мережевими насосами з

АРМ оперативного персоналу;

- вимірювання рівня в ємностях запасу води, тиску води в системі, витрата води до споживачів;

- управління пожежною засувкою по витраті або тиску;

- вивід інформації та аварійної сигналізації на АРМ оперативного персоналу.

Опис роботи АСУ

На АРМ виводяться такі показники:

- рівень води в резервуарах запасу (датчики L1, L2, L3);
- тиск у трубопроводі (P1);
- витрата води до споживачів (F1).

При зростанні споживання води тиск у системі знижується, що призводить до автоматичного ввімкнення додаткового насоса.

MasterSCADA також контролює перевищення допустимих витрат і, якщо значення виходить за верхню межу уставки, автоматично перемикає засувку на подачу води в обхід блоку водопідготовки.

Управління клапаном можливе як у ручному режимі з АРМ, так і безпосередньо через пристрій ПКП1.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

Рівень води в резервуарах контролюється за допомогою датчиків тиску. Дані відображаються у цифровому вигляді на дисплеї ТРМ138, а також у цифровому та графічному вигляді на моніторі АРМ:

- при зниженні рівня води нижче мінімального значення (MIN), MasterSCADA подає команду на ввімкнення свердловин;
- при перевищенні максимального рівня (MAX) – команда на відключення насосів.

Зв'язок із модулем МДВВ, встановленим у приміщенні свердловини №1, здійснюється по радіоканалу через велику віддаленість.

Усі отримані параметри архівуються без обмежень за глибиною збереження даних і можуть відображатися у вигляді цифрових значень, графіків або таблиць.

В системі використано обладнання ОВЕН:

- перетворювач гідростатичного тиску (занурювальний) ОВЕН ПД100-ДГ0,06 – 3 шт;
 - перетворювач надлишкового тиску ОВЕН ПД100-ДІ (діапазон вимірювання 0...1,6 МПа, вихідний сигнал 4...20 мА) – 2 шт;
 - універсальний 8-канальний вимірювач-регулятор ОВЕН ТРМ138 – 1 шт;
 - модуль дискретного введення-виведення ОВЕН МДВВ – 1 шт;
 - модуль дискретного вводу ОВЕН МВ110-16ДН – 1 шт;
 - модуль дискретного виводу ОВЕН МУ110-16Р – 1 шт;
 - прилад контролю і управління засувкою ОВЕН ПКП1 – 1 шт;
 - блоки живлення ОВЕН БП14Б-Д4.4.24 – 2 шт;
 - блоки живлення ОВЕН БП30Б-Д3.24 – 2 шт;
 - блок мережевого фільтра ОВЕН БСФ-Д3-1,2 – 1 шт;
 - перетворювачі інтерфейсів ОВЕН АС4 і ОВЕН АС5 – по 1 шт.
- Також у складі системи використано обладнання інших виробників:
- радіомодем НЕВОД5 – 4 шт;

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

- антена SIRO SA703 – 1 шт;
 - антени АН5-433 – 2 шт;
- витратомір РСМ-05 – 1 шт.

Прилади в шафі автоматики №1

- універсальний 8-канальний вимірювач-регулятор ТРМ138.Р-Щ7 з RS-485 – 1 шт%
- пристрій управління і захисту електроприводу засувки без кінцевих вимикачів ПКП1-ТИ з RS-485 – 1 шт;
- модуль дискретного введення МВ110-16ДН з RS-485 – 1 шт;
- модуль дискретного виводу МУ110-16Р з RS-485 – 1 шт;
- радіомодем для систем телеметрії НЕВІД-5 – 1 шт.

Прилади в шафі автоматики №2:

- модуль дискретного введення-виведення МДВВ з RS-485 (12 каналів дискретного введення, 8 каналів дискретного виводу) – 1 шт;
- радіомодем для систем телеметрії НЕВІД-5 – 1 шт;
- прилади на АРМ оператора;
- радіомодем для систем телеметрії НЕВІД-5 – 1 шт4
- автоматичний перетворювач АС4 USB / RS-485 – 1 шт;
- персональний комп'ютер зі встановленою SCADA-системою MasterSCADA MSRT100 (на 100 точок введення-виведення).

Після аналізу всіх блоків функціональної схеми переходимо до розгляду діаграми взаємодії процесів, які відбуваються в системі.

3.4 Розробка діаграми процесів

Аналіз розробленої діаграми процесів

На рисунку 3.3 представлена діаграма процесів, яка демонструє взаємодію компонентів у системі. Основна мета такої діаграми — графічно відобразити

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

склад сукупностей даних та їхні взаємозв'язки, що дає змогу проводити детальний аналіз функціонування системи.

Графічне представлення даних може ґрунтуватися як на абсолютних, так і на відносних показниках, що дозволяє глибше аналізувати систему та визначати можливі шляхи її вдосконалення.

Призначення діаграми взаємодії процесів

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних під час структурного проектування.

Зазвичай, процес розробки починається з побудови контекстної діаграми, яка показує загальну взаємодію системи із зовнішніми елементами. Надалі ця діаграма деталізується шляхом розширення процесів та потоків даних, що дозволяє краще зрозуміти структуру та механізм функціонування системи.

При детальному аналізі можна побачити всі аспекти взаємодії у розробленій системі.

Використовується модель проектування, яка дозволяє графічно представити "потоки" даних у системі.

При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі.

Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграми потоків даних містять чотири типи елементів:

- зовнішні по відношенню до системи сутності;
- потоки даних між елементами трьох попередніх типів;
- процеси які являють собою трансформацію даних в рамках описуваної системи;
- сховища даних (репозиторії).

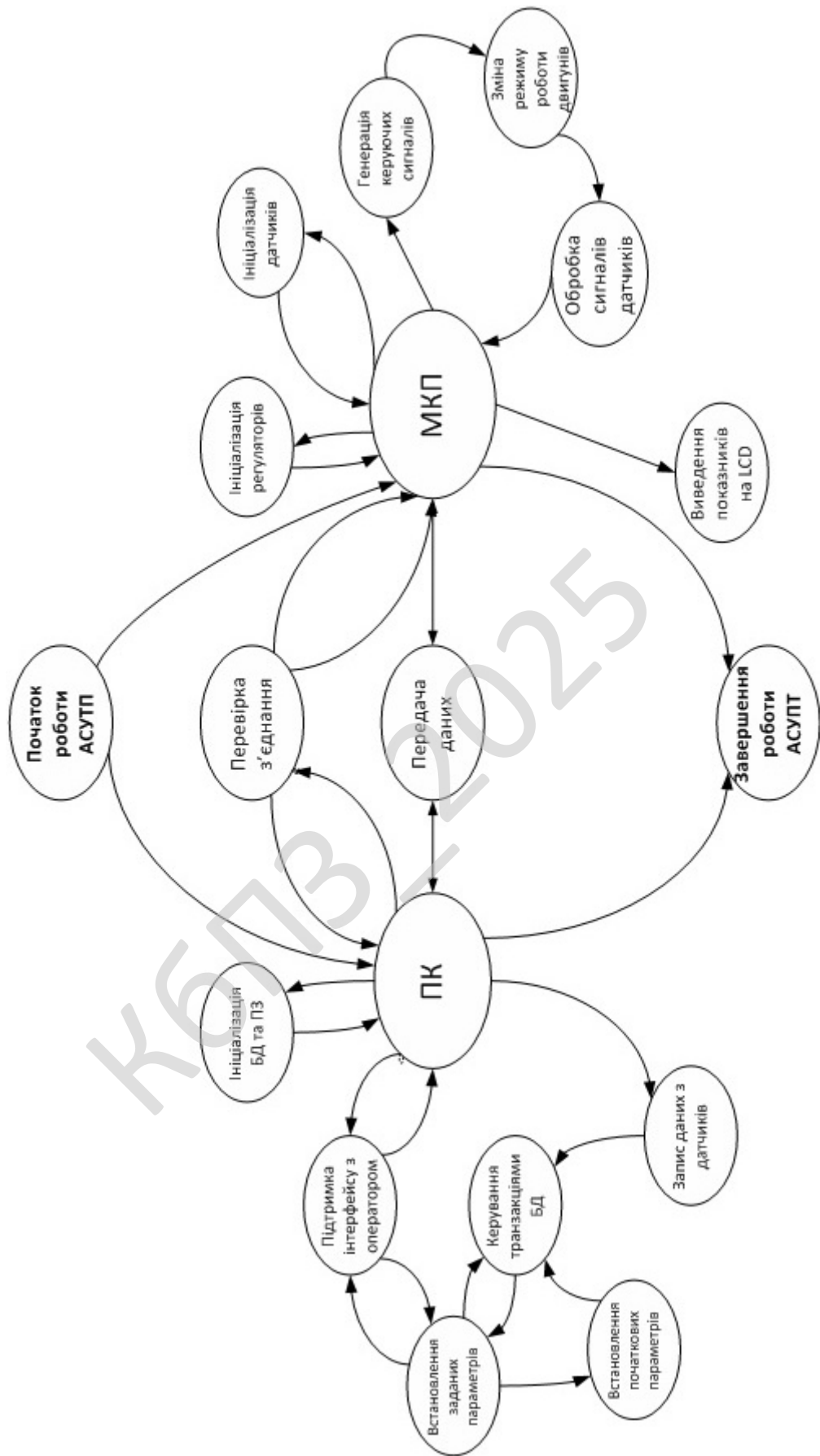


Рисунок 3.7 – Діаграма взаємодії процесів

Таким чином, після детального аналізу системи, а також розгляду її структурної та функціональної схем, а також діаграми взаємодії процесів, наступним кроком є опис блок-схем основної програми та підпрограм, що використовуються для реалізації системи.

Блок-схеми дозволяють графічно відобразити алгоритм роботи програмного забезпечення, включаючи логіку прийняття рішень, умови виконання операцій та послідовність обробки даних. Вони допомагають чітко структурувати програму та оптимізувати її функціонування.

КБПЗ_2025

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

Під час написання даної кваліфікаційної роботи було розроблено програмний продукт, а саме віконний застосунок, призначений для керування процесом водопостачання.

Для реалізації автоматизованої системи керування будемо використовувати одну з найбільш розповсюджених SCADA систем – TRACE MODE (ТРЕЙС МОУД). TRACE MODE призначена для розробки великих розподілених АСКТП широкого призначення. Ця система створена в 1992 році фірмою AdAstra Research Group Ltd і до теперішнього часу має більш ніж 4500 інсталяцій. Системи які розроблені на базі ТРЕЙС МОУД працюють в енергетиці, металургії, нафтовій, газовій, хімічній і іншій галузях промисловості та у комунальному господарстві. По числу впроваджень ТРЕЙС МОУД значно випереджає інші закордонні пакети подібного класу.

ТРЕЙС МОУД - заснована на інноваціях, що не мають аналогів. Серед них: розробка розподіленої АСУТП як єдиного проекту, автопобудова, оригінальні алгоритми обробки сигналів і керування, об'ємна векторна графіка мнемосхем, єдиний мережевий час. ТРЕЙС МОУД - це перша інтегрована SCADA- і softlogic-система, що підтримує наскрізне програмування операторських станцій і контролерів за допомогою єдиного інструменту.

Основними функціями ТРЕЙС МОУД являються наступні:

- модульна структура - від 128 до 64000x16 I/O;
- кількість тегів необмежена;
- мінімальний цикл системи рівний 0.001 с;
- відкритий формат драйвера для зв'язку з будь-яким УСО;
- відкритість для програмування (Visual Basic, Visual C++ і т.д.);

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

- вбудовані бібліотеки з більш ніж 150 алгоритмами обробки даних і керування в т.ч. фільтрація, PID, PDD, нечітке, адаптивне, позиційне регулювання, ШІМ, керування пристроями (клапан, засувка, привод і т.д.), статистичні функції і довільні алгоритми;

- багато інших можливостей.

Суть автопобудови полягає в автоматичній генерації баз каналів операторських станцій і контролерів, що входять у проект АСУТП на основі інформації про число точок введення/виведення, номенклатурі використовуваних контролерів і УСО, наявності і характері зв'язків між ПК і контролерами.

У ТРЕЙС МОУД 5 реалізовані наступні процедури автопобудови:

- автопобудова баз каналів для зв'язку з УСО в РС-контролерах - автоматичне формування баз каналів кожного контролера і його настроювання на УСО на основі інформації про число і марку РС-контролерів, використовуваних у проекті. Технологія автопобудови підтримується в контролерах Micro PC, Круїз, МФК, MIC2000, Advantech PCL і д.р.;

- автопобудова баз каналів для зв'язку зі звичайними контролерами - автоматичне генерування бази каналів операторських станцій і настроювання на найбільш розповсюджені контролери, наприклад Реміконт, Ломіконт, Ш-711, TCM, ЭК-2000, ADAM 4000, ADAM5000, Allen Bradley, Siemens і ін.;

- автопобудова зв'язків між вузлами: "ПК-ПК", "ПК-контролери" - автоматичне створення, підтримка і відновлення комунікацій (наприклад мережевих, RS-232/485, Profibus і т.д.) між вузлами розподіленої АСУТП;

- автопобудова при імпорті баз технологічних параметрів.

При роботі в реальному часі технологія автопобудови відслідковує зміни бази каналів на різних вузлах розподіленої АСКТП (на операторських станціях і в контролерах) і автоматично проводить необхідні зміни. Так наприклад, якщо додати(видалити) датчик, ТРЕЙС МОУД автоматично додасть(видалить) і настроїть канали на усіх вузлах розподіленої АСУ.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

АСКТП реалізовується на контролерах Lagoon, а також комунікаційних модулях, модулях аналогового вводу/виводу, модулях дискретного вводу/виводу серії I-7000.

Базовим процесорним модулем, що використовуються при розробці АСКТП є контролер I-7188 (I-7188 – аналог контролера Lagoon). Власне кажучи I-7188 - це маленький PC- сумісний комп'ютер. У ньому є процесор AMB 188-40МГц, 256 кбайт SRAM пам'яті (ОЗУ), електронний Flash-диск (аналог твердого диска) обсягом 512кбайт, годинник реального часу, 4 послідовних порти, тобто майже всі необхідні атрибути звичайного комп'ютера.

Організація бази каналів

У редакторі бази каналів створюється математична основа системи керування - описуються конфігурації всіх робочих станцій, контролерів і УСО, використовуваних у системі керування, набудовуються інформаційні потоки між ними. Тут же описуються вхідні і вихідні сигнали і їхній зв'язок із пристроями збору даних і керування. У цьому редакторі задаються періоди чи опитування формування сигналів, набудовуються закони первинної обробки і керування, технологічні границі, структура математичної обробки даних. Тут установлюється, які дані, і при яких умовах зберігати в різних архівах, набудовується мережний обмін, описуються задачі керування архівами, документуванням, корекції тимчасових характеристик системи керування, а також зважаються деякі інші задачі.

Для реалізації проекту автоматизації процесу фільтрування створимо 2 вузли:

- ARM – автоматизоване робоче місце;
- Lagoon – вузол керування.

Головне вікно бази каналів має наступний вигляд:

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

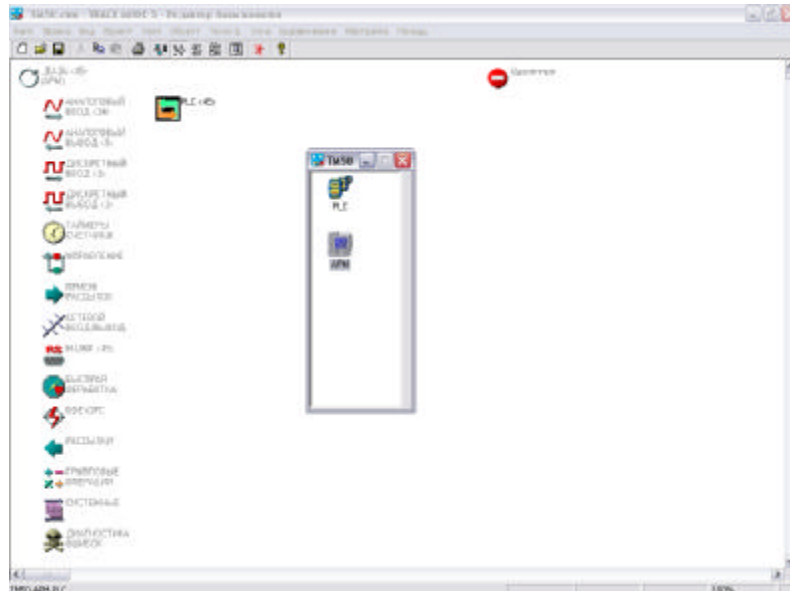


Рисунок 4.1 - Головне вікно бази каналів

Канали, які входять до АWP

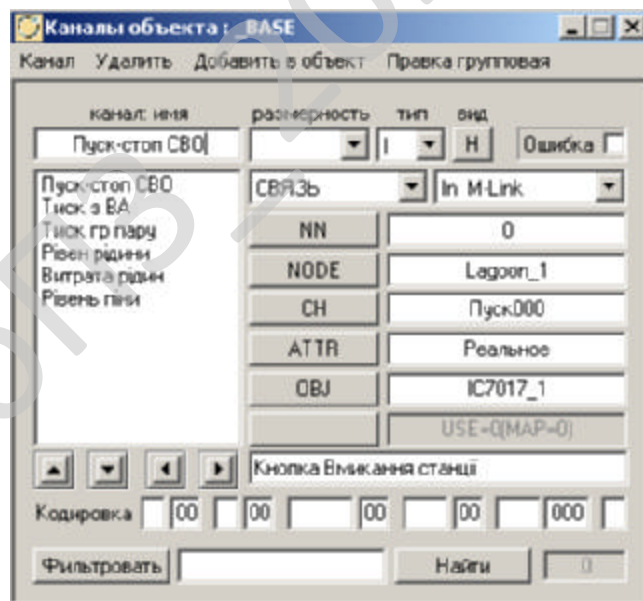


Рисунок 4.2 – Канали АWP

Розробка FBD-програм

Для реалізації програмного забезпечення в базі каналів створюємо FBD програми. У цьому вікні редактора бази каналів здійснюється створення і редагування задач обробки даних і керування, оформлених у виді окремих FBD-

програм мовою Техно FBD. Вхід у це вікно здійснюється або по команді з головного меню, або натисканням ЛК на відповідній іконці панелі інструментів.

Створена FBD програма регулювання рівня

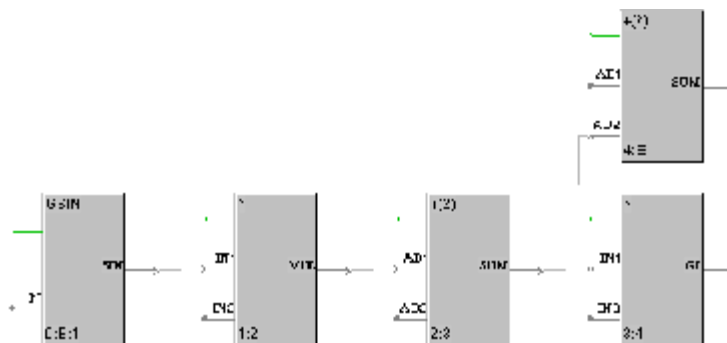


Рисунок 4.3 - FBD програма регулювання рівня

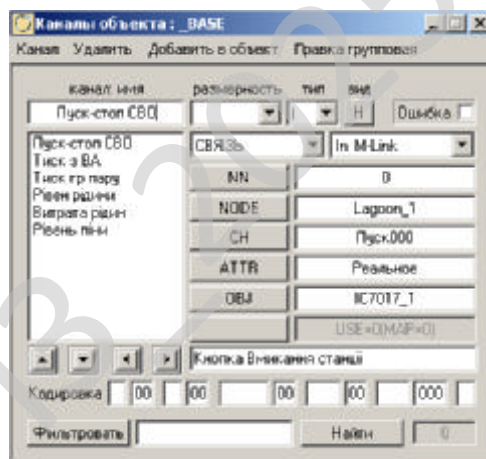


Рисунок 4.4 - Створена FBD програма вимірювання витрати води

Система КАСКАД

КАСКАД включає в себе наступні компоненти.

Серверні модулі:

- сервер доступу до даних здійснює отримання, обробку та накопичення даних, ведення бази даних, аналіз і передачу керуючих сигналів. Накопичення даних ведеться у вигляді SQL - бази даних під керуванням сервера InterBase;
- інтерфейсні модулі доступу до даних здійснюють зв'язок з джерелами даних (мікроконтроллерами тощо);

- конфігуратор СДД надає уніфікований інтерфейс для налаштування модулів доступу до даних (формування набору опитуваних пристроїв, тегів, настройка параметрів опитування).

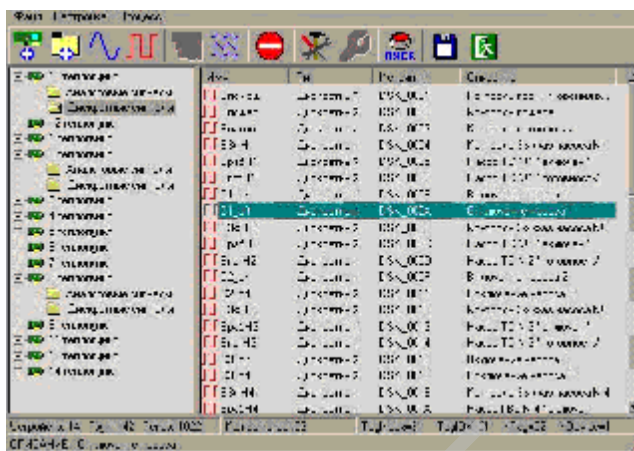


Рисунок 4.5 – Конфігуратор системи «каскад»

Клієнтські модулі

Модуль візуалізації ТП є основним засобом візуального контролю поточних параметрів ТП, а також головним інструментом управління процесами. Відображені дані групуються у вигляді панелей мнемосхем. Кожна панель може відображати інформацію в будь-якому зручному для сприйняття та аналізу вигляді: текстовому, графічному (растрове або векторне зображення), анімовані зображення, відеоролики, тренди, гістограми і т.ін. Причому види відображення можуть комбінуватися в будь-якому поєднанні. Навігація по мнемосхемах максимально проста. Налаштування мнемосхем проводиться у вбудованому редакторі.

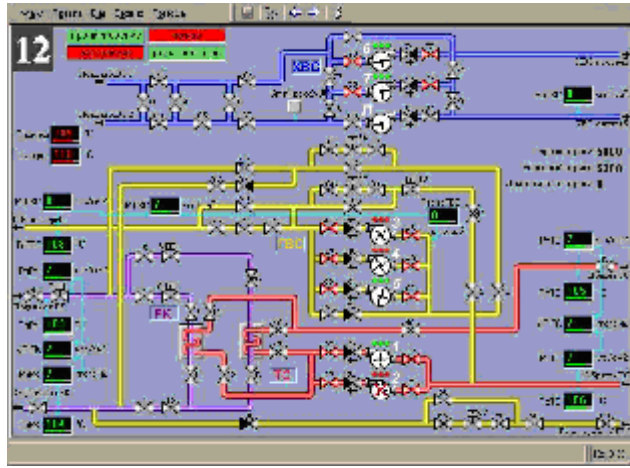


Рисунок 4.6 – Редактор системи «Каскад»

Модуль перегляду історії даних ТП

Являє собою потужний і зручний засіб перегляду історії технологічного процесу, відстеження динаміки ТП завдяки розгортанню даних в графічному вигляді. Інформація може представлятися як у двох, так і в трьох вимірах, в абсолютних одиницях (одиниці виміру), у відсотках. Можливий перегляд як історичних, так і поточних даних (спостерігаючий режим). Дані при відображенні логічно групуються у вигляді панелей передісторії. Кожна панель може працювати як незалежно від інших панелей, так і синхронно з ними. Додавання та видалення графіків проводиться нальоту, як і зміна масштабу відображення. Кількість одночасно відображуваних панелей і графіків на кожній панелі в принципі не обмежена і вибирається з міркувань зручності сприйняття і здорового глузду.

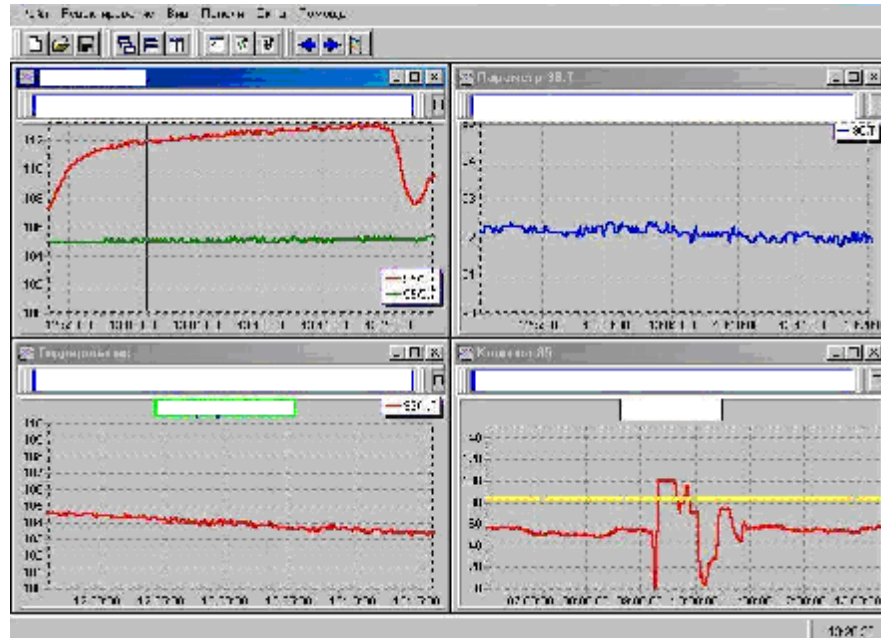


Рисунок 4.7 - Модуль перегляду історії

Модуль формування звітної документації

Дозволяє створювати звіти будь-якого виду за будь-який період часу, вести як змінну, так і наскрізну документацію, а також аналіз даних. Формування звітів здійснюється у форматі і під управлінням Microsoft Excel. По-перше, це дає користувачеві можливість налаштувати вид вихідної документації, використовуючи весь потужний інструментарій, що надається програмою Microsoft Excel, а по-друге, дозволяє використовувати сформовані документи надалі без додаткових перетворень.

Вид документа налаштовується один раз і запам'ятовується у вигляді шаблону. За цим шаблоном у будь-який час може бути сформований вихідний документ на будь-який момент часу.

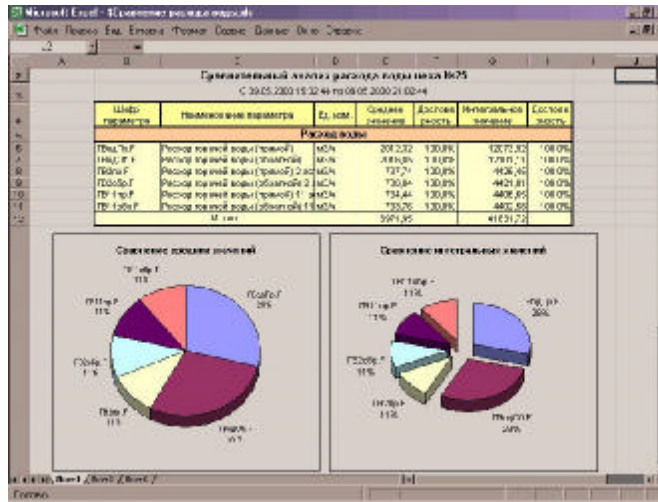


Рисунок 4.8 - Модуль формирования звітної документації

Модуль звукової сигналізації

Здійснює контроль відповідності технологічного процесу встановленим режимам. У разі порушень відбувається інформування користувача програванням звукових файлів. Завдяки надзвичайно гнучким налаштуванням модуль може бути використаний також і для коментування ходу технологічного процесу. В якості звукової інформації можуть бути використані голосові повідомлення; повідомлення можна складати з декількох елементів, зациклювати довільну ділянку ланцюжка. Вузол, що викликав аларм, відображається модулем візуалізації, що дозволяє негайно вжити необхідних заходів. Кожному контрольованому параметру задається пріоритет, що дозволяє в першу чергу обробляти більш важливі сигнали.

Приоритет	Идентификатор	Шифр параметра	Тип паспорта	Значение	Наименование параметра
0	5000	SRBK2осн	дискретный	Вкл.	Сигнализация отказа БК2 основного
0	5001	SRBK2рез	дискретный	Вкл.	Сигнализация отказа БК2 резервного
0	5002	SRBK3	дискретный	Вкл.	Сигнализация отказа БК3

Рисунок 4.9 - Модуль звукової сигналізації

Комплекс апаратного забезпечення АСУ ТП

До складу системи включені:

- 6 насосів, які розбиті на дві групи - основні і додаткові;
- ПП Micromaster 430 фірми Siemens - комутується з одним з основних насосів;
- програмований логічний мікроконтролер DeCont - 182 фірми ДЕК - збирає інформацію з датчиків і управляє технологічним обладнанням та регулює тиск;
- панель PanelView 550 фірми Allen - Bradley - відображає поточні параметри системи, аварійні повідомлення, передісторію подій, звіт по Напрацювання, здійснює введення команд оператора.

Передбачено два режими роботи насосів - штатний і автоматичний.

У штатному режимі управління насосами збережено від існуючих контакторів і кнопок управління. В автоматичному режимі керування роботою насосів і засувки здійснює мікроконтролер

Регулювання тиску води в колекторі в автоматичному режимі здійснюється одним з основних насосів. Залежно від зміни сигналу з аналогового датчика, ЧРП змінює частоту обертання електродвигуна насоса. Після розкрути основного насоса до максимальних обертів і при подальшому зниженні тиску, система через м'який пускач забезпечує плавний, при мінімальних пускових токах і гідродинамічних навантаженнях, пуск додаткового насоса. При цьому, після включення додаткового насоса, точне регулювання тиску здійснюється основним насосом. При збільшенні тиску внаслідок зменшення розбору води система виробляє відключення додаткового насоса і знижує частоту обертання двигуна основного насоса до мінімальних обертів. Уставка тиску в колекторі змінюється автоматично залежно від часу доби. Визначено три види уставок: нічна, денна, вечірня. Крім того, система підтримує різні уставки тиску у вихідні та робочі дні та здійснює плавний перехід з одної уставки на іншу.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

При несправність насосів або засувок, при максимальному або мінімальному тиску на викиді система автоматично зупиняє аварійний насос, запускає резервний і продовжує роботу на справному обладнанні до втручання оператора.

Через панель PanelView оператор може змінювати:

- режим роботи системи - автоматичний / штатний;
- готовність до пуску насосів - готовий / не готовий;
- вибір основних і додаткових насосів;
- уставки тиску води в колекторі.

Модулі системи КАСКАД працюють незалежно один від одного, тому можна, наприклад, одночасно формувати звіт, аналізувати історичні дані і стежити за поточним ходом процесу.

Для розмежування рівнів доступу до інформації введена система користувачів і паролів. Кожному користувачеві визначаються права на запуск додатків, перегляд даних і зміна налаштувань.

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.10 наведено блок-схему основної програми.

Її робота складається з виконання наступних кроків.

Спершу відбувається виведення основного вікна програми. Після цього відбувається перевірка, чи необхідно додати в БД нові пристрої. Якщо необхідно, тоді відбувається додавання в БД нових пристроїв.

Після цього відбувається перевірка, чи є підключені пристрої. Якщо є, тоді відбувається виведення на екран поточних параметрів підключених пристроїв.

Якщо необхідно керувати роботою пристроїв, тоді відбуваються наступні дії:

- налаштовуються параметри зв'язку з пристроями;
- призначаються команди пристроям;

- відбувається налагодження команд;
- відбувається запуск призначених команд.

Якщо треба обробити вимірювання, тоді запускається підпрограма обробки змін параметрів пристроїв.

Якщо необхідно створити звіти, то відбувається створення звітів з використанням архівних даних. Після цього користувач обирає, працювати йому далі з програмою, або ні.

КБПЗ_2025

					VKPB-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

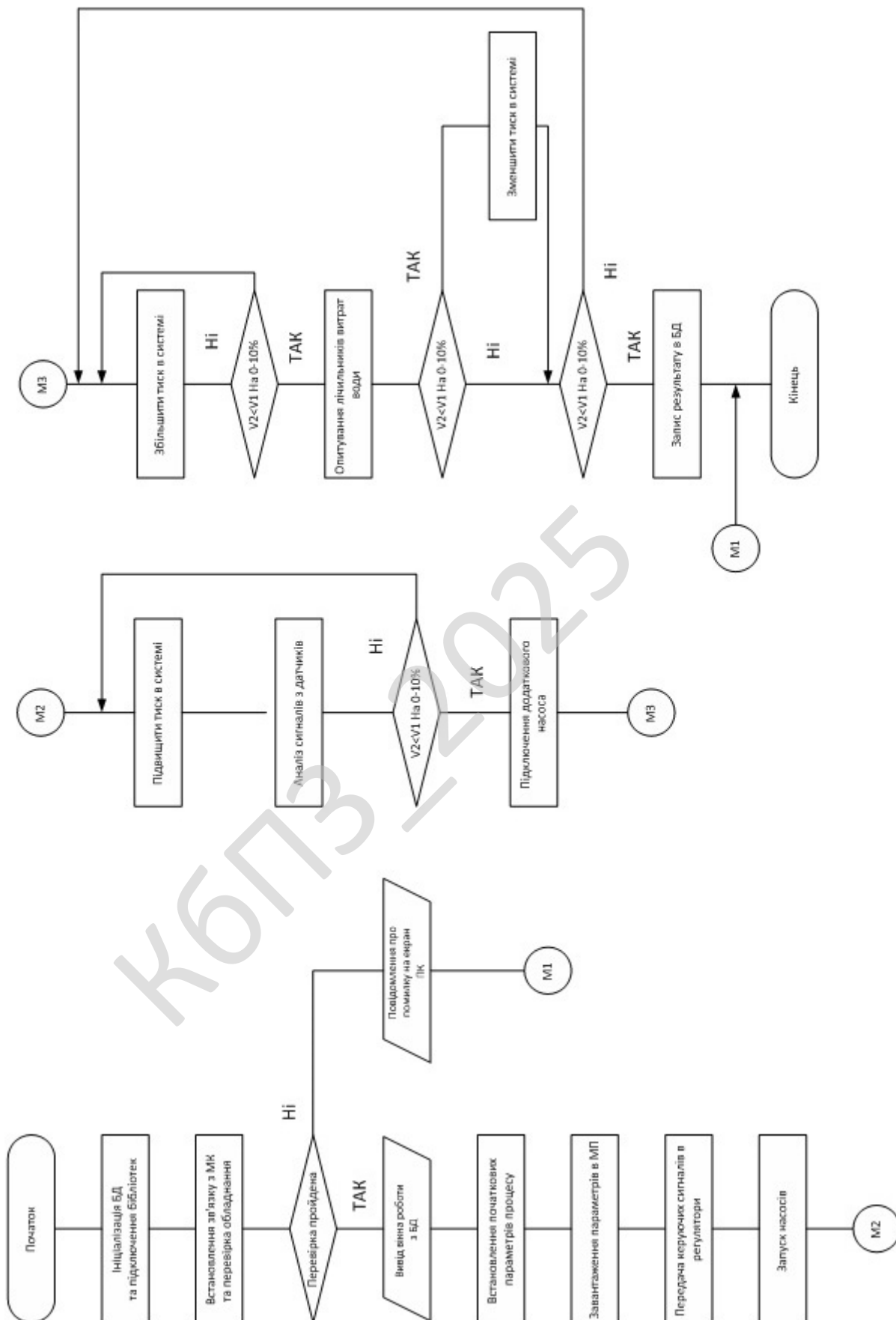


Рисунок 4.10 – Блок-схема основної програми

При розробці ПЗ було використано підходи ризик-менеджменту – це система управління ризиками, яка включає в себе стратегію та тактику управління, направлені на досягнення основних цілей. Ефективний ризик-менеджмент включає:

- систему управління;
- систему ідентифікації і вимірювання;
- систему супроводження (моніторингу та контролю).

Сучасна наука представляє ризик як вірогідну подію, в результаті настання якої можуть відбутися позитивні, нейтральні або негативні наслідки. Якщо ризик припускає наявність як позитивних, так і негативних результатів, він відноситься до спекулятивних ризиків. Якщо ж наслідки негативні, або відсутні взагалі, такий ризик іменується чистим.

КБПЗ – 2025

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

установленої довжини блоку. Далі на відповідних етапах перетворення відбуваються або над незалежними стовпцями, або над незалежними рядками, або взагалі над окремими байтами в таблиці.

Всі перетворення в шифрі мають строге математичне обґрунтування. Сама структура й послідовність операцій дозволяють виконувати даний алгоритм ефективно як на 16-бітних так і на 64-бітних процесорах. У структурі алгоритму закладена можливість паралельного виконання деяких операцій, що на багатопроцесорних робочих станціях може ще підняти швидкість шифрування в 4 рази.

Алгоритм складається з деякої кількості раундів (від 10 до 14 – це залежить від розміру блоку й довжини ключа), у яких послідовно виконуються наступні операції :

ByteSub – Таблична підстановка 8x8 біт (рисунок 4.12).

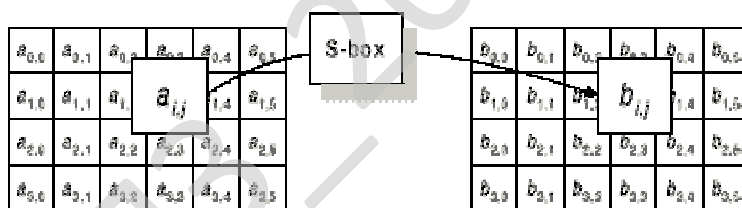


Рисунок 4.12 – Таблична підстановка 8x8 біт

ShiftRow – зрушення рядків у двовимірному масиві на різні зсуви (рисунок 4.13).

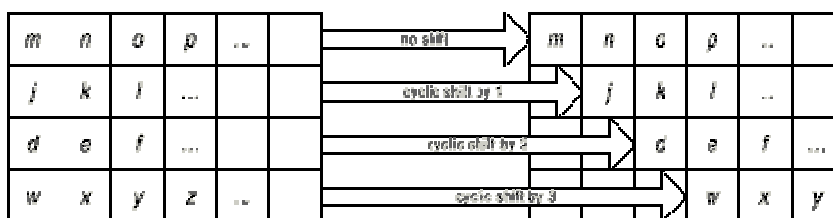


Рисунок 4.13 – Зрушення рядків у двовимірному масиві на різні зсуви

MixColumn – математичне перетворення, що перемішує дані усередині стовпця (рисунок 4.14).

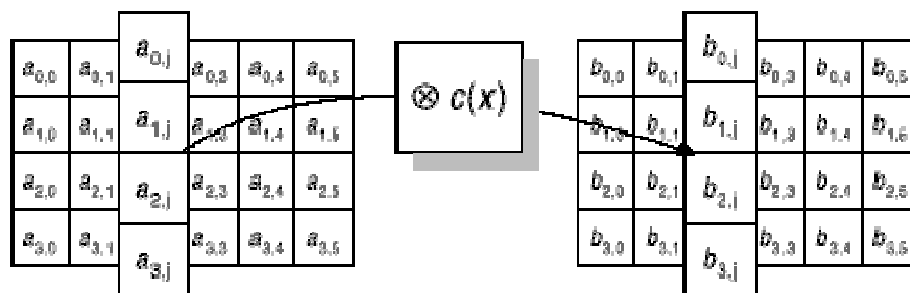


Рисунок 4.14 – Математичне перетворення, що перемішує дані усередині стовпця

AddRoundKey – додавання матеріалу ключа операцією XOR (рисунок 4.15).

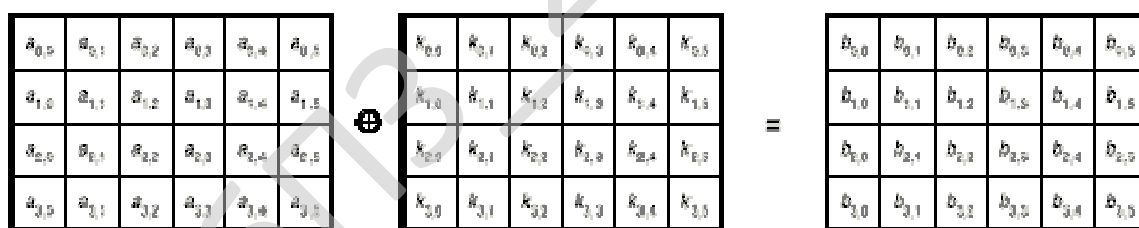


Рисунок 4.15 – Додавання матеріалу ключа операцією XOR

В останньому раунді операція перемішування стовпців відсутня, що робить всю послідовність операцій симетричною.

5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Пробне впровадження системи диспетчеризації водопостачання підприємстві водоканалу смт.Бобринець

03 травня 2025 року впроваджена в промислову експлуатацію система диспетчеризації на станціях водопостачання смт.Бобринець

Система складається з надійних промислових станцій передачі даних з використанням бездротового зв'язку і програмного забезпечення власної розробки для управління обладнанням, а також зберігання, обробки і візуалізації інформації.

В результаті впровадження отримані наступні переваги:

- оперативність збору інформації про стан системи водопостачання;
- можливість дистанційного керування об'єктами водопостачання;
- автоматична підтримка рівня води в резервуарах;
- автоматична підтримка тиску і витрати води в напірному трубопроводі першого і другого підйомів;
- скорочення витрат на технічний огляд віддаленого обладнання;
- зменшення кількості виїздів персоналу на об'єкти;
- дистанційне визначення причин несправностей обладнання на віддалених об'єктах водопостачання;
- оперативна реакція на нештатні ситуації, що в свою чергу покращує якість постачання води для населення та промислових об'єктів.

На рисунку 5.1 представлено вигляд розробленого АРМ.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69



Рисунок 5.1 - Загальний вигляд розробленого АРМ

На рисунку 5.2 представлено розроблену візуальну систему управління а на рисунку 5.3 мнемосхему управління насосною станцією.

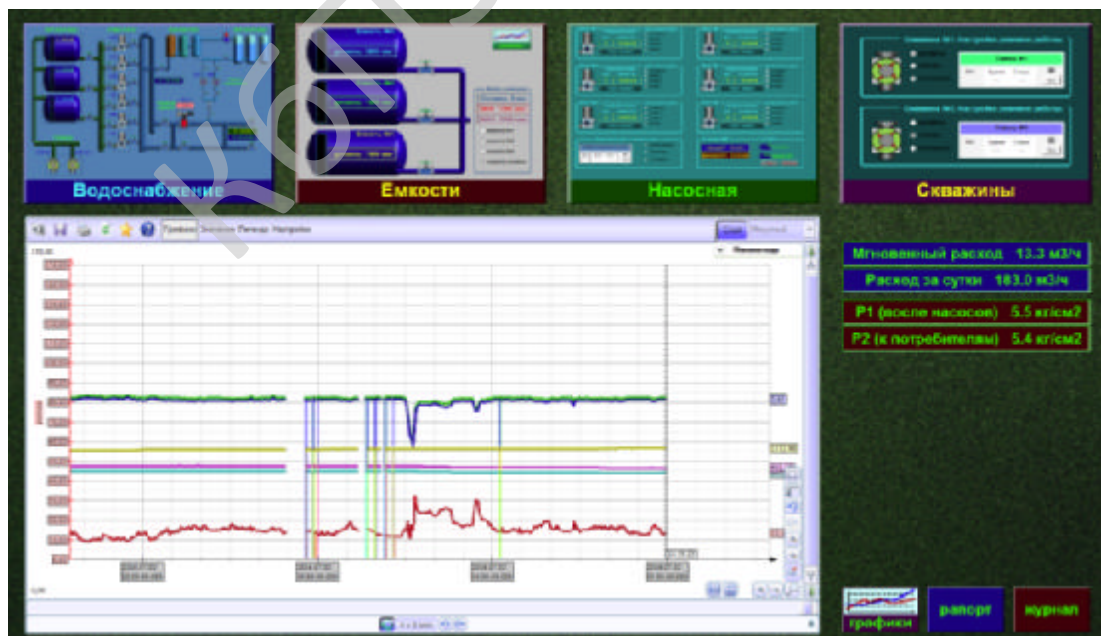


Рисунок 5.2 - Работа SCADA системи

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

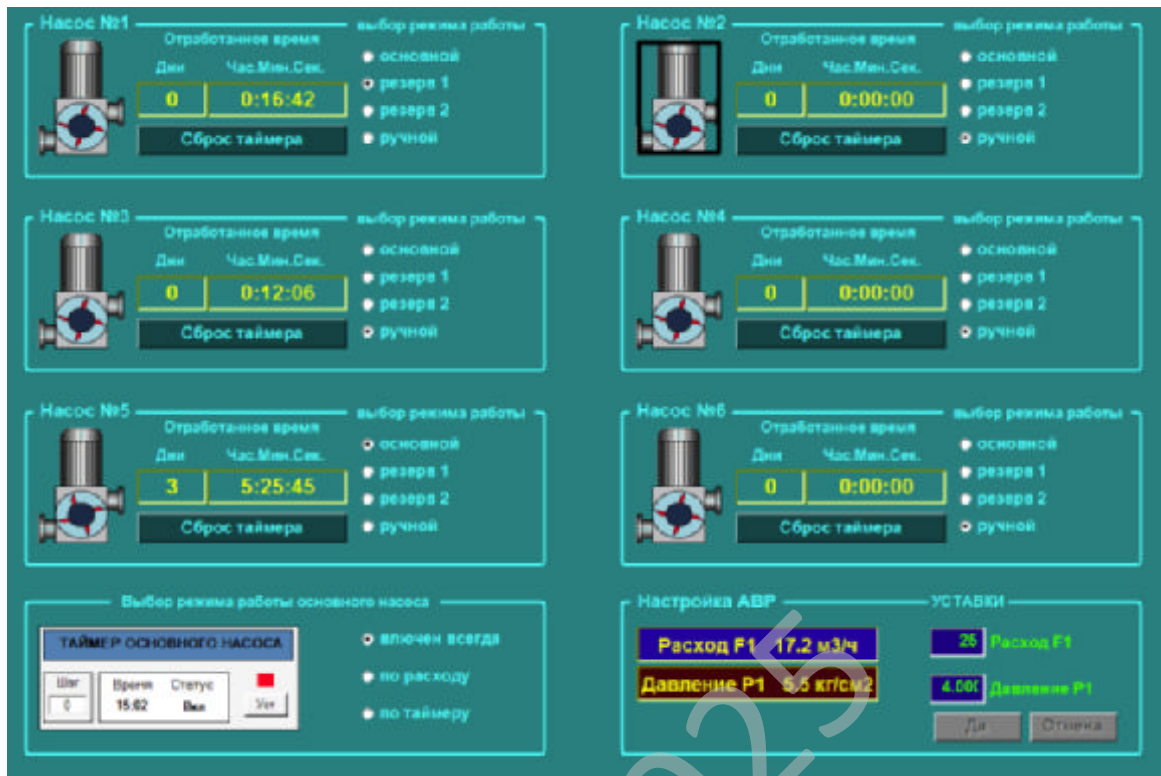


Рисунок 5.3 – Мнемосхема управления насосной станцией

6 ОСНОВНІ ВИСНОВКИ

В даній кваліфікаційній роботі мною було розроблена автоматизована система управління процесом водопостачання смт.Бобринець з використанням мікроконтролерних засобів та розробкою програмного забезпечення для візуального управління процесом водопостачання.

Також розроблено і спроектовано схеми: функціональна, схема зовнішніх з'єднань, принципова електрична схема, схема монтажу електрокомпонентів, також було поставлено і виконано наступні задачі:

– на підставі аналізу існуючих розробок у галузі автоматизованого управління насосними станціями систем водопостачання визначено необхідність розробки нових принципів формування технологічного завдання для НС, прийняття рішення з вибору робочої технологічної схеми НС в виробничих умовах і управління регульованими насосними агрегатами;

– розроблено топологічну і матричні моделі технологічної схеми НС, що дозволяють закласти в програмне забезпечення, створене з орієнтацією на універсальну технологічну схему НС, процедури оптимізації шляхів помпування і раціонального використання технологічного обладнання НС;

– розроблено метод енергозберігаючого управління режимом роботи технологічного обладнання НС, його алгоритмічне та програмне забезпечення, що дозволяє виконати з мінімальними енерговитратами технологічне завдання по тиску і подачі на виході НС в умовах коливань водоспоживання абонентів і може використовуватися операторами НС або безпосередньо в СУ автоматичних НС;

– розроблено метод формування технологічного завдання для НС за узагальненим критерієм ефективності, що максимально враховує втрати абонентів від недотримання заявлених вимог та СВ від перевитрати електроенергії і є одним із шляхів вирішення задачі енергозбереження в процесі управління технологічним обладнанням НС;

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

– розроблена комп'ютерна технологія управління регульованими насосними агрегатами насосної станції, яка забезпечує оптимальний розподіл продуктивності між ними за рахунок координації режимів роботи регульованих насосних агрегатів при коливаннях водоспоживання;

– перспективність практичного застосування розроблених математичних моделей, процедур прийняття рішень і методів автоматизованого енергозберігаючого управління режимом роботи НС, функціональної структури автоматизованої СУ технологічним обладнанням НС підтверджена результатами імітаційного моделювання діючих НС.

КБПЗ – 2025

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Koval V., Adamiv O., Proc. of the Third IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2005). – Sofia (Bulgaria). – 2005. – P. 120- 124.
2. Yasin, J.N., Mohamed, S.A.S., Haghbayan, MH. et al. Low-cost ultrasonic based object detection and collision avoidance method for autonomous robots. Int. j. inf. tecnol. 13, 97–107 (2021). <https://doi.org/10.1007/s41870-020-00513-w>.
3. Sabry F. Self Driving Car: Solving Full Self-driving Need Solving Real-world artificial Intelligence. One Billion Knowledgeable. 2022 . 293 p.
4. Shapiro D. G. Three Anecdotes from the DARPA Autonomous Land Vehicle Project. AI Magazine, 2008. 29(2), 40. <https://doi.org/10.1609/aimag.v29i2.2108>
5. Young R. Critical Analysis of Prototype Autonomous Vehicle Crash Rates Six Scientific Studies from 2015-2018 SAE International. 2021. 252 p
6. Рудик А. В. Наукові основи та принципи побудови приладової системи вимірювання прискорення мобільного робота : дис. ... д-ра техн. наук : 05.11.01– Прилади та методи вимірювання механічних величин. Київ, 2018. – 460 с.
7. Поліщук М. М., Ткач М.М. Робототехнічні системи: проектування і моделювання: навч. посіб. Київ: КПІ ім. Ігоря Сікорського, 2021. 112 с.
8. Мигаль В. Д. Інтелектуальні системи в технічній експлуатації автомобілів: монографія. Х.: Майдан, 2018. 262 с.
9. Staron M. Automotive Software Architectures: An Introduction. Second Edition Springer International Publishing A&G. 2022. 274 p
10. Vargas J. et al. An Overview of Autonomous Vehicles Sensors and Their Vulnerability to Weather Conditions. Sensors. 2021; 21(16):5397.
11. Wevolver 2020 Autonomous Vehicle Technology Report. URL:

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

<https://www.coursehero.com/file/62601304/Wevolver2020AutonomousVehiclesTechnologyReportpdf/> (accessed 2023 June 1)

12. React.js [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.reactjs.org/>

13. AngularJS MVC [Електронний ресурс] – Режим доступу до ресурсу: https://www.tutorialspoint.com/angularjs/angularjs_mvc_architecture.htm

14. Vue.js [Електронний ресурс] – Режим доступу до ресурсу: <https://vuejs.org/>

15. A Look Inside ADAS Modules. URL: <https://amkor.com/semiconductor-story/alook>.

16. AV and ADAS Sensors. URL <https://community.sw.siemens.com/s/article/AVand-ADAS-Sensors> (accessed 2023 June 1)

17. Гуржій А. М. Основи автоматики та робототехніки: Навчальний посібник/ А.М. Гуржій, А. Т. Нельга, В. М. Співак, О. С. Ітякін:—Дніпро:«Гарант СВ», 2021.- 243с.

18. Lim, B.S. et al. Autonomous Vehicle Ultrasonic Sensor Vulnerability and Impact Assessment. In Proceedings of the IEEE World Forum on Internet of Things (WFIoT), Singapore, 5–8 February 2018; pp. 231–236.

19. Саліхов М. М. Arduino – перспективний інструмент протитопування у робототехніці // Концептуальні шляхи розвитку науки та освіти: матеріали VIII Міжнародної науково-практичної конференції м. Львів, 9-10 червня 2023 року.— Львів: Львівський науковий форум, 2023. – с. 73-79.

20. Bräunl T. Embedded Robotics: From Mobile Robots to Autonomous Vehicles with RaspberryPi and Arduino Springer Nature, 2022. 519 p.

21. Навігаційні системи [Електронний ресурс] : навч. посіб. для студ. Спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / С.Л. Лакоза; КПІ ім. Ігоря Сікорського, 2021. — 80 с.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

22. Imad M. et al. Navigation system for autonomous vehicle: A survey. JCSTS., vol. 2, no. 2, pp. 20–35, 2020.

23. Ковалець І. В. та ін. Технологія планування траєкторій руху мобільних об'єктів з урахуванням перешкод на складній місцевості. Енергетика і автоматика. - 2017. - № 1. - С. 110-122.

24. HTML [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/uk/docs/Web/HTML>

25. Aguilo I. et al. Artificial Intelligence Research and Development Hardcover by IOS Press. 2003. 500 p.

26. Гребенюк Б. А. Розробка підсистеми управління інтелектуальним роботом / Б. А. Гребенюк // «Automation and Development of Electronic Devices» ADED-2023: Collection of Students' Scientific Paper. – Kharkiv : Kind of Kharkiv National University of Radio Electronics [electronic edition], 2023. – Part 1. –336p. P. 263-269

27. MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/MongoDB>

28. Becker M. et al Obstacle avoidance procedure for mobile robots. ABCM Symposium series in Mechatronics. Vol. 2, 2006. P. 250-257.

29. CORS, XSS [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.to/maleta/cors-xss-and-csrf-with-examples-in-10-minutes-35k3>

30. AJAX [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/AJAX>

31. Fetch API [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/ru/docs/Web/API/Fetch_API

32. Звенігородський О.С. Інтелектуальна система планування тактики руху автономного робота в квазістаціонарному середовищі: Дис... канд. техн. наук: 05.13.23. — Д., 2002. — 127с.

33. Адамів, О.П. Моделі та інтелектуальні засоби адаптивного керування автономним мобільним роботом. Дис. канд.техн. наук. Одеса, 2007. 124 с.

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

34. Koval V., Adamiv O., Proc. of the Third IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2005). – Sofia (Bulgaria). – 2005. – P. 120- 124.

35. npm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.npmjs.com/>

36. Install MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.mongodb.com/manual/administration/install-on-linux/>

37. Как установить Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://www.digitalocean.com/community/tutorials/node-js-ubuntu-18-04-ru>

38. Linux [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Linux>

39. npm-audit [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.npmjs.com/cli/audit>

40. Kojima T. et al. Intelligent Technology for More Advanced Autonomous Driving. 2018. Hitachi Review Vol. 67, No. 1. p. 58–63

41. SQL [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/SQL>

42. MongoDB Compass [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/products/compass>

43. Postman [Електронний ресурс] – Режим доступу до ресурсу: <https://www.postman.com/>

44. Саліхов М. М. Від автоматизованих до автономних транспортних засобів // Наукові досягнення та інновації: шлях до успіху. X Всеукраїнська мультидисциплінарна науково-практична Інтернет-конференція, 31 травня 2023, Україна, Київ : зб. матеріалів — Електрон. дан. — Київ : Ярочё нко Я. В., 2023. — с. 136-142.

45. SPA (Single-page application) [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Glossary/SPA>

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

46. Саліхов М.М. Самокеровані автомобілі та системи їх навігації// Практичні та теоретичні питання розвитку науки та освіти: матеріали VIII Міжнародної науково-практичної конференції: м. Львів, 19-20 червня 2023 року. – Львів: Львівський науковий форум, 2023. – С.53-59.

47. Мигаль В. Д. Інтелектуальні системи в технічній експлуатації автомобілів: монографія. Х.: Майдан, 2018. 262 с.

48. Vargas J. et al. An Overview of Autonomous Vehicles Sensors and Their

49. Vulnerability to Weather Conditions. Sensors. 2021; 21(16):5397.

50. <https://www.coursehero.com/file/62601304/Wevolver2020AutonomousVehi>
cech nologyReportpdf/ (accessed 2023 June 1).

КБПЗ_2025

					ВКРБ-123.25.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Перелік документів, що розробляються.....	5
8 Етапи розробки.....	6
9 Порядок контролю та приймання.....	6

					ВКРБ-123.25.0032.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Іванченко О.В.				Програмне забезпечення мікропроцесорної системи управління насосною станцією водозабору смт. Бобринець		
Перевірів	Кислун О.В.						
Н. Контр.	Коваленко А.С.				Літ.	Аркуш	Аркушів
Затв.	Смірнов О.А.				Б	1	6
					ЦНТУ КІ-21-2		

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку програмного забезпечення системи керування водозабором насосної станції смт.Бобринець

2 Підстава для розробки

Підставою для розробки служить завдання на випуск кваліфікаційну роботу, видане на кафедрі кібербезпеки та програмного забезпечення (нак. №47-02 від 17.01.2025 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної бакалаврської роботи є розробка програмного забезпечення системи керування водозабору насосної станції смт.Бобринець.

4 Джерела розробки

Джерелом цієї кваліфікаційної бакалаврської дипломної роботи є відносна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРБ-123.25.0032.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- роботу системи керування процесом водопостачання;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРБ-123.25.0032.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows XP/Vista/7/8/10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows XP/Vista/7/8/10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

При розробці ПЗ потрібно використовувати наступні технології та мови програмування: HTML, JavaScript, бібліотека jQuery, PHP + PDO, AJAX. Для програмування мікроконтролера мова низького рівня C#. Середовище програмування – PHP-Shtorm.

					ВКРБ-123.25.0032.00.00.ТЗ	Арк.
						4
Вим.	Арк.	№ документа	Підпис	Дата		

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Перелік документів, що розробляються

- Структурна схема системи керування – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма взаємодії процесів – 1 аркуш.
- Блок-схема алгоритму основної програми регулювання тиску в системі – 1 аркуш.
- Блок-схема роботи підпрограми керування – 1 аркуш.
- Пояснювальна записка – 78 аркушів.

8 Етапи розробки

8.1 Збір і обробка інформації по темі кваліфікаційної бакалаврської роботи.

					ВКРБ-123.25.0032.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

Постановка задачі на виконання кваліфікаційної роботи (складання ТЗ).

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень кваліфікаційної роботи.

8.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

9 Порядок контролю та приймання

9.1 Подання кваліфікаційної роботи на попередній захист 21.05.2025 р.

9.2 Подання кваліфікаційної роботи на захист 10.06.2025 р.

					ВКРБ-123.25.0032.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ
Керівник випускної кваліфікаційної
бакалаврської роботи
_____ Кислун О.А

Програмне забезпечення мікропроцесорної системи управління насосною станцією водозабору смт. Бобринець

Лістинг програми

Код документу 12
Носій: CD/DVD-диск

Загальна кількість аркушів: 13

Літера: РП

Файл Main.exe – програмування мікроконтролера для системи управління

R7. EQU 7; адреса регистров R0-R7
 R6. EQU 6
 R5. EQU 5
 R4. EQU 4
 R3. EQU 3
 R2. EQU 2
 R1. EQU 1
 R0. EQU 0
 ACC. EQU0E0H; адрес аккумулятора
 B. EQU0F0H; адрес регистра B
 PSW. EQU000H; адрес регистра (слова) состояния
 SP. EQU81H; адрес указателя стека
 DPL. EQU82H; адрес младшей половины DPTR
 DPH. EQU83H; адрес старшей половины DPTR
 P0. EQU80H; адрес регистра порта P0
 P1. EQU90H; адрес регистра порта P1
 P2. EQU0A0H; адрес регистра порта P2
 P3. EQU0B0H; адрес регистра порта P3
 B.0. EQU0F0H; адреса отдельных битов регистра B
 B.1. EQU0F1H
 B.2. EQU0F2H
 B.3. EQU0F3H
 B.4. EQU0F4H
 B.5. EQU0F5H
 B.6. EQU0F6H
 B.7. EQU0F7H
 ACC.0. EQU0E0H; адреса отдельных битов аккумулятора
 ACC.1. EQU0E1H
 ACC.2. EQU0E2H
 ACC.3. EQU0E3H
 ACC.4. EQU0E4H
 ACC.5. EQU0E5H
 ACC.6. EQU0E6H
 ACC.7. EQU0E7H
 PSW.0. EQU0D0H; адреса отдельных битов регистра PSW
 PSW.1. EQU0D1H
 PSW.2. EQU0D2H
 PSW.3. EQU0D3H
 PSW.4. EQU0D4H
 PSW.5. EQU0D5H
 PSW.6. EQU0D6H
 PSW.7. EQU0D7H
 P0.0. EQU080H; адреса отдельных линий порта P0
 P0.1. EQU081H
 P0.2. EQU082H
 P0.3. EQU083H
 P0.4. EQU084H
 P0.5. EQU085H
 P0.6. EQU086H
 P0.7. EQU087H
 P1.0. EQU090H; адреса отдельных линий порта P1
 P1.1 EQU091H
 P1.2 EQU092H
 P1.3 EQU093H
 P1.4 EQU094H
 P1.5 EQU095H
 P1.6. EQU096H
 P1.7 EQU097H
 P2.0. EQU0A0H; адреса отдельных линий порта P2
 P2.1 EQU0A1H
 P2.2 EQU0A2H
 P2.3 EQU0A3H

```

P2.4 EQU0A4H
P2.5 EQU0A5H
P2.6 EQU0A6H
P2.7 EQU0A7H
P3.0. EQU0B0H; адреса отдельных линий порта P3
P3.1 EQU0B1H
P3.2 EQU0B2H
P3.3 EQU0B3H
P3.4 EQU0B4H
P3.5 EQU0B5H
P3.6 EQU0B6H
P3.7 EQU0B7H
;
CS. EQU P3.7
DCLOCK. EQU P3.6
DOUT. EQU P3.5
;
.org0; нижеследующая команда с адреса 0
;
LJMPSTART; на команду после метки START
;
.org100H; нижеследующая команда с адреса 100H
;
START:
MOV P0, #11111111B; начальная установка
MOV P1, #11111111B
MOV P2, #11111111B
MOV P3, #11111111B
CLRDCLOCK; установка DCLOCK в 0
;
L7816; собственно чтение
;
CLRCS; импульс старта преобразования
;
SETBDCLOCK; 1-й тактовый импульс
CLRDCLOCK
;
SETBDCLOCK; 2-й тактовый импульс
CLRDCLOCK
;
SETBDCLOCK; 3-й тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV B.3. C; DB11 В В.3
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV B.2. C; DB10 В В.2
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV B.1. C; DB9 В В.1
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV B.0. C; DB8 В В.0
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV ACC.7. C; DB7 В ACC.7
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT

```

```

MOV ACC.6. C; DB6 В ACC.6
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV ACC.5. C; DB5 В ACC.5
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV ACC.4. C; DB4 В ACC.4
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV ACC.3. C; DB3 В ACC.3
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV ACC.2. C; DB2 В ACC.2
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV ACC.1. C; DB1 В ACC.1
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV ACC.0. C; DB0 В ACC.0
;
SETBCS; завершение считывания
;
MOV R4, A; сохраняем мл. и ср. тетрады в R4
;
MOVA, B; читаем из регистра в ст. тетраду
ANL A, #00001111B; зануляем старшие 4 бита
MOV R5, A; в R5R4 - результат
;
SJMP L7816; зацикливание
;
. END

```

Файл MainForm.cs - головне вікно програми

```

namespace RecoveryDisk
{
    partial class MainForm
    {
        /// <summary>
        ///
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        ///
        /// </summary>
        /// <param name="disposing">правда, якщо керуючі ресурси повині бути
        розташовані, неправда в іншому випадку.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
        }
    }
}

```

```

        base.Dispose(disposing);
    }

    #region Windows Form Designer generated code

    /// <summary>
    /// Викликаємо метод для підтримки інтерфейсу
    /// </summary>
    private void InitializeComponent()
    {
        this.components = new System.ComponentModel.Container();
        System.Windows.Forms.TreeNode treeNode1 = new
System.Windows.Forms.TreeNode("");
        System.Windows.Forms.TreeNode treeNode2 = new
System.Windows.Forms.TreeNode("Documents and Settings", 18, 20, new
System.Windows.Forms.TreeNode[] {
            treeNode1});
        System.Windows.Forms.TreeNode treeNode3 = new
System.Windows.Forms.TreeNode("");
        System.Windows.Forms.TreeNode treeNode4 = new
System.Windows.Forms.TreeNode("Downloads", 18, 20, new
System.Windows.Forms.TreeNode[] {
            treeNode3});
        System.Windows.Forms.TreeNode treeNode5 = new
System.Windows.Forms.TreeNode("");
        System.Windows.Forms.TreeNode treeNode6 = new
System.Windows.Forms.TreeNode("Program Files", 18, 20, new
System.Windows.Forms.TreeNode[] {
            treeNode5});
        System.Windows.Forms.TreeNode treeNode7 = new
System.Windows.Forms.TreeNode("");
        System.Windows.Forms.TreeNode treeNode8 = new
System.Windows.Forms.TreeNode("RapidDriver", 18, 20, new
System.Windows.Forms.TreeNode[] {
            treeNode7});
        System.Windows.Forms.TreeNode treeNode9 = new
System.Windows.Forms.TreeNode("");
        System.Windows.Forms.TreeNode treeNode10 = new
System.Windows.Forms.TreeNode("Server", 18, 20, new
System.Windows.Forms.TreeNode[] {
            treeNode9});
        System.Windows.Forms.TreeNode treeNode11 = new
System.Windows.Forms.TreeNode("");
        System.Windows.Forms.TreeNode treeNode12 = new
System.Windows.Forms.TreeNode("WINDOWS", 18, 20, new
System.Windows.Forms.TreeNode[] {
            treeNode11});
        System.Windows.Forms.TreeNode treeNode13 = new
System.Windows.Forms.TreeNode("");
        System.Windows.Forms.TreeNode treeNode14 = new
System.Windows.Forms.TreeNode("WINDOWS.0", 18, 20, new
System.Windows.Forms.TreeNode[] {
            treeNode13});
        System.Windows.Forms.TreeNode treeNode15 = new
System.Windows.Forms.TreeNode("SYSTEM2 (C:)", 23, 24, new
System.Windows.Forms.TreeNode[] {
            treeNode2,
            treeNode4,
            treeNode6,
            treeNode8,
            treeNode10,
            treeNode12,
            treeNode14});
        System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(MainForm));
        this.menuStrip = new System.Windows.Forms.MenuStrip();
        this.fileToolStripMenuItem = new
System.Windows.Forms.ToolStripMenuItem();
    }

```

```

        this.instrumentToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.adjustmentToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.encodingfilterToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.quickextractionToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.helpToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.separatorToolStripMenuItem = new
System.Windows.Forms.ToolStripItemSeparator();
        this.aboutToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.coderConfigGroupBox = new System.Windows.Forms.GroupBox();
        this.redundancyGroupBox = new System.Windows.Forms.GroupBox();
        this.redundancyMacTrackBar = new
EConTech.Windows.MACUI.MACTrackBar();
        this.allVolCountGroupBox = new System.Windows.Forms.GroupBox();
        this.allVolCountMacTrackBar = new
EConTech.Windows.MACUI.MACTrackBar();
        this.toolTip = new System.Windows.Forms.ToolTip(this.components);
        this.browser = new FileBrowser.Browser();
        this.repairButton = new System.Windows.Forms.Button();
        this.testButton = new System.Windows.Forms.Button();
        this.recoverButton = new System.Windows.Forms.Button();
        this.protectButton = new System.Windows.Forms.Button();
        this.exitToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.testspeedToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.menuStrip.SuspendLayout();
        this.coderConfigGroupBox.SuspendLayout();
        this.redundancyGroupBox.SuspendLayout();
        this.allVolCountGroupBox.SuspendLayout();
        this.SuspendLayout();
        //
        // menuStrip
        //
        this.menuStrip.Items.AddRange(new
System.Windows.Forms.ToolStripItem[] {
        this.fileToolStripMenuItem,
        this.instrumentToolStripMenuItem,
        this.adjustmentToolStripMenuItem,
        this.helpToolStripMenuItem});
        this.menuStrip.LayoutStyle =
System.Windows.Forms.ToolStripLayoutStyle.Flow;
        this.menuStrip.Location = new System.Drawing.Point(0, 0);
        this.menuStrip.Name = "menuStrip";
        this.menuStrip.Size = new System.Drawing.Size(986, 21);
        this.menuStrip.TabIndex = 0;
        this.menuStrip.Text = "menuStrip";
        //
        // fileToolStripMenuItem
        //
        this.fileToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
        this.exitToolStripMenuItem});
        this.fileToolStripMenuItem.Name = "файлToolStripMenuItem";
        this.fileToolStripMenuItem.Size = new System.Drawing.Size(45, 17);
        this.fileToolStripMenuItem.Text = "Файл";
        //
        // instrumentsToolStripMenuItem
        //
        this.instrumentToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
        this.testspeedToolStripMenuItem});
        this.instrumentToolStripMenuItem.Name =
"instrumentsToolStripMenuItem";

```

```

17);
        this.instrumentToolStripMenuItem.Size = new System.Drawing.Size(82,
        this.instrumentToolStripMenuItem.Text = "Інструменти";
        //
        // adjustmentToolStripMenuItem
        //
        this.adjustmentToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
        this.encodingfilterToolStripMenuItem,
        this.quickextractionToolStripMenuItem});
        this.adjustmentToolStripMenuItem.Name =
"adjustmentToolStripMenuItem";
        this.adjustmentToolStripMenuItem.Size = new System.Drawing.Size(74,
17);
        this.adjustmentToolStripMenuItem.Text = "Параметри";
        //
        // encodingfilterToolStripMenuItem
        //
        this.encodingfilterToolStripMenuItem.ImageScaling =
System.Windows.Forms.ToolStripItemImageScaling.None;
        this.encodingfilterToolStripMenuItem.Name =
"encodingfilterToolStripMenuItem";
        this.encodingfilterToolStripMenuItem.Size = new
System.Drawing.Size(216, 22);
        this.encodingfilterToolStripMenuItem.Text = "Шифруючий фільтр";
        this.encodingfilterToolStripMenuItem.Click += new
System.EventHandler(this.encodingfilterToolStripMenuItem_Click);
        //
        // helpToolStripMenuItem
        //
        this.helpToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
        this.separatorToolStripMenuItem,
        this.aboutToolStripMenuItem});
        this.helpToolStripMenuItem.Name = "helpToolStripMenuItem";
        this.helpToolStripMenuItem.Size = new System.Drawing.Size(60, 17);
        this.helpToolStripMenuItem.Text = "Довідка";
        //
        // separatorToolStripMenuItem
        //
        this.separatorToolStripMenuItem.Name = "separatorToolStripMenuItem";
        this.separatorToolStripMenuItem.Size = new System.Drawing.Size(163,
6);
        //
        // aboutToolStripMenuItem
        //
        this.aboutToolStripMenuItem.Name = "aboutToolStripMenuItem";
        this.aboutToolStripMenuItem.Size = new System.Drawing.Size(166, 22);
        this.aboutToolStripMenuItem.Text = "Про програму...";
        this.aboutToolStripMenuItem.Click += new
System.EventHandler(this.aboutToolStripMenuItem_Click);
        //
        // coderConfigGroupBox
        //
        this.coderConfigGroupBox.Controls.Add(this.redundancyGroupBox);
        this.coderConfigGroupBox.Controls.Add(this.allVolCountGroupBox);
        this.coderConfigGroupBox.FlatStyle =
System.Windows.Forms.FlatStyle.Flat;
        this.coderConfigGroupBox.Location = new System.Drawing.Point(414,
26);
        this.coderConfigGroupBox.Name = "coderConfigGroupBox";
        this.coderConfigGroupBox.Size = new System.Drawing.Size(561, 98);
        this.coderConfigGroupBox.TabIndex = 5;
        this.coderConfigGroupBox.TabStop = false;
        this.coderConfigGroupBox.Text = "Конфігурація системи";
        //
        // redundancyGroupBox
        //
        this.redundancyGroupBox.Controls.Add(this.redundancyMacTrackBar);

```

```

        this.redundancyGroupBox.FlatStyle =
System.Windows.Forms.FlatStyle.Flat;
        this.redundancyGroupBox.Location = new System.Drawing.Point(286,
21);
        this.redundancyGroupBox.Name = "redundancyGroupBox";
        this.redundancyGroupBox.Size = new System.Drawing.Size(264, 65);
        this.redundancyGroupBox.TabIndex = 4;
        this.redundancyGroupBox.TabStop = false;
        this.redundancyGroupBox.Text = "Надлишковість кодування";
        //
        // allVolCountGroupBox
        //
        this.allVolCountGroupBox.Controls.Add(this.allVolCountMacTrackBar);
        this.allVolCountGroupBox.FlatStyle =
System.Windows.Forms.FlatStyle.Flat;
        this.allVolCountGroupBox.Location = new System.Drawing.Point(12,
21);
        this.allVolCountGroupBox.Name = "allVolCountGroupBox";
        this.allVolCountGroupBox.Size = new System.Drawing.Size(264, 65);
        this.allVolCountGroupBox.TabIndex = 3;
        this.allVolCountGroupBox.TabStop = false;
        this.allVolCountGroupBox.Text = "Кількість дисків";
        //
        // toolTip
        //
        this.toolTip.AutoSize = false;
        this.toolTip.AutoPopDelay = 20000;
        this.toolTip.InitialDelay = 2000;
        this.toolTip.ReshowDelay = 1000;
        //
        // redundancyMacTrackBar
        //
        this.redundancyMacTrackBar.BackColor =
System.Drawing.Color.Transparent;
        this.redundancyMacTrackBar.BorderColor =
System.Drawing.SystemColors.ActiveBorder;
        this.redundancyMacTrackBar.Font = new System.Drawing.Font("Verdana",
8.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte) 0));
        this.redundancyMacTrackBar.ForeColor =
System.Drawing.Color.FromArgb(((int) ((byte) 123))), ((int) ((byte) 125))),
((int) ((byte) 123))));
        this.redundancyMacTrackBar.IndentHeight = 6;
        this.redundancyMacTrackBar.Location = new System.Drawing.Point(6,
24);
        this.redundancyMacTrackBar.Maximum = 199;
        this.redundancyMacTrackBar.Minimum = 0;
        this.redundancyMacTrackBar.Name = "redundancyMacTrackBar";
        this.redundancyMacTrackBar.Size = new System.Drawing.Size(252, 28);
        this.redundancyMacTrackBar.TabIndex = 6;
        this.redundancyMacTrackBar.TextTickStyle =
System.Windows.Forms.TickStyle.None;
        this.redundancyMacTrackBar.TickColor =
System.Drawing.Color.FromArgb(((int) ((byte) 148))), ((int) ((byte) 146))),
((int) ((byte) 148))));
        this.redundancyMacTrackBar.TickHeight = 4;
        this.redundancyMacTrackBar.TickStyle =
System.Windows.Forms.TickStyle.None;
        this.redundancyMacTrackBar.TrackerColor =
System.Drawing.Color.FromArgb(((int) ((byte) 24))), ((int) ((byte) 130))),
((int) ((byte) 198))));
        this.redundancyMacTrackBar.TrackerSize = new System.Drawing.Size(10,
16);
        this.redundancyMacTrackBar.TrackLineColor =
System.Drawing.Color.FromArgb(((int) ((byte) 90))), ((int) ((byte) 93))),
((int) ((byte) 90))));
        this.redundancyMacTrackBar.TrackLineHeight = 3;
        this.redundancyMacTrackBar.Value = 19;

```

```

        this.redundancyMacTrackBar.ValueChanged += new
EConTech.Windows.MACUI.ValueChangedHandler(this.redundancyMacTrackBar_ValueChang
ed);
        this.redundancyMacTrackBar.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.redundancyMacTrackBar_MouseUp);
        //
        // allVolCountMacTrackBar
        //
        this.allVolCountMacTrackBar.BackColor =
System.Drawing.Color.Transparent;
        this.allVolCountMacTrackBar.BorderColor =
System.Drawing.SystemColors.ActiveBorder;
        this.allVolCountMacTrackBar.Font = new
System.Drawing.Font("Verdana", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)0));
        this.allVolCountMacTrackBar.ForeColor =
System.Drawing.Color.FromArgb(((int)((byte)(123))), ((int)((byte)(125))),
((int)((byte)(123))));
        this.allVolCountMacTrackBar.IndentHeight = 6;
        this.allVolCountMacTrackBar.Location = new System.Drawing.Point(6,
24);
        this.allVolCountMacTrackBar.Maximum = 15;
        this.allVolCountMacTrackBar.Minimum = 0;
        this.allVolCountMacTrackBar.Name = "allVolCountMacTrackBar";
        this.allVolCountMacTrackBar.Size = new System.Drawing.Size(252, 28);
        this.allVolCountMacTrackBar.TabIndex = 5;
        this.allVolCountMacTrackBar.TextTickStyle =
System.Windows.Forms.TickStyle.None;
        this.allVolCountMacTrackBar.TickColor =
System.Drawing.Color.FromArgb(((int)((byte)(148))), ((int)((byte)(146))),
((int)((byte)(148))));
        this.allVolCountMacTrackBar.TickHeight = 4;
        this.allVolCountMacTrackBar.TickStyle =
System.Windows.Forms.TickStyle.None;
        this.allVolCountMacTrackBar.TrackerColor =
System.Drawing.Color.FromArgb(((int)((byte)(24))), ((int)((byte)(130))),
((int)((byte)(198))));
        this.allVolCountMacTrackBar.TrackerSize = new
System.Drawing.Size(10, 16);
        this.allVolCountMacTrackBar.TrackLineColor =
System.Drawing.Color.FromArgb(((int)((byte)(90))), ((int)((byte)(93))),
((int)((byte)(90))));
        this.allVolCountMacTrackBar.TrackLineHeight = 3;
        this.allVolCountMacTrackBar.Value = 2;
        this.allVolCountMacTrackBar.ValueChanged += new
EConTech.Windows.MACUI.ValueChangedHandler(this.allVolCountMacTrackBar_ValueChan
ged);
        this.allVolCountMacTrackBar.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.allVolCountMacTrackBar_MouseUp);
        //
        // browser
        //
        this.browser.AutoValidate =
System.Windows.Forms.AutoValidate.EnablePreventFocusChange;
        this.browser.ListViewMode = System.Windows.Forms.View.List;
        this.browser.Location = new System.Drawing.Point(12, 131);
        this.browser.Name = "browser";
        treeNode1.Name = "";
        treeNode1.Text = "";
        treeNode2.ImageIndex = 18;
        treeNode2.Name = "backreg";
        treeNode2.SelectedImageIndex = 20;
        treeNode2.Text = "backreg";
        treeNode3.Name = "";
        treeNode3.Text = "";
        treeNode4.ImageIndex = 18;
        treeNode4.Name = "CISCO_CCNA";
        treeNode4.SelectedImageIndex = 20;
        treeNode4.Text = "CISCO_CCNA";

```

```

treeNode5.Name = "";
treeNode5.Text = "";
treeNode6.ImageIndex = 18;
treeNode6.Name = "Documents and Settings";
treeNode6.SelectedImageIndex = 20;
treeNode6.Text = "Documents and Settings";
treeNode7.Name = "";
treeNode7.Text = "";
treeNode8.ImageIndex = 18;
treeNode8.Name = "Downloads";
treeNode8.SelectedImageIndex = 20;
treeNode8.Text = "Downloads";
treeNode9.Name = "";
treeNode9.Text = "";
treeNode10.ImageIndex = 18;
treeNode10.Name = "Inprise";
treeNode10.SelectedImageIndex = 20;
treeNode10.Text = "Inprise";
treeNode11.Name = "";
treeNode11.Text = "";
treeNode12.ImageIndex = 18;
treeNode12.Name = "Program Files";
treeNode12.SelectedImageIndex = 20;
treeNode12.Text = "Program Files";
treeNode13.ImageIndex = 33;
treeNode13.Name = "Recycled";
treeNode13.SelectedImageIndex = 34;
treeNode13.Text = "Recycled";
treeNode14.ImageIndex = 18;
treeNode14.Name = "RECYCLER";
treeNode14.SelectedImageIndex = 20;
treeNode14.Text = "RECYCLER";
treeNode15.ImageIndex = 18;
treeNode15.Name = "System Volume Information";
treeNode15.SelectedImageIndex = 20;
treeNode15.Text = "System Volume Information";
treeNode16.ImageIndex = 18;
treeNode16.Name = "temp";
treeNode16.SelectedImageIndex = 20;
treeNode16.Text = "temp";
treeNode17.Name = "";
treeNode17.Text = "";
treeNode18.ImageIndex = 18;
treeNode18.Name = "WINDOWS";
treeNode18.SelectedImageIndex = 20;
treeNode18.Text = "WINDOWS";
treeNode19.ImageIndex = 23;
treeNode19.Name = "Локальный диск (C:)";
treeNode19.SelectedImageIndex = 24;
treeNode19.Text = "Локальный диск (C:)";
this.browser.SelectedNode = treeNode19;
this.browser.ShowFoldersButton = false;
this.browser.ShowNavigationBar = false;
this.browser.Size = new System.Drawing.Size(962, 432);
this.browser.SplitterDistance = 398;
this.browser.StartupDirectoryOther = "C:\\";
this.browser.TabIndex = 0;
this.browser.Load += new System.EventHandler(this.browser_Load);
//
// testButton
//
this.testButton.FlatStyle = System.Windows.Forms.FlatStyle.Popup;
this.testButton.Image =
global::RecoveryStar.Properties.Resources.table_sql_view32451;
this.testButton.ImageAlign =
System.Drawing.ContentAlignment.TopCenter;
this.testButton.Location = new System.Drawing.Point(111, 27);
this.testButton.Name = "testButton";
this.testButton.Size = new System.Drawing.Size(100, 97);

```

```

        this.testButton.TabIndex = 4;
        this.testButton.Text = "Перевірка цілісності";
        this.testButton.TextAlign =
System.Drawing.ContentAlignment.BottomCenter;
        this.testButton.UseVisualStyleBackColor = true;
        this.testButton.Click += new
System.EventHandler(this.testButton_Click);
        //
        // repairButton
        //
        this.repairButton.FlatStyle = System.Windows.Forms.FlatStyle.Popup;
        this.repairButton.Image =
global::RecoveryStar.Properties.Resources.medical_bag465471;
        this.repairButton.ImageAlign =
System.Drawing.ContentAlignment.TopCenter;
        this.repairButton.Location = new System.Drawing.Point(210, 27);
        this.repairButton.Name = "repairButton";
        this.repairButton.Size = new System.Drawing.Size(100, 97);
        this.repairButton.TabIndex = 3;
        this.repairButton.Text = "Відновлення цілісності";
        this.repairButton.TextAlign =
System.Drawing.ContentAlignment.BottomCenter;
        this.repairButton.UseVisualStyleBackColor = true;
        this.repairButton.Click += new
System.EventHandler(this.repairButton_Click);
        //
        // recoverButton
        //
        this.recoverButton.FlatStyle = System.Windows.Forms.FlatStyle.Popup;
        this.recoverButton.Image =
global::RecoveryStar.Properties.Resources.redo6786987;
        this.recoverButton.ImageAlign =
System.Drawing.ContentAlignment.TopCenter;
        this.recoverButton.Location = new System.Drawing.Point(309, 27);
        this.recoverButton.Name = "recoverButton";
        this.recoverButton.Size = new System.Drawing.Size(100, 97);
        this.recoverButton.TabIndex = 2;
        this.recoverButton.Text = "Скидання дисків до початкового стану";
        this.recoverButton.TextAlign =
System.Drawing.ContentAlignment.BottomCenter;
        this.recoverButton.UseVisualStyleBackColor = true;
        this.recoverButton.Click += new
System.EventHandler(this.recoverButton_Click);
        //
        // protectButton
        //
        this.protectButton.BackColor = System.Drawing.SystemColors.Control;
        this.protectButton.FlatStyle = System.Windows.Forms.FlatStyle.Popup;
        this.protectButton.Font = new System.Drawing.Font("Microsoft Sans
Serif", 8.25F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte) 204));
        this.protectButton.Image =
global::RecoveryStar.Properties.Resources.Database_1_64x64e65768;
        this.protectButton.ImageAlign =
System.Drawing.ContentAlignment.TopCenter;
        this.protectButton.Location = new System.Drawing.Point(12, 27);
        this.protectButton.Name = "protectButton";
        this.protectButton.Size = new System.Drawing.Size(100, 97);
        this.protectButton.TabIndex = 1;
        this.protectButton.Text = "Створення RaidSSD масиву";
        this.protectButton.TextAlign =
System.Drawing.ContentAlignment.BottomCenter;
        this.protectButton.UseVisualStyleBackColor = false;
        this.protectButton.Click += new
System.EventHandler(this.protectButton_Click);
        //
        // ToolStripMenuItem
        //

```

```

        this.ToolStripMenuItem.Image =
global::RecoveryStar.Properties.Resources.Exit;
        this.ToolStripMenuItem.Name = "ToolStripMenuItem";
        this.ToolStripMenuItem.Size = new System.Drawing.Size(152, 22);
        this.ToolStripMenuItem.Text = "Вихід";
            this.ToolStripMenuItem.Click += new
System.EventHandler(this.виходToolStripMenuItem_Click);
        //
        // ToolStripMenuItem
        //
            this.тестБыстродействияToolStripMenuItem.Image =
global::RecoveryStar.Properties.Resources.StartBenchmark;
        this.тестБыстродействияToolStripMenuItem.ImageScaling =
System.Windows.Forms.ToolStripItemImageScaling.None;
        this.ToolStripMenuItem.Name = "ToolStripMenuItem";
        this.ToolStripMenuItem.Size = new System.Drawing.Size(161, 22);
        this.ToolStripMenuItem.Text = "Тест швидкодії";
            this.ToolStripMenuItem.Click += new
System.EventHandler(this.тестБыстродействияToolStripMenuItem_Click);
        //
        // MainForm
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(986, 576);
        this.Controls.Add(this.coderConfigGroupBox);
        this.Controls.Add(this.testButton);
        this.Controls.Add(this.repairButton);
        this.Controls.Add(this.recoverButton);
        this.Controls.Add(this.protectButton);
        this.Controls.Add(this.browser);
        this.Controls.Add(this.menuStrip);
            this.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedDialog;
        this.Icon =
((System.Drawing.Icon) (resources.GetObject("$this.Icon")));
        this.MainMenuStrip = this.menuStrip;
        this.MaximizeBox = false;
        this.Name = "MainForm";
            this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
        this.Text = "Гарантоване збереження інформації ";
        this.Load += new System.EventHandler(this.MainForm_Load);
        this.FormClosing += new
System.Windows.Forms.FormClosingEventHandler(this.MainForm_FormClosi
ng);
        this.menuStrip.ResumeLayout(false);
        this.menuStrip.PerformLayout();
        this.coderConfigGroupBox.ResumeLayout(false);
        this.redundancyGroupBox.ResumeLayout(false);
        this.redundancyGroupBox.PerformLayout();
        this.allVolCountGroupBox.ResumeLayout(false);
        this.allVolCountGroupBox.PerformLayout();
        this.ResumeLayout(false);
        this.PerformLayout();

    }

#endregion

private System.Windows.Forms.MenuStrip menuStrip;
private System.Windows.Forms.ToolStripItem ToolStripMenuItem;
private System.Windows.Forms.ToolStripItem ToolStripMenuItem;
private System.Windows.Forms.Button protectButton;
private System.Windows.Forms.ToolStripItem ToolStripMenuItem;
    private System.Windows.Forms.ToolStripItemSeparator
separatorToolStripMenuItem;
private System.Windows.Forms.ToolStripItem ToolStripMenuItem;
private System.Windows.Forms.Button repairButton;

```

```
private System.Windows.Forms.Button testButton;
private System.Windows.Forms.GroupBox coderConfigGroupBox;
private System.Windows.Forms.GroupBox redundancyGroupBox;
private System.Windows.Forms.GroupBox allVolCountGroupBox;
private EConTech.Windows.MACUI.MACTrackBar allVolCountMacTrackBar;
private EConTech.Windows.MACUI.MACTrackBar redundancyMacTrackBar;
private System.Windows.Forms.ToolTip toolTip;
private System.Windows.Forms.ToolStripItem ToolStripMenuItem;
private System.Windows.Forms.ToolStripItem ToolStripMenuItem;
internal FileBrowser.Browser browser;
private System.Windows.Forms.ToolStripItem ToolStripMenuItem;
private System.Windows.Forms.ToolStripItem ToolStripMenuItem;
private System.Windows.Forms.Button recoverButton;
}
}
```

K6П3_2025