

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет радіоелектроніки

Невлюдов І.Ш., Андрусевич А.О., Євсєєв В.В.,
Максимова С.С., Стародубцев М.Г., Невлюдова В.В.

**АВТОМАТИЗОВАНА СИСТЕМА КЕРУВАННЯ
ТЕХНОЛОГІЧНИМИ ПРОЦЕСАМИ
В SCADA СИСТЕМІ TRACE MODE 6
(частина 1)**

КК НАУ
Кривий Ріг
2018

Рецензенти:

доктор технічних наук, професор,
завідувач кафедри «Інформаційних управляючих систем»
Харківського національного університету радіоелектроніки
ЛЕВИКІН ВІКТОР МАКАРОВИЧ

доктор технічних наук, професор,
професор кафедри «Теоретичної та прикладної інформатики»
Харківського національного університету імені В. Н. Каразіна
ФРОЛОВ В'ЯЧЕСЛАВ ВІКТОРОВИЧ

Н 40 **Невлюдов І.Ш., Андрусевич А.О., Євсєєв В.В., Максимова С.С., Стародубцев М.Г., Невлюдова В.В.** Автоматизована система керування технологічними процесами в SCADA системі TRACE MODE 6: Навчальний посібник. – Кривий Ріг: Криворізький коледж НАУ, 2018 р. – 320 с.

У навчальному посібнику викладено питання огляду структури інформаційних потоків сучасного виробництва та класифікації АСК ТП. Розглянуто структуру TRACE MODE 6, основні модулі й їх призначення, проведено аналіз сучасних SCADA систем та їх вибір в залежності від поставлених задач. Описані сучасні датчики та переграмовані логічні контролери для проектування АСК ТП в SCADA TRACE MODE 6.

Наведені приклади задач, які розв'язуються за допомогою SCADA TRACE MODE 6. Розглянуто механізм обміну даними та типи інтерфейсів, метод розподілених комплексів. Наведено опис посібника користувача та основний функціонал SCADA TRACE MODE 6, описані основні модулі й їх призначення, проаналізовані мови програмування Techno ST, Techno FBD, Techno SFC, Techno IL. Розглянуто приклади розв'язання типових задач з розробки АСК ТП, що постають перед технологами на виробництві.

Навчальний посібник призначений для студентів закладів вищої освіти за спеціальністю 151 – Автоматизація та комп'ютерно-інтегровані технології, зокрема спеціалізацій: «Автоматизація та комп'ютерно-інтегровані ресурсозберігаючі технології. Альтернативна енергетика», «Автоматизація та комп'ютерно-інтегровані технології видобування нафти і газу», а також для фахівців в галузі автоматизації приладобудування.

Рекомендовано Вченою радою Харківського національного університету радіоелектроніки (протокол №8 від 03.07.2018 року).

ISBN 980-970-518-233-7

ББК 32.944.1+30.860.2

© І.Ш. Невлюдов
А.О. Андрусевич
В.В. Євсєєв
С.С. Максимова
М.Г. Стародубцев
В.В. Невлюдова.
2018

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	5
ВСТУП.....	9
1. СУЧАСНІ АВТОМАТИЗОВАНІ СИСТЕМИ КЕРУВАННЯ ТЕХНОЛОГІЧНИМИ ПРОЦЕСАМИ.....	12
1.1. Структура інформаційних потоків сучасного виробництва.....	12
1.2. Класифікація сучасних АСК ТП.....	25
1.3. Поняття SCADA систем і їхнє призначення.....	27
1.4. Аналіз сучасних SCADA систем і вибір в залежності від поставлених задач.....	35
1.5. Контрольні питання.....	38
2. СУЧАСНІ ДАВАЧІ ТА ПЕРЕГРАМОВАНІ ЛОГІЧНІ КОНТРОЛЕРИ ДЛЯ ПРОЕКТУВАННЯ АСК ТП В SCADA TRACE MODE 6.....	40
2.1. Структура автоматизованої системи керування технологічними процесами на прикладах.....	40
2.2. Сучасні давачі й їх характеристики для отримання даних технологічного процесу.....	69
2.3. Програмовані логічні контролери їх структура та характеристика.....	95
2.4. Контрольні питання.....	112
3. TRACE MODE 6.0 В РОЗВ'ЯЗАННІ ЗАДАЧ АСК ТП.....	114
3.1. Галузь задач, які розв'язуються в SCADA TRACE MODE 6....	114
3.2. Структура TRACE MODE 6, основні модулі й їх призначення	119
3.3. Інсталяція TRACE MODE 6.....	124
3.4. Основні визначення та поняття в SCADA TRACE MODE 6....	130
3.5. Механізм обміну даними та типи інтерфейсів.....	137
3.6. Метод розподілених комплексів.....	146
3.7. Контрольні питання.....	148
4. ІНТЕРФЕЙС КОРИСТУВАЧА ТА ОСНОВНИЙ ФУНКЦІОНАЛ TRACE MODE	149

6.....	
4.1. Інтерфейс і функціонал TRACE MODE 6.....	149
4.1.1. Навігатор проекту.....	150
4.1.2. Редагування каналу.....	150
4.1.3. Прив'язка аргументів.....	152
4.2. Створення об'єктів екрана.....	154
4.2.1. Статичне зображення.....	157
4.2.2. Динамічне зображення.....	161
4.3. Тривоги та події.....	165
4.4. Тренди.....	166
4.5. Програмування в SCADA–системі TRACE MODE.....	167
4.6. Звіти.....	170
4.7. Контрольні питання.....	178
5. ПРОГРАМУВАННЯ В TRACE MODE.....	179
5.1. Створення програм в TRACE MODE.....	179
5.2. Синтаксис мови Techno ST.....	181
5.3. Мова Techno FBD.....	191
5.4. Мова Techno SFC.....	194
5.5. Мова Techno IL.....	202
5.6. Контрольні питання.....	207
6. РОЗРОБКА АСК ТП НА ПРИКЛАДАХ.....	208
6.1. Підключення стандартних графічних елементів з бібліотеки.....	208
6.2. Створення проекту та налаштування каналів.....	209
6.3. Створення статичних і динамічних об'єктів.....	226
6.4. Програмування на мовах Techno IL і Techno SFC.....	245
6.5. Створення звіту тривоги та СПАД архіву.....	254
6.6. Контрольні питання.....	261
6.7. Контрольні завдання.....	261
ПЕРЕЛІК ПОСИЛАНЬ.....	265
Додаток А. Опис основних елементів інтерфейсу.....	309
Додаток Б. Арифметичні FBD блоки.....	310
Додаток В. FBD блоки порівняння.....	312
Додаток Г. FBD блоки вибору.....	315
Додаток Д. FBD блоки – генератори.....	318

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Active-X –	розроблена Microsoft технологія, призначена для надання доступу до Windows-додатків через Internet;
ATMS –	Advanced Traffic Management System (перспективне керування зверху вниз, яка об'єднує технології, перш за все, для поліпшення потоку руху транспортних засобів та підвищення безпеки);
CS –	Communication System (комунікаційна система);
DCOM –	Distributed Component Object Model (розширення Component Object Model для підтримки зв'язку між об'єктами на різних комп'ютерах по мережі);
DCS –	Distributed Control System (розподілена система керування (РСК));
DDE –	Dynamic Data Exchange (механізм взаємодії застосування додатків в операційних системах Microsoft Windows и OS/2);
DRAM –	Dynamic Random Access Memory (динамічна пам'ять з довільним доступом);
DVI –	Digital Visual Interface (стандарт на інтерфейс, призначений для передачі відеозображення на цифрові пристрої відображення);
EAM –	Enterprise Asset Management (системи управління основними фондами);
EEPROM –	Electrically Erasable Programmable Read-Only Memory (електрично стерте перепрограмоване ПЗП);
ERP –	Enterprise Resource Planning (планування ресурсів підприємства);
FBD –	Function Block Diagram, (графічна мова програмування стандарту МЕК 61131-3. Призначений для програмування програмованих логічних контролерів (ПЛК));
HMI –	Human-machine interface (людино-машинний інтерфейс);

HRM –	Human resources management (керування персоналом);
IL –	Instruction List (програма створена з використанням мови інструкцій Trace Mode);
LAD –	Ladder Diagrams (мова релейний (сходовий) логіки);
LVDS –	Low-voltage differential signaling (спосіб передачі електричних сигналів);
MTU –	Master Terminal Unit (головний термінал керування);
MMI –	Man Machine Interface (людино-машинний інтерфейс);
MPB –	Real-time monitor (монітор реального часу, має в системі TRACE MODE такі підкласи: MPB+, ДокMPB+, Double Force MPB+, GSM MPB+)
OLE –	Object Linking and Embedding (технологія зв'язування та впровадження об'єктів в інші документи і об'єкти, розроблена корпорацією Microsoft);
PLC –	Programmable Logic Controller (програмний логічний контролер);
RISC –	Reduced Instruction Set Computing (архітектура процесора, в якому швидкодія збільшується за рахунок спрощення інструкцій, щоб їх декодування було простішим, а час виконання - меншим);
RTU –	Remote Terminal Unit (віддалений термінал);
SCADA –	Supervisory Control And Data Acquisition (система супервизорного керування та збору даних);
SDRAM –	Synchronous Dynamic Random Access Memory (синхронна динамічна пам'ять з довільним доступом);
SFC –	Sequential Function Chart (мова програмування стандарту IEC61131-3. Призначений для програмування промислових контролерів. Широко використовується в SCADA / НМІ пакетах.);
SQL –	Structured Query Language (декларативна мова програмування, застосовувана для створення, модифікації та керування даними в реляційній базі даних, керованої відповідною системою керування базами даних);
SRAM –	Static Random Access Memory (напівпровідникова

	оперативна пам'ять);
STL –	Standard Template Library (реалізація мови Instruction List для програмування контролерів);
VGA –	Video Graphics Array (компонентний відеоінтерфейс);
A –	абоненти;
АЛП –	арифметико-логічний пристрій;
АСКП –	автоматизована система керування підприємством;
АСК ТП –	автоматизована система керування технологічними процесами;
АСКО –	автоматизована система комерційного обліку;
АСОДК –	автоматизована система оперативного диспетчерського керування;
АРМ –	автоматизоване робоче місце;
АЦП –	аналого-цифровий перетворювач;
БР –	багатоканальні реєстратори;
БЦК –	безпосереднього цифрового керування;
ВМ –	виконавчі механізми;
Д –	датчик;
ЕРС –	електрорушійна сила;
ІАСК –	інформаційна автоматизована систем керування;
ІАСКП –	інформаційна автоматизована система керування підприємства;
К –	контролери;
КОК –	керуючий обчислювального комплексу;
ЛСК –	локальні системи керування;
ЛР –	локальний регулятор;
ЛТС –	локальні технологічні станції;
НІС –	надвелика інтегральна схема;
НП –	номеруючий претворювач;
ОЗП –	оперативний запам'ятовуючий пристрій;
ПАЗ –	протиаварійний захист
ПДУ –	пульта дистанційного керування;
ПЗ –	програмне забезпечення;
ПЗО –	пристрій зв'язку з об'єктом;
ПЗП –	постійний запам'ятовуючий пристрій;

ПСО –	пристрою сполучення з об'єктом керування. Елементи апаратної частини АСК ТП, призначені для узгодження ЕВМ або в загальному випадку мікропроцесорної системи з датчиками і виконавчими пристроями, встановленими на об'єкті керування;
ПУ –	пульт оператора (пульт керування);
РБК –	редактор бази каналів, інструментальне засіб TRACE MODE для розробки структури проекту АСК ТП;
РКІ –	рідко-кристалічний індикатор;
РО –	регулюючий орган;
РІД –	редактор представлення даних. Інструментальний засіб TRACE MODE для розробки екранів візуалізацій технологічних процесів;
СПАД –	локальний архів МРВ, який реалізує технологію структурованих промислових архівів даних;
СКБД –	система керування баз даних;
ТЕП –	техніко-економічні показники;
УВМ –	універсальна обчислювальна машина
ФВ –	форма відображення. Комплекс графічних елементів, які використовуються для візуалізації динамічної та статичної технологічної інформації. З ФВ збирають екрани візуалізації технологічних процесів;
ЦВ –	центральний вузол;
ЦПЗ –	центральний пристрій зв'язку;
ЦПП –	центральний процесорний пристрій;
ШІМ –	широко-імпульсна модуляція;

ВСТУП

Сучасне виробництво в умовах ринкової економіки неможливе без застосування систем автоматизованого проектування. Використання нових підходів до автоматизації знайшло своє втілення в міжнародному стандарті Industry 4.0, який поєднує нові концепції в розвитку керування виробництвом, у його моніторингу та управлінні життєвим циклом продукції. Зазначимо, що даний концепт неможливо застосовувати без використання вже існуючих підходів до автоматизованих систем керування технологічними процесами (АСК ТП). Такі системи мають складну структуру реалізації, що базується на симбіозі апаратного (давачі, ПЛК) та програмного забезпечення (CAD/CAM/CAE, SCADA системи), і поєднані за допомогою промислових і гло-бальних мереж. Зауважимо, що реалізація завдань керування технологічними процесами кожен рік ускладнюються, з чого випливає проблема зростання потреби контролю його в режимі реального часу. Це накладає жорсткі обмеження на вибір та побудову АСК ТП. Однією з найважливіших складових побудови АСК ТП є розробка та впровадження в процес виробництва SCADA систем і вибір необхідного апаратного забезпечення. Використання таких систем не залежить від типу та виду виробництва: від важкої промисловості (машинобудування, важкий прокат металів, газотурбінне виробництво) до переробки та видобутку природних ресурсів (нафти, газу) та контролю отримання альтернативної енергії (біопалива).

Навчальний посібник являє собою логічну послідовність розділів, яка дозволяє вивчити та поглибити знання з сучасних SCADA систем на базі TRACE MODE 6.

Перший розділ присвячений розгляду сучасних автоматизованих систем керування технологічними процесами. Наведено приклад і проаналізовано структуру інформаційних потоків виробництва, класифікація АСК ТП, методи їх

проектування. Визначені основні поняття SCADA систем і їх призначення, обґрунтований їх вибір в залежності від розв'язуваних завдань.

У другому розділі розглянуті апаратні складові АСК ТП першого і другого рівня. Наведено характеристики сучасних датчиків і призначення тих, що використовуються для контролю технологічного процесу. Дана класифікація та типи програмованих логічних контролерів, їх види та способи підключення. Проаналізовано та наведено технічні характеристики ПЛК.

Третій розділ присвячений SCADA TRACE MODE 6 в галузі вирішення завдань розробки АСК ТП, розглянуто його структура, дані основні визначення та поняття необхідні для роботи, описані механізми обміну даними та типи інтерфейсу. Наведено покроковий алгоритм інсталяції TRACE MODE 6 на комп'ютер користувача.

Особливості інтерфейсу користувача з описом всіх його елементів розглянуті в четвертому розділі. Так само наведені приклади створення статичних і динамічних зображень, тривог і подій, трендів. Особливу увагу приділено створенню та редагуванню каналів і прив'язці аргументів.

Окремо в п'ятому розділі розглянуті сучасні мови програмування в SCADA TRACE MODE 6. Описано синтаксис і правила створення програм для мов Techno ST, Techno FBD, Techno SFC, Techno IL, наведені приклади програм під вирішення завдань, в додатку наведені основні блоки їх опис та синтаксис.

Для закріплення вивченого матеріалу в попередніх розділах шостий розділ представлений у вигляді постановки певних завдань від найпростішої реалізації графічного інтерфейсу управління з простими трендами до складних з елементами програмування, а також покрокове рішення. Завдання розроблені таким чином, що на базі виконаного й отриманого результату можна ускладнити завдання з використанням нових підходів до розробки.

Навчальний посібник призначений для студентів закладів вищої освіти за спеціальністю 151 – Автоматизація та комп'ютерно-інтегровані технології, зокрема спеціалізацій: «Автоматизація та комп'ютерно-інтегровані ресурсозберігаючі технології. Альтернативна енергетика», «Автоматизація та комп'ютерно-інтегровані технології видобування нафти і газу», а також для фахівців в галузі автоматизації приладобудування.

1. СУЧАСНІ АВТОМАТИЗОВАНІ СИСТЕМИ КЕРУВАННЯ ТЕХНОЛОГІЧНИМИ ПРОЦЕСАМИ

1.1. Структура інформаційних потоків сучасного виробництва

Разом з виникненням дійсного власника на підприємствах з'явилася задача підвищення ефективності виробництва й якості продукції, а також забезпечення нової якості керованості за рахунок створення єдиного інформаційного простору підприємства. Цього можна досягнути шляхом використання всієї вірогідної оперативної інформації від усіх об'єктів виробництва. Реальним інструментом для досягнення поставленої мети є комплексна інтеграція окремих підсистем всього підприємства.

Передовий світовий досвід доводить, що підвищення ефективності діяльності сучасного підприємства можливе лише при наявності єдиної системи, що поєднує керування фінансами, персоналом, постачанням, збутом і власне виробництвом. Такі системи розглядаються як засіб досягнення основних цілей бізнесу: підвищення якості продукції, зниження втрат і збільшення обсягів виробництва, займання стійких позицій і отримання суттєвих конкурентних переваг на ринку.

Використання замкнених на себе систем керування підприємством (R3, BAAN, Галактика і ін.) не вирішує проблеми промислового підприємства в цілому, вони є самодостатніми лише для автоматизації задач адміністративно-керівного рівня підприємства. Для використання всіх можливостей таких систем в повному обсязі необхідне введення в них оперативних і вірогідних даних з рівня технологічних і виробничих процесів. Оперативність отримання виробничої інформації дозволяє всім рівням керування підприємством забезпечити поточний контроль і моніторинг основного та допоміжного виробничого процесу в реальному масштабі часу.

Для того, щоб інформаційна система підприємства вирішувала задачу зниження загальної собівартості продукції та призводила до інтегрованого поняття економічної ефективності виробництва в цілому, інформаційна система має містити наступні компоненти:

- необхідна кількість достатньо точного первісного обладнання (давачі, виконавчі механізми, контролери) на основному та допоміжному технологічному процесі;
- сучасний промисловий контролерний і комп'ютерний парк і промислові комунікації;
- сучасний програмний інструментарій для обробки, архівування та подання оперативної технологічної, облікової та комерційної інформації;
- розвинена комунікаційна інфраструктура підприємства.

У своїй діяльності виробниче підприємство оперує не лише термінами вихідна сировина, основні фонди, продукція і т.д. Сьогодні на перший план виходить така категорія як інформація. Всіляка новація на підприємстві має розглядатися з точки зору придбання нової якості інформації. Комплексна інтеграція, безумовно, цю якість надає. Вона забезпечує керівництво підприємства об'єктивною інформацією, що є першим кроком для побудови ефективного виробництва.

Крім цього, комплексна інтеграція підприємства сприяє створенню в межах підприємства єдиного банку даних про продукцію (з введенням паспорта на виріб), технологічні процеси, дані допоміжних виробництв, знижує ступінь дублювання інформації та забезпечує стандартизацію всієї діяльності підприємства. Внаслідок чого знижується рівень витрат виробництва, підвищується якість продукції, з'являється можливість оперативної зміни номенклатури продукції. Більш того, досягнута стандартизація забезпечує можливість оперативного впровадження на підприємстві всіх сучасних технологічних досягнень.

Актуальність робіт з комплексної інтеграції зумовлено наступним набором об'єктивних чинників:

- розв'язати задачу підвищення ефективності виробництва можливо лише на основі об'єктивної картини технічних і технологічних параметрів;
- інформаційні й організаційні бар'єри, що існують між керівними та технологічними рівнями підприємства, призводять до блокування важливої для аналізу діяльності підприємства інформації, а також різко знижують оперативність прийняття керівних рішень;
- ринок засобів і систем автоматизації вже сьогодні пропонує всі необхідні компоненти для здійснення комплексної інтеграції.

Інтеграція систем автоматизації має йти в двох напрямках: горизонтальному та вертикальному.

Горизонтальна інтеграція – це поєднання між собою всіх автономних систем автоматизації технологічних і виробничих процесів, а також адміністративних відділів цехового рівня в єдину інформаційну мережу, що забезпечує необхідний обмін даними в реальному масштабі часу між всіма підрозділами основного та допоміжного виробництва. В тому числі це і система обліку, організована на кожному етапі виробничого процесу від поставки вихідної сировини до складу готової продукції. Вертикальна інтеграція – це інтеграція систем керування підприємством (АСКП, ERP) і систем керування технологічними процесами (АСК ТП) з метою забезпечення максимальної ефективності всіх систем автоматизації. Вертикальна інтеграція базується на організації потоків інформації від нижнього рівня (давачів і контролерів технологічного обладнання) у внутрішні та зовнішні комп'ютерні мережі підприємства і через них в адміністративні системи керування. Дана задача вирішується на основі поєднання промислових і адміністративних мереж. Основна мета вертикальної інтеграції – видалення перешкод на шляху інформаційних потоків між рівнями АСКП і АСК ТП для оперативного обміну даними.

В основі сучасних інтегрованих систем автоматизації лежить використання компонентів, які задовольняють таким властивостям як відкритість, стандартизація, типізація рішень, масштабованість систем, комплексність підходу та тиражованість.

Відкритість системи – категорія не лише технічна, але й економічна. Відкритість стандартів на системи автоматизації означає відсутність патентів або авторських прав на специфікацію стандарта та його поширення, відсутність ліцензійної сплати за використання стандартів, широку доступність всіх спеціалістів до розробки, виробництва та використання продукції в даному стандарті. Відкритість означає застосування відкритих стандартів, визначає гнучкість архітектури системи автоматизації, дружність користувацького інтерфейсу, можливість взаємодії з іншими системами за рахунок сумісності широкого спектра стандартизованих виробів і програм на різних рівнях

Стандартизація передбачає використання компонентів систем автоматизації, що основані на існуючих стандартах на програмні та технічні рішення. Сучасні тенденції в галузі стандартизації такі, що системи автоматизації, побудовані на основі різних рішень, мають інтегруватися в єдині системи та комплекси без серйозних додаткових розробок. На місце конкретних фірмових рішень мають приходити відкриті міжнародні стандарти. Застосування стандартів при побудові системи автоматизації є показником її якості, гарантує користувачу сучасний технічний рівень і можливість успадкування параметрів системи в процесі її подальшого розвитку та модернізації.

Типізація системних рішень має бути закладена в основу кожної системи автоматизації, що розробляється. Дотримання цього принципу дозволяє мати наявний набір як компонентів, так і рішень, що сприяє економії при закупівлі компонентів і впровадженні систем.

Масштабованість систем. Дотримання принципу використання стандартів для компонентів і рішень дозволить створювати масштабовані, нарощувані системи автоматизації, що найбільш відповідають цілям і завданням, які стоять перед ними. Масштабованість дозволяє створювати і модернізувати системи автоматизації з мінімальними засобами, що забезпечують необхідні функції.

Комплексний підхід до систем автоматизації має на увазі не тільки традиційний (об'єктовий) підхід, але і цільовий, орієнтований на використання засобів автоматизації різного рівня для досягнення конкретних цілей (зниження енерговитрат, собівартості продукції, підвищення якості продукції та послуг).

Тиражованість. Кожна система автоматизації повинна мати властивість відтворюваності і тиражованості, що має прямий вплив на витрати на створення систем і в кінцевому підсумку на собівартість продукції.

Особливість українського ринку автоматизації полягає в складному поєднанні представлених на ньому систем, відмінних один від одного за класом і ідеологією, побудованих часом на несумісних один з одним протоколах, інтерфейсах і контролерах. Навіть в рамках одного підприємства часто можна зустріти цілий букет підсистем як сучасних, так і безнадійно фізично і морально застарілих. В даний час на багатьох підприємствах експлуатуються засоби автоматики і телемеханіки, розроблені ще в 80-х роках. Практично всі вони відпрацювали свій технологічний ресурс, не можуть бути модернізовані і вимагають заміни. Крім того, автоматизація, що існує, носить клаптевий характер. Як правило, відсутній автоматичний контур, який об'єднує систему обліку вихідної сировини і відвантаження продуктів. Відсутній автоматизований моніторинг і диспетчеризація основного й допоміжних виробництв, що стримує прийняття оперативних рішень.

Це розмаїття призводить до того, що часто, вводячи нові підсистеми АСК ТП, необхідно забезпечувати їх взаємодію з тими, що вже існують. У великих проектах, де беруть участь

кілька субпідрядників, потрібно забезпечувати сумісність і взаємодію підсистем від різних виробників. У цій ситуації багато часу і сил доводиться витратити на опрацювання взаємодії підсистем, часто змінюючи їх і додаючи в них нові комунікаційні елементи. Архітектура великих систем стає громіздкою, штучною і важко модифікується. Всі ці складнощі і проблеми ми намагалися узагальнити та запропонувати ідеї та рішення, що дозволяють задачу автоматизації будь-якої складності розбити на окремі логічні рівні і запропонувати для неї набір стандартних, в нашому розумінні, засобів і технологій. Отже, наш досвід проектування і розробки різних систем автоматизації говорить про те, що, з точки зору комплексності, можна виділити три класи інформаційних автоматизованих систем керування (ІАСК) самого різного призначення:

Клас 1: інтеграційні сервери АСК ТП / АСКО (автоматизована система комерційного обліку). Практична робота зі створення різних систем АСК ТП для різних галузей дозволяє ввести такі типи інтеграції окремих підсистем АСК ТП / АСКО в єдину систему АСК ТП

Тип 1 на основі комунікаційного сервер-шлюзу. Забезпечення прозорого транспорту даних з однієї підсистеми в іншу на основі підтримки різних промислових і локальних мереж. У таких конфігураціях, як правило, немає необхідності вести додаткову обробку даних. На цьому підході, наприклад, побудований проху-сервер для ProfiNet (шлюз Ethernet-Profibus)

Тип 2 на основі концентратора даних. Як правило, він організовується на базі промислового комп'ютера або високопродуктивного контролера, що забезпечує як функції комунікаційного сервера, так і функції, пов'язані з обробкою оперативних даних нижнього рівня і їх архівуванням в спеціалізованій базі даних.

Тип 3 на основі інтеграційного сервера АСК ТП. Його завдання об'єднання різних підсистем АСК ТП в єдину систему з прозорою організацією даних і розвиненими прикладними

функціями. Сервери цього типу включають в себе функції комунікаційного сервера і концентратора даних та виконують при цьому широкий набір спеціальних функцій для обробки даних, а також реалізують комплексні алгоритми управління, забезпечують синхронізацію роботи підсистем АСК ТП і підтримку єдиного часу в системі. Крім цього, забезпечення можливості функціонування в промислових умовах (температура, вологість, вібрація), підтримка промислових комунікацій (наприклад, PROFIBUS, FF, CAN, MODBUS) і базових протоколів локальних мереж (TCP / IP, IPX / SPX, NETBIOS), забезпечення роботи з модемами і телекомунікаційними каналами і протоколами, а також забезпечення взаємодії з SCADA-системами та системами керування базами даних (СКБД) верхнього рівня.

Перераховані типи інтеграційних серверів АСК ТП вирішують цілий ряд спеціальних функцій, перелік яких безпосередньо залежить від області застосування. Це функції для обробки даних (верифікація, масштабування); функції комплексних алгоритмів керування, тестування та діагностики апаратно-програмних засобів; функції підтримки єдиного часу та синхронізації роботи окремих підсистем; функції підтримки єдиної адресації параметрів системи і організація архівів за обраними параметрами; функції буферизації інформації та резервування каналів передачі даних. Конкретний тип і конфігурація інтеграційного сервера АСК ТП визначається набором функцій. Він повинен бути оптимальним за ціною і функціями, що виконуються.

Клас 2: АСОДК (автоматизована система оперативного диспетчерського керування) підприємства. Актуальність робіт зі створення та впровадження на своєму підприємстві АСОДК кожен керівник розуміє по-своєму. Однак є ряд об'єктивних факторів, що впливають на прийняття рішення. Так, вирішити задачу підвищення ефективності власного виробництва неможливо без отримання об'єктивної картини технічних і технологічних параметрів виробництва. Іншим фактором є

наявність інформаційних і організаційних бар'єрів між управлінськими та технологічними рівнями. Така роз'єднаність призводить до блокування важливої для аналізу діяльності підприємства інформації, а також різко знижує оперативність прийняття управлінських рішень. І нарешті ринок засобів і систем автоматизації володіє всіма необхідними продуктами в області відкритих стандартних рішень для систем комплексної інтеграції. Без використання цих розробок неможливо вирішувати завдання побудови інформаційно прозорого підприємства. Типовий комплекс АСОДК призначений для організації збору, обробки, уявлення та архівування даних, що надходять з різних цехів і ділянок підприємства та різних підсистем АСК ТП / АСКО. База технологічних даних служить джерелом оперативних зведень для центрального диспетчерського пульта, головних фахівців заводу, керівництва підприємства. АСОДК представляє собою відкриту систему з підтримкою стандартних міжсистемних інтерфейсів, що дозволяють вирішувати завдання інтеграції підсистем АСК ТП / АСКО з різними системами управління бізнес-процесами підприємства (напр. SAP / R3, BAAN, Галактика і т.д.). Впровадження на підприємстві АСОДК забезпечує досягнення таких цілей:

- підвищення ефективності керування підприємством і окремими структурними підрозділами на основі оперативної та достовірної інформації;
- створення інформаційного інструменту для виявлення та використання джерел економії виробничих і невиробничих витрат та створення основи для прийняття управлінських рішень високої якості;
- зниження впливу людського фактора при підготовці і прийнятті управлінських рішень.

У загальному вигляді основні програмно-технічні рівні АСОДК більшості підприємств можна представити у вигляді моделі (рис.1.1).

Проаналізуємо основні рівні АСОДК, у відповідності з рис. 1.1. він складається з наступних.

- Рівень додатків інформаційної системи підприємства – цей рівень представлений набором відкритих інтерфейсів і програмних шлюзів передачі агрегованої технологічної інформації від оперативно-технологічного сервера АСК ТП в АСК підприємства. Для кожного підприємства цей набір свій.

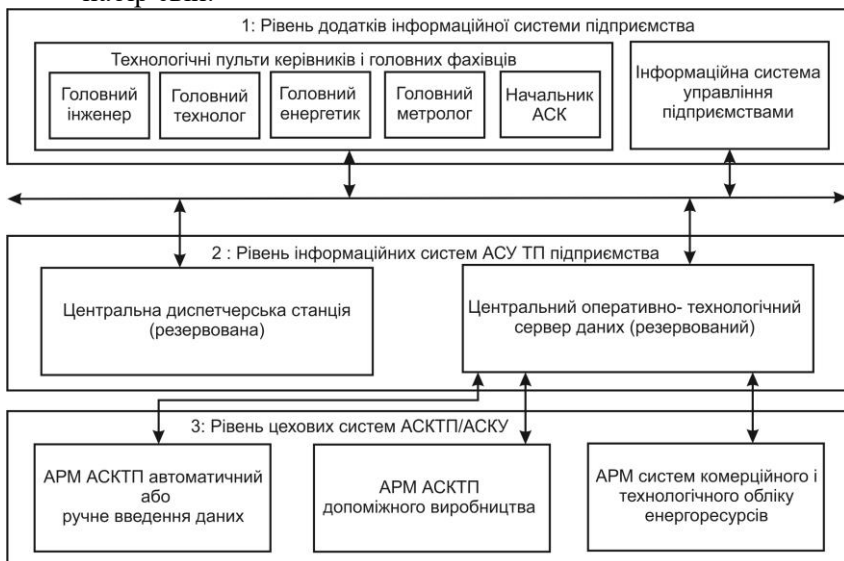


Рисунок 1.1 – Програмно-технічні рівні АСОДК

- Рівень інформаційної системи АСК ТП підприємства – цей рівень представлений двома апаратно-програмними комплексами: центральна диспетчерська станція й оперативно-технологічний сервер даних АСК ТП. Центральна диспетчерська станція є робочим місцем диспетчера (АРМ оператора АСОДК), яке виконує основне завдання з відображення оперативної технологічної інформації, що надходить від цехів підприємства, і централізований диспетчерський контроль

за виробничими та технологічними процесами. Оперативно-технологічний сервер АСК ТП призначений для збору і зберігання технологічної інформації, а також забезпечує загальний хід виробничих робіт.

- Рівень цехових систем АСК ТП / АСКО. Інформаційний рівень цехів підприємства містить підсистеми автоматизації, що існують, та системи введення технологічних даних (автоматичне або ручне введення). Прикладне програмне забезпечення цього рівня вирішує завдання введення технологічних параметрів з даного цеху / дільниці, забезпечення локального архіву введеної інформації та передачі через канали зв'язку отриманих даних на інформаційний рівень АСК ТП.

Всі типові компоненти АСОДК можна розділити на такі

групи:

- АСК ТП основного і допоміжного виробництва;
- АРМ цеху і заводууправління;
- автоматизовані робочий місця заводууправління;
- АРМ систем комерційного та технологічного обліку енергоресурсів;
- оперативно-технологічний сервер даних.

Реалізація АСОДК може здійснюватися поетапно, оскільки її повнота і обсяг визначаються конкретними потребами та можливостями підприємства. При цьому в основі створення АСОДК підприємства повинні бути покладені два основних принципи:

- при створенні АСОДК необхідно дотримуватися корпоративних і міжнародних стандартів в галузі використання інструментальних засобів та технологій інтеграції підсистем автоматизації;
- інформація на підприємстві повинна розглядатися як найважливіший виробничий ресурс.

Клас 3: інформаційна автоматизована система керування підприємства (ІАСК підприємства) покликана забезпечити процес безперервного і цілеспрямованого збору, перетворення і

подання інформативних показників, необхідних для проведення аналізу, планування, підготовки та реалізації управлінських рішень на основі оперативної технологічної інформації на рівні підприємства.

Основу ІАСК складають продукти і рішення інтеграційного шару, які повинні забезпечити отримання технологічних даних, їх передачу на рівень системи управління підприємством і видачу керуючих впливів на технологічний рівень.

Загальний хід робіт зі створення ІАСК можна розбити на кілька етапів. Кожне підприємство самостійно деталізує кожен етап в залежності від його готовності і переліку завдань, поточного рівня автоматизації і власних можливостей.

Етап 1: рішення облікових завдань. З появою реального власника підприємства на перше місце висуваються завдання зі створення системи комерційного (для зв'язку з податковими та генеруючими компаніями) і технологічного (межцехового) обліку енергоресурсів. Сюди включаються системи обліку електро-енергії, тепла, води, газу, пари і т.д. Всі ці підсистеми повинні бути сертифіковані органами Держстандарту, виконані на базі сучасних комп'ютеризованих засобів і мати відкриті комунікаційні інтерфейси. Це необхідна умова для подальшого створення інтегрованого інтелектуального, прозорого, ефективного підприємства.

Етап 2: горизонтальна інтеграція підсистем автоматизації. Кожне підприємство відрізняється ступенем автоматизації своїх основних і допоміжних виробництв. Однак завдання узагальнення різноманітних технологічних, облікових, комерційних даних є сьогодні загальним для всіх. Дане завдання може бути вирішене в рамках створення оперативно-технологічного сервера даних, який буде об'єднувати всі потоки даних з підсистем АСК ТП / АСКО і служити джерелом для інформаційних програм підприємства.

Етап 3: вертикальна інтеграція підсистем. При створенні комплексної системи автоматизації горизонтальна інтеграція

повинна розглядатися як фундамент для забезпечення виконання вертикальної інтеграції.

Вертикальна інтеграція включає в себе такі послідовні кроки:

- інтеграція оперативно-технологічної бази даних реального часу (оперативно-технологічний сервер АСК ТП) з корпоративною базою даних, наприклад Oracle;
- інтеграція з автоматизованою системою управління бізнес-процесами підприємства (напр. SAP / R3, BAAN, Галактика і т.д.);
- створення або інтеграція з системою управління, що існує, активами (основними фондами) підприємства.

Основні інструменти створення комплексної ІАСК. Інтенсивний розвиток виробництва та створення сучасних систем керування неможливо без знання і впровадження сучасних технічних та програмних засобів. Це актуально і для промисловості, що знаходиться на шляху модернізації основного обладнання, так і для галузей, які виробляють системи оборонного, спеціального призначення і вбудовані системи автоматизації. До базових компонентів інтегрованої ІАСК відносяться такі продукти і класи продуктів, які представлені на рис. 1.2.

Галузеві рішення

ПТК «Інтегратор»	ІАСУ підприємства	ПТК «Диспетчер»	Оперативно-технічний сервер даних	ПТК «Машиніст»	АСОДУ підприємства
БД реального часу	Web рішення і системи	MES - системи	EAM - системи	ERP- системи	SCADA- системи
Промислові мережі (PROFIBUS, ControlNET)		Industrial Ethernet		Ethernet	
Інтелектуальні датчики; Промислові контролери;					

Рисунок 1.2 – Базові компоненти інтегрованої ІАСК

Апаратні засоби інтеграції:

- апаратні засоби для інтеграційних серверів (промислові комп'ютери та контролери), що відповідають вимогам підвищеної надійності та стійкості для промислових умов експлуатації. Такі вимоги, як відкрита стандартна архітектура, апаратна масштабованість, гаряча заміна комунікаційних модулів, резервування є також особливостями цих апаратних засобів. Всі перераховані вимоги на сьогоднішній день задовольняють дві основні апаратні платформи: VME і CompactPCI;
- інтелектуальні кошти низової автоматики, що мають крім вбудованих сервісів первинної обробки та WEB-сервери для дистанційної діагностики та налагодження.

Програмні засоби інтеграції:

- SCADA-системи, призначені для створення АРМ фахівців. Вони мають підтримку стандартних протоколів обміну (OPC, SQL, DDE, OLE DB) як з технологічними системами АСК ТП, так і з бізнес-додатками;
- бази даних реального часу, що служать основою створення оперативно-технологічних серверів АСК ТП, які, в свою чергу, є джерелами даних для бізнес-додатків рівня підприємства;
- системи керування виробництвом (MES-системи), що забезпечують регулювання процесу руху матеріалів від сировини до кінцевого продукту;
- системи керування основними фондами (ЕАМ-системи), що підтримують в робочому стані верстатний парк і інше виробниче обладнання.

Комунікаційні засоби інтеграції:

- мережа Ethernet (Ethernet, Industrial Ethernet, Radio Ethernet) для зв'язку оперативно-технологічного сервера АСК ТП, АРМ операторів цехів і дільниць, підсистем

обліку енергоресурсів, центральної диспетчерської і АСК підприємства;

- промислові мережі (fieldbus) для організації зв'язку інтелектуальних давачів і механізмів, локальних (цехових) підсистем автоматизації, оперативно-технологічного сервера АСК ТП. Основними прикладними протоколами є Profibus, MODBUS, TCP / IP-MODBUS, Fieldbus Foundation.

1.2. Класифікація сучасних АСК ТП

Сучасна класифікація АСК ТП відповідно до міжнародних стандартів і на базі аналізу зарубіжної літератури всі АСК ТП ділить на три глобальних класи:

- SCADA (Supervisory Control and Data Acquisition) перекладається як «Система диспетчерського керування». Термін SCADA часто використовується в більш вузькому сенсі: багато так називають програмний пакет візуалізації технологічного процесу. Однак в даній книзі під словом SCADA ми будемо розуміти цілий клас систем управління, про які йтиме мова, нижче на прикладі TRACE MODE 6.0;
- PLC (Programmable Logic Controller) перекладається як «програмований логічний контролер» (або скорочено ПЛК);
- електронна складова промислового контролера, спеціалізованого пристрою, що використовується для автоматизації технологічних процесів.

В якості основного режиму роботи програмованих логічних контролерів, найчастіше в несприятливих умовах, виступає його автономне використання, без серйозного обслуговування і практично без втручання людини.

У системах керування технологічними об'єктами логічні команди переважають над числовими операціями, що дозволяє при порівняльній простоті мікроконтролера (з восьми- або

шістнадцяти бітової шиною), отримати потужні системи, що діють в режимі реального часу.

В сучасних програмованих логічних контролерах числові операції реалізуються нарівні з логічними. У той же час, на відміну від більшості комп'ютерних процесорів, в програмованих логічних контролерів забезпечується доступ до окремих бітам пам'яті.

Як правило, програмовані логічні контролери в своєму складі не мають інтерфейсу для людини, типу клавіатури і дисплея. Їх програмування, діагностика та обслуговування проводиться програматорами, що спеціально підключаються, – спеціальними пристроями або пристроями на базі більш сучасних технологій – персонального комп'ютера або ноутбука, зі спеціальними інтерфейсами та зі спеціальним програмним забезпеченням.

В автоматизованих системах керування технологічними процесами програмовані логічні контролери взаємодіють з різними компонентами систем людино-машинного інтерфейсу (наприклад, з операторськими панелями) або робочими місцями операторів на базі ПК, часто промислових, зазвичай через промислову мережу.

- DCS (Distributed Control System) перекладається як розподілена система управління (PCY). Це комплекс технічних і програмних рішень для побудови АСК ТП, характерною рисою якої є децентралізована обробка даних і наявність розподілених систем введення, і виведення інформації, підвищена відмовостійкість, стандартна і єдина структура бази даних.

Інтерес до розподілених систем управління виник в процесі збільшення кількості давачів, площ виробництва, модернізації й ускладнення стандартних алгоритмів керування технологічними процесами. Контролери, модулі введення і виведення даних, давачі, виконавчі механізми рознесені в просторі. Кожен ПЛК працює зі своїми давачами та виконавчими механізмами, працює з конкретною частиною

об'єкта управління, і не залежить від інших ПЛК, однак, взаємодіє з іншими контурами і пристроями для виконання спільного завдання, досягнення заданих показників якості. Ділянки Distributed Control System можуть перебувати один від одного на будь-якій відстані, а зв'язок між ними буде підтримуватися через Інтернет. Подібна тенденція почала розвиватися в зв'язку з успіхами ООП (об'єктно-орієнтованого програмування). Для досягнення оптимального режиму роботи DCS обмін даними між контролерами повинен бути зведений до мінімуму. Синхронізація ж роботи ПЛК проводиться на більш високому рівні АСК ТП

Основними вимогами до DCS є:

- відмовостійкість. Зупинка технологічного процесу для Distributed Control System є неприпустимою. Досягається це резервуванням технічної та програмної бази;
- масштабованість. Так само вкрай важлива для розподіленої системи управління, оскільки нарощування технологічної бази, додавання нових контурів і частин об'єкта управління повинні відбуватися просто і без зупинки функціонування вже підключених рішень;
- просте конфігурування та розробка DCS. До складу Distributed Control System має входити безліч необхідних елементів. Серед них: системи програмування, бібліотеки стандартних елементів, налаштовані мережеві інтерфейси і протоколи

1.3. Поняття SCADA систем і їхнє призначення

SCADA системи дозволяють проводити моніторинг процесу збору інформації реального часу з віддалених точок (об'єктів) для обробки, аналізу і можливого управління віддаленими об'єктами. Вимога обробки реального часу обумовлена необхідністю доставки (видачі) всіх необхідних подій (повідомлень) і даних на центральний інтерфейс оператора

(диспетчера). У той же час поняття реального часу відрізняється для різних SCADA систем.

Однією з основних елементів будь-SCADA системи є набір давачів, які здійснюють знімання інформації з об'єкта, зі спеціалізованих багатопроцесорних відмовостійких обчислювальних комплексів, які здійснюють обробку інформації та керування в режимі жорсткого реального часу. Прообразом сучасних систем SCADA на ранніх стадіях розвитку автоматизованих систем керування були системи телеметрії та сигналізації.

Всі сучасні SCADA системи включають три основних структурних компоненти, що представлені на рис. 1.3.

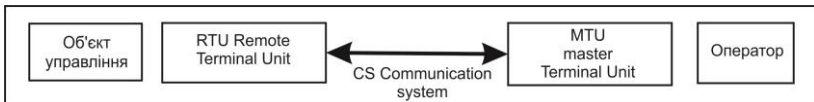


Рисунок 1.3 – Основні структурні компоненти SCADA системи

Розглянемо докладніше кожен з них:

Remote Terminal Unit (RTU) віддалений термінал, який здійснює обробку завдання (керування) в режимі реального часу. Спектр його втілень дуже широкий. Конкретна його реалізація визначається конкретним застосуванням. Використання пристроїв низькорівневої обробки інформації дозволяє знизити вимоги до пропускної здатності каналів зв'язку з центральним диспетчерським пунктом;

Master Terminal Unit (MTU), Master Station (MS) диспетчерський пункт керування (головний термінал); здійснює обробку даних і керування високого рівня, як правило, в режимі м'якого (квазі-) реального часу; одна з основних функцій – забезпечення інтерфейсу між людиною-оператором і системою (НМІ, ММІ). Залежно від конкретної системи MTU може бути реалізований в самому різноманітному вигляді від одиночного комп'ютера з додатковими пристроями підключення до каналів зв'язку, до великих обчислювальних систем (мейнфреймів) і / або об'єднаних в локальну мережу робочих станцій і серверів.

Як правило, при побудові MTU використовуються різні методи підвищення надійності та безпеки роботи системи;

Communication System (CS) комунікаційна система (канали зв'язку), необхідна для передачі даних з віддалених точок (об'єктів, терміналів) на центральний інтерфейс оператора-диспетчера і передачі сигналів керування на RTU (або віддалений об'єкт в залежності від конкретного виконання системи).

SCADA системи побудовані за принципами модульності, багатоплатформності та масштабованості.

В основі будь-якої SCADA системи лежать наступні за-гальні принципи побудови вимірювально-діагностичних систем машин і обладнання:

- принцип достатності регламентує вибір мінімального числа датчиків вторинних процесів, які супроводжують роботу машин, обладнання та технологічної системи в цілому, що забезпечують можливість спостерігати технічний стан;
- принцип інформаційної повноти відображає обмеженість наших знань про навколишній світ і в загальному вигляді може бути сформульовано так, що крім відомих нам діагностичних ознак, що описують технічний стан об'єкта відомим чином, з спектра сигналу після видалення з нього відомих ознак виділяється залишковий «шум», який також використовується для діагностики;
- принцип інваріантності регламентує вибір і селекцію таких діагностичних ознак, які інваріантні до конструкції машини та форми зв'язку з параметрами її технічного стану, що забезпечує застосування швидких рангових процедур, що самі навчаються, без еталонної діагностики та прогнозування ресурсу машин і, відповідно, швидкі темпи розробки і впровадження ІДС;
- принцип самодіагностики всіх вимірювальних і каналів, що управляють, ІДС забезпечує легкий пуск систем в експлуатацію, простоту обслуговування та ремонт

окремих каналів, високу метрологічну і функціональну надійність системи, її виживання і пристосованість до постійно мінливих умов реального виробництва. Принцип реалізується подачею спеціальних стимулюючих сигналів в ланцюг датчика, а далі комп'ютерним аналізом цього сигналу на виході системи після АЦП. Таким чином перевіряється функціонування всього тракту від давача, через виносні модулі та розгалужувачі до комп'ютерної програми та монітора;

- принцип структурної гнучкості і програмованість забезпечує реалізацію оптимальної паралельно-послідовної структури ІДС, виходячи з критеріїв необхідної швидкодії при мінімальній вартості.
- принцип дружності інтерфейсу при максимальній інформаційній ємності забезпечує сприйняття оператором стану технологічної системи в цілому при одному погляді на монітор і отримання припису, що вказує найближчі невідкладні дії;
- принцип багаторівневої організації забезпечує роботу з системою фахівцям різних рівнів кваліфікації і відповідальності, а також дозволяє задовольняти допитливість персоналу в міру підвищення його кваліфікації

Концепція SCADA зумовлена всім ходом розвитку систем управління і результатами науково-технічного прогресу. Застосування набору технологій SCADA дозволяє досягти високого рівня автоматизації у вирішенні завдань розробки систем управління, збору, обробки, передачі, зберігання і відображення інформації.

Дружність людино-машинного інтерфейсу (НМІ / ММІ), що надається SCADA системами, повнота і наочність представленої на екрані інформації, доступність «важелів» керування, зручність користування підказками і довідковою системою і т.д. – підвищує ефективність взаємодії диспетчера з системою і зводить нанівець його критичні помилки при керуванні.

Слід зазначити, що концепція SCADA, основу якої становить автоматизована розробка систем керування, дозволяє вирішити ще ряд завдань, довгий час вважалися нерозв'язними: скоротити терміни розробки проектів з автоматизації і прямі фінансові витрати на їх розробку.

Термін SCADA зазвичай відноситься до централізованих систем контролю та керування всією системою, або комплексами систем, розташованих на великих областях (між промисловою установкою і споживачем). Більшість керуючих впливів виконується автоматично RTU або ПЛК. Першорядні функції управління зазвичай обмежуються за рівнями скасування або контролюючому втручання. Наприклад, на рис. 1.4 наведена найпростіша SCADA система керування системою активного охолодження, PLC може управляти потоком води, що охолоджує всередині частини виробничого процесу, а SCADA система може дозволити операторам змінювати ставку для потоку і встановити умови сигналізації, такі як втрата потоку і висока температура, які повинні бути відображені і записані. Цикл керування зі зворотним зв'язком проходить через RTU або ПЛК, в той час як SCADA система контролює повне виконання циклу.

Збір даних починається в RTU або на рівні PLC і включає покази вимірювального приладу і звіти про відмову обладнання (тривоги), з'єднаного з SCADA, у міру потреби. Далі дані збираються та форматуються у такий спосіб, щоб оператор диспетчерської, використовуючи НМІ, міг прийняти контролюючі рішення – коригувати або перервати стандартне управління коштами RTU / ПЛК. Дані можуть також бути розміщені в Історію, часто засновану на БД, для побудови трендів та іншої аналітичної обробки накопичених даних.

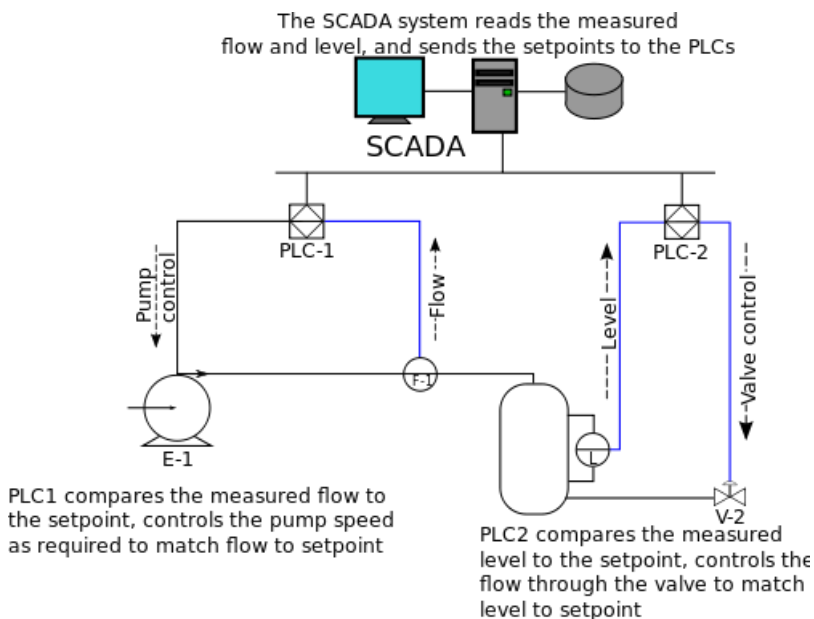


Рисунок 1.4 – Приклад реалізації управління активною системою охолодження технологічного процесу виробництва в SCADA

Системи SCADA зазвичай оснащуються розподіленою базою даних, часто званою базою даної тегів. Ця база містить елементи даних, названі тегами або точками. Точка є одиничним введенням або виведенням, значення якого контролюють або регулюють в системі. Точки можуть бути або апаратними (hard) або програмними (soft). Апаратна («hard») точка представляє собою фактичне введення або виведення в межах системи, в той час як точка «soft» – результат математичних і логічних операцій з іншими точками. (Більшість реалізацій систем знімає концептуальне розходження між «soft» і «hard» точками, роблячи кожен властивість у виразі точкою «soft», яка може, в самому простому випадку, рівнятися одиничній апаратній точці). Точки зазвичай зберігаються як пари значення-штамп_часу: значення, і штамп часу – той час, коли подію було зареєстровано або обчислено. Серія пар значення-штамп_часу є хронологію даної точки. Також поширене збереження

додаткових метаданих з тегами, такими як шлях до польового пристрою або регістра ПЛК, коментарі під час розробки, і сигнальна інформація.

Існує два типи керування віддаленими об'єктами в SCADA:

- автоматичне;
- таке, що ініціюється оператором системи.

Один із засновників SCADA Шерідан (Sheridan) виділив чотири основних функціональних компонента систем диспетчерського управління та збору даних: людина-оператор (рис. 1.5), комп'ютер взаємодії з людиною, комп'ютер взаємодії із завданням (об'єктом), завдання (об'єкт управління).

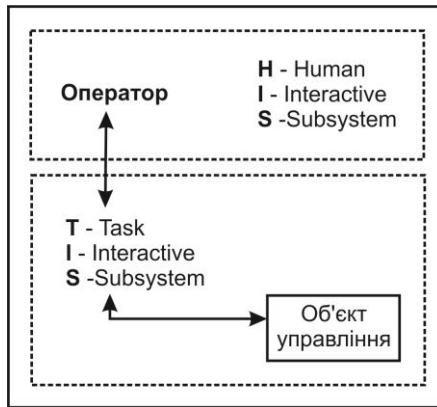


Рисунок 1.5 – Основні структурні компоненти SCADA систем

На базі цього було виділено п'ять основних функцій людини-оператора в системі диспетчерського управління і охарактеризовано їх як набір вкладених циклів, в яких оператор:

- планує, які наступні дії необхідно виконати;
- навчає (програмує) комп'ютерну систему на наступні дії;
- відстежує результати (напів) автоматичної роботи системи;

- втручається в процес у разі критичних подій, коли автоматика не може впоратися, або при необхідності підстроювання (регулювання) параметрів процесу;
- навчається в процесі роботи (отримує досвід).

Зазначене подання SCADA стало основою для розробки сучасних методологій побудови ефективних диспетчерських систем.

Особливості процесу управління в сучасних диспетчерських системах полягає у такому:

- SCADA застосовується в системах, в яких обов'язкова наявність людини (оператора, диспетчера);
- SCADA було розроблено для систем, в яких будь-який неправильний вплив може призвести до відмови (втрати) об'єкта управління або навіть катастрофічних наслідків;
- оператор несе, як правило, спільну відповідальність за управління системою, яка за нормальних умов тільки зрідка вимагає підстроювання параметрів для досягнення оптимальної продуктивності;
- активну участь оператора в процесі управління бере нечасто і в непередбачувані моменти часу, зазвичай в разі настання критичних подій (відмови, позаштатні ситуації та ін.);
- дії оператора в критичних ситуаціях можуть бути жорстко обмежені часом (декількома хвилинами або навіть секундами).

Тому до SCADA систем пред'являються такі основні вимоги:

- надійність системи (технологічна і функціональна);
- безпека управління;
- точність обробки і подання даних;
- простота розширення системи.

Вимоги безпеки і надійності управління в SCADA включають такі:

- жодна одинична відмова обладнання не повинна викликати видачу помилкового вихідного впливу (команди) на об'єкт управління;
- жодна одинична помилка оператора не повинна викликати видачу помилкового вихідного впливу (команди) на об'єкт управління;

Основними галузями застосування систем диспетчерського управління з аналізу міжнародних джерел, є:

- управління передачею і розподілом електроенергії;
- промислове виробництво;
- виробництво електроенергії;
- водозабір, водоочищення і водорозподілення;
- видобуток, транспортування і розподіл нафти та газу;
- управління космічними об'єктами;
- управління на транспорті (всі види транспорту: авіа, метро, залізничний, автомобільний, водний);
- телекомунікації;
- військова галузь.

В даний час в європейських країнах спостерігається справжній підйом модернізації автоматизованих систем управління, що існують, в різних галузях економіки; в переважній більшості випадків ці системи будуються за принципом диспетчерського управління та збору даних. Характерно, що в індустріальній сфері (в обробній і видобувній промисловості, енергетиці та ін.) найбільш часто згадується саме модернізація існуючих виробництв SCADA системами нового покоління. Ефект від впровадження нової системи управління обчислюється, в залежності від типу підприємства, від сотень тисяч до мільйонів доларів на рік. Велика увага приділяється модернізації виробництв, що представляють собою екологічну небезпеку для навколишнього середовища (хімічні та ядерні підприємства), а також відіграють ключову роль в життєзабезпеченні населених пунктів (водопровід, каналізація та ін.). З початку 2015 року в США почалися інтенсивні дослідження і розробки в галузі створення автоматизованих

систем управління наземним (автомобільним) транспортом ATMS (Advanced Traffic Management System).

1.4. Аналіз сучасних SCADA систем і вибір в залежності від поставлених задач

Сучасний бізнес в області розробки ПЗ все більш і більш сегментується і спеціалізується. Розробники операційних систем, розробники інструментальних засобів, розробники прикладного ПЗ і т.д., розмовляють різними мовами. Таким чином, сама логіка розвитку сучасного бізнесу в частині розробки ПЗ для кінцевих систем управління вимагає використання все більш розвинених інструментальних засобів типу SCADA систем. Розробка сучасної SCADA системи вимагає великих вкладень і виконується в тривалі терміни. І саме тому в більшості випадків розробникам керуючого ПЗ, зокрема ПЗ для АСК ТП, представляється доцільним йти іншим шляхом, набуваючи, засвоюючи і адаптуючи будь-якої готовий, вже випробований універсальний інструментарій.

У табл. 1.1 наведено деякі з популярних SCADA системи, які використовуються на даний момент часу, для вирішення завдань автоматизації АСК ТП.

Таблиця 1.1

Популярні SCADA системи

SCADA системи	Фірма розробник	Країна
Factory Link	United States DATA Co.	США
InTouch	Wonderware	США
Genesis	Iconocs	США
RealFlex	BJ.Software System	США
Sitex	Jade Software	Англія
FIX	Intellution	США
Trace Mode	AdAstra	Росія
IGSS	Seven Tehnologies	Данія

Image	Технолінк	Росія
RSView	Rockwell Software Inc.	США

Аналіз програмно-апаратних платформ, необхідний, оскільки від нього залежить відповідь на питання поширення SCADA системи на наявні обчислювальні засоби, а також оцінка вартості експлуатації системи (прикладна програма, розроблена в одному операційному середовищі, може виконуватися в будь-якому іншому, яку підтримує обраний SCADA пакет). У різних SCADA системах це питання вирішено по-різному. Так, FactoryLink має вельми широкий список підтримуваних програмно-апаратних платформ

У той же час в таких SCADA системах, як RealFlex і Sitex основу програмної платформи принципово становить єдина, хоча і задовольняє багатьом вимогам, операційна система реального часу QNX.

Переважає більшість SCADA систем реалізовано на платформах MS Windows. Саме такі системи пропонують найбільш повні і легко нарощувані людино-машинні інтерфейсні (Man Machine Interface MMI) засоби. З огляду на подальше посилення позицій Microsoft на ринку операційних систем (ОС) слід зазначити, що навіть розробники багатоплатформених SCADA систем, таких як United States DATA Co, пріоритетним вважають подальший розвиток своїх SCADA систем на серверних платформах типу MS Windows.

Одна з основних вимог даних систем – це автоматизація високого ступеня інтеграції. У кожній із них можуть бути задіяні об'єкти управління, виконавчі механізми, апаратура, яка реєструє і оброблює інформацію, робочі місця операторів, сервери баз даних і т.д. Узагальнена схема подібної системи приведена на рис. 1.6.

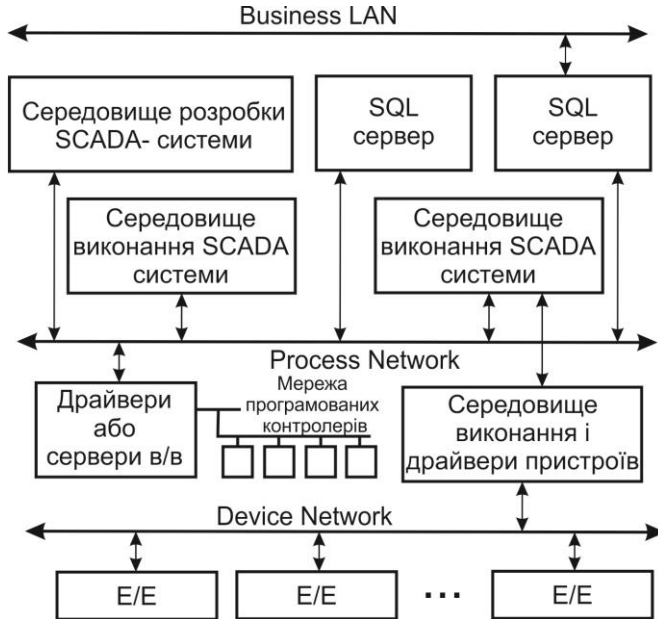


Рисунок 1.6 – Схема інтеграції SCADA додатків в комплексні системи керування

Цим вимогам в тій чи іншій мірі задовольняють практично всі розглянуті SCADA системи, з тією лише відмінністю, що набір підтримуваних мережевих інтерфейсів, звичайно ж, різний.

Більшість SCADA систем мають вбудовані мови високого рівня, VBasic подібні мови, що дозволяють згенерувати адекватну реакцію на події, пов'язані зі зміною значення змінної, з виконанням деякої логічної умови, з натисканням комбінації клавіш, а також з виконанням деякого фрагмента із заданою частотою щодо всієї програми або окремого вікна.

Практично всі SCADA системи, зокрема, Genesis, InTouch використовують синтаксис ANSI SQL, який не залежить від типу бази даних. Таким чином, додатки віртуально ізольовані, що дозволяє змінювати базу даних без серйозної зміни самої прикладної завдання, створювати незалежні програми для

аналізу інформації, використовувати вже напрацьоване програмне забезпечення, орієнтоване на обробку даних. Для фахівця-розробника системи автоматизації, так як і для фахівця-технолога, чиє робоче місце створюється, дуже важливий графічний користувальницький інтерфейс (Graphic Users Interface MMI). Функціонально графічні інтерфейси SCADA систем дуже схожі. У кожній з них існує графічний об'єктно-орієнтований редактор з певним набором анімаційних функцій. Використовувана векторна графіка дає можливість здійснювати широкий набір операцій над обраним об'єктом, а також швидко оновлювати зображення на екрані, використовуючи засоби анімації.

1.5. Контрольні питання

1. Які компоненти має містити інформаційна система?
2. На що впливає комплексна інтеграція підприємства?
3. Внаслідок чого знижується рівень витрат виробництва?
4. Яким набором чинників зумовлені роботи з комплексної інтеграції?
5. Що таке горизонтальна інтеграція?
6. Що таке вертикальна інтеграція?
7. Поясніть, що таке відкритість системи.
8. Які сучасні тенденції в галузі стандартизації ви знаєте?
9. Поясніть принцип типізації системних рішень.
10. Що таке масштабованість системи?
11. Дайте визначення комплексному підходу до систем автоматизації.
12. Дайте визначення поняттю тиражованості.
13. Наведіть три класи інформаційних автоматизованих систем керування (ІАСК) різного призначення.
14. В чому різниця між класами ІАСК?
15. Наведіть всі складові програмно-технічного рівня АСОДК.
16. Рівень інформаційної системи АСК ТП.
17. Рівень цехових систем АСК ТП/АСКО.

18. На які типові компоненти можна розділити АСОДК?
19. На які етапи можна розбити загальний хід робіт зі створення ІАСК?
20. Наведіть апаратні програмні та комунікаційні засоби інтеграції.
21. Класифікація сучасних АСК ТП та їх призначення.
22. Поняття SCADA систем та їхнє призначення.
23. Основні структурні компоненти SCADA системи та їхнє призначення.
24. Чим зумовлена концепція SCADA?
25. Наведіть приклад реалізації управління технологічного процесу виробництва в SCADA.
26. Типи керування віддаленими об'єктами в SCADA.
27. Наведіть чотири основних функціональних компоненти систем диспетчерського управління та збору даних.
28. Які сучасні SCADA систем ви знаєте для вирішення завдань автоматизації АСК ТП?
29. Наведіть приклад схеми інтеграції SCADA додатків в комплексні системи керування.
30. В чому різниця між Business LAN та Process Network?

2. СУЧАСНІ ДАВАЧІ ТА ПЕРЕГРАМОВАНІ ЛОГІЧНІ КОНТРОЛЕРИ ДЛЯ ПРОЕКТУВАННЯ АСК ТП В SCADA TRACE MODE 6

2.1. Структура автоматизованої системи керування технологічними процесами на прикладах

Автоматизовані системи керування технологічними процесами – це людино-машинні системи керування, що забезпечують збір і обробку інформації, що необхідна для оптимізації керування технологічним об'єктом відповідно до критерію якості функціонування, та реалізацію керівних впливів на технологічний об'єкт. Оператор при цьому виконує лише аналіз отриманих керівних інструкцій і їхню реалізацію. Об'єктом керування в АСКТП є технологічне обладнання, агрегати, установки, окремі виробництва – ділянки, цехи. Основні потоки інформації характеризуються таким:

- змінними, що вимірюються, сюди відносяться змінні, що вимірюються, але якими неможливо керувати, залежні від зовнішніх умов; вихідні змінні, наприклад, за якими визначається якість готових виробів і продуктів, або визначається економічність, продуктивність тощо;
- змінні, якими можна керувати, які змінюються відповідними виконавчими механізмами;
- змінні, що неможливо виміряти та керувати, до них відносяться характеристики сировини, відмова обладнання, знос інструменту тощо.

Узагальнену схему АСК ТП наведено на рис. 2.1.

Коротко роботу АСК ТП можна описати у такий спосіб: на вхід керуючого обчислювального комплексу (КОК) від давачів поступає інформація про поточні значення змінних, що характеризують хід технологічного процесу.

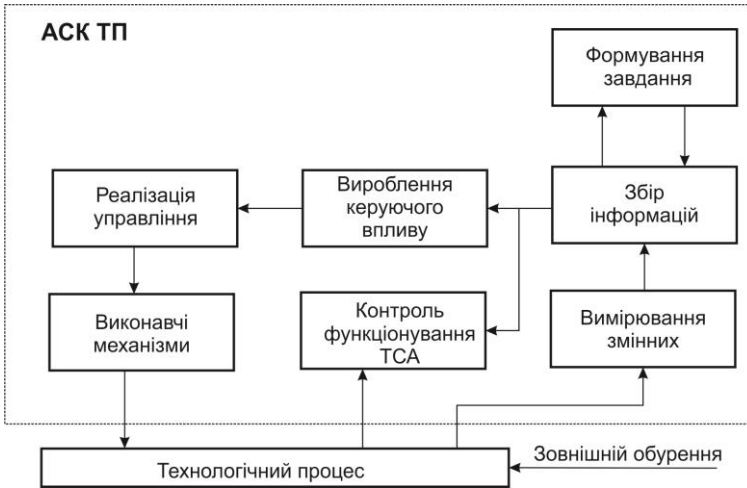


Рисунок 2.1 – Узагальнена структура АСК ТП

В КОК ця інформація оброблюється, та відповідно до прийнятого закону (алгоритму) керування визначаються керівні впливи, що мають біти прикладені до виконавчих механізмів для зміни змінних, що можна змінювати, щоб технологічний керований процес перетікає оптимальним чином. Оптимальне керування в АСК ТП визначається шляхом періодичного рішення певних формалізованих оптимізаційних задач на екстремум заданого критерію якості. Математичні моделі об'єкта керування, що використовуються при цьому, можуть неадекватно описувати поведінку останнього. Це призводить до похибок визначення критерію й оптимальних керувань. Знизити ймовірність реалізації неточних керуючих впливів дозволяє виконати аналіз результатів розв'язання оптимізаційних задач, що проводиться оператором. Важливою особливістю АСК ТП є її робота в режимі реального часу, що полягає в знаходженні оптимальних керувань за певний відрізок часу, зумовлений динамічними властивостями технологічного об'єкта керування. Інформація, що поступає ззовні, або оброблюється безпосередньо в момент поступання, або фіксується та поступає на обробку відповідно

до пріоритету, причому для кожної задачі встановлюється проміжок часу, протягом якого обробка інформації повинна бути обов'язково виконана. Все це передбачає використання в АСК ТП «швидких» методів розв'язання оптимізаційних задач. Але як правило, вони ведуть до зниження точності оптимальних керувань і вимагають проведення оператором їхнього додаткового аналізу. Інтелектуальний рівень АСК ТП визначається алгоритмами керування, що реалізуються КВК.

Всі функції АСК ТП поділяються на інформаційні, керуючі та допоміжні. Інформаційні функції: збір, обробка та подання інформації для наступної обробки. Таким чином, до інформаційних функцій відносять централізований контроль і вимірювання технологічних змінних, дотичне вимірювання, обчислення параметрів процесу, формування та видавання поточних і узагальнених технологічних і економічних показників оперативному персоналу, підготування та передача інформації до суміжних систем керування, оцінка та перевірка стану обладнання. Керуючими функціями АСК ТП є такі: регулювання технологічних змінних, логічне керування операціями або апаратами, програмне логічне керування групою обладнання, оптимальне керування встановленими або перехідними режимами, а також окремими стадіями процесу, адаптивне керування об'єктом в цілому тощо. Допоміжні функції АСК ТП полягають в забезпеченні контролю за станом функціонування технічних і програмних засобів.

АСК ТП як компонент ІАСК призначена для цілеспрямованого ведення технологічних процесів і забезпечення суміжних і тих, що стоять вище, систем управління оперативної та достовірної техніко-економічною інформацією. Залежно від розподілу функцій контролю і управління між технічними засобами в даний час розрізняють три різновиди АСК ТП: централізована, супервізорна і розподілена.

Централізована АСК ТП, структура якої зображена на рис. 2.2, характеризується тим, що всі функції обробки і управління виконує одна УОМ. Централізовані АСК ТП є комплексами, як правило, займають єдиний обмежений виробничий

простір з централізованою підсистемою забезпечення електроживленням і магістралями обміну інформаційними потоками

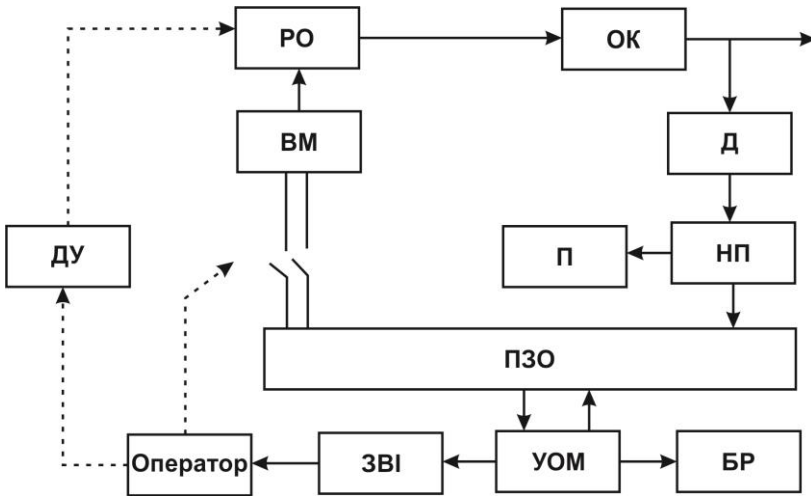


Рисунок 2.2 – Структура централізованої АСК ТП

До складу централізованої АСК ТП входять: датчик (Д), нормуючий перетворювач (НП), лінії зв'язку, пристрій зв'язку з об'єктом (ПЗО), багатоканальні реєстратори (БР), засоби відображення інформації (ЗВІ), виконавчі механізми (ВМ), пристрої дистанційного управління (ДУ), регулюючий орган (РО). УОМ під керуванням ПЗО здійснює комутацію вимірювальних каналів, перетворення аналогових сигналів в цифрові сигнали і навпаки, розподіл командних сигналів на ВМ. Для підвищення надійності подібних систем замість однієї КОМ застосовують кілька однотипних ЕОМ з власними операційними системами і міжмашинним апаратним зв'язком. Всі машини вирішують одночасно одні й ті самі завдання контролю й управління, також використовуються багатопроесорні обчислювальні комплекси (БОК), проте, вони значно підвищують вартість централізованих АСК ТП. У централізованих АСК ТП, як правило, реалізують найпростіші лінійні закони регулювання, зокрема, закони

стабілізації найменш відповідальних змінних, але під контролем оператора можуть виконувати і функцію безпосереднього цифрового управління (БЦУ).

Структура супервізорної АСК ТП складається з ряду локальних ПЗО, побудованих на базі технічних засобів індивідуального користування, і централізованої КОМ, має інформаційні лінії зв'язку з локальними системами, приклад представлений на рис. 2.3.

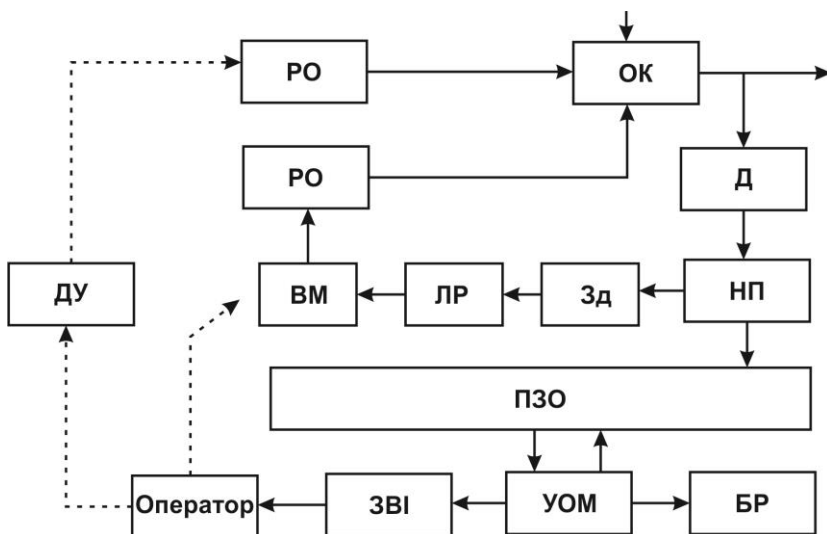


Рисунок 2.3 – Структура супервізорної АСК ТП

У супервізорних системах на УСО і УВМ покладено функції збору і переробки інформації про стан об'єкта управління, обчислення критерію якості, знаходження оптимального режиму, відповідних управлінь і передачі їх, як завдань локальним АСР, що складається з датчика (Д), що нормують перетворювач (НП), задатчика (Зд), локального регулятора (ЛР), виконавчого механізму (ВМ) і регулюючого органу (РО). Локальні АСР регулюють (стабілізують) вихідну змінну на рівні завдання, одержуваного через ПЗО від УОМ. На оператора покладаються

функції діагностики всієї АСК ТП, корекції алгоритмів контролю і управління в непередбачених ситуаціях, він також здійснює за допомогою МР і СОІ контроль за станом ОУ і аналіз оптимальних управлінь (завдань), дозволяє або забороняє їх передачу в локальні АСР, а також може безпосередньо змінювати вхідні змінні ОУ за допомогою пристроїв дистанційного керування (ДУ), або датчиків ЛР. Система супервізорного управління характеризується деякою розподіленістю функцій між центральною КОК і локальними АСР, що призводить до її надійності і живучості в порівнянні з централізованою АСК ТП.

Розподілені АСК ТП характеризуються поділом функцій контролю, обробки інформації й управління між декількома територіально розподіленими УСО і обчислювальними пристроями, з'єднаними мережами для передачі інформації та програм. Також розподілені АСК ТП будуються на базі об'єктів, розташованих на різних, деколи далеко розташованих, закритих і відкритих майданчиках. Саме ця особливість накладає певні структурні вимоги при проектуванні цих систем. Для побудови таких АСК ТП, що застосовуються для автоматизації складних ОУ, проводиться топологічно-просторова або функціонально-цільова декомпозиція управління. При топологічно-просторовій декомпозиції ОУ умовно ділиться на ряд більш простих локальних об'єктів з компактним територіальним розміщенням датчиків і регулюючих органів, що дозволяє наблизити УСО і УВМ до локальних об'єктів управління. При функціонально-цільовій декомпозиції складний ОУ підрозділяється на ряд більш простих локальних об'єктів, кожен з яких має менше число управлінь та власний критерій якості, адитивним чином входить в загальний критерій оптимальності. У цьому випадку завдання оптимізації роботи всього ОУ може бути декомпозована на ряд більш простих завдань оптимального управління локальними об'єктами і завдання координації (узгодження). Така декомпозиція задачі управління ОУ дозволяє прискорити процес знаходження оптимального управління, підвищує надійність і живучість АСК ТП. Для розподілених АСК ТП характерно також поділ функцій між

засобами і оператором, а також наявність локальних обчислювальних мереж для обміну та передачі інформації. При побудові цих систем можливі три структури: радіальна, магістральна і кільцева (рис. 2.4).

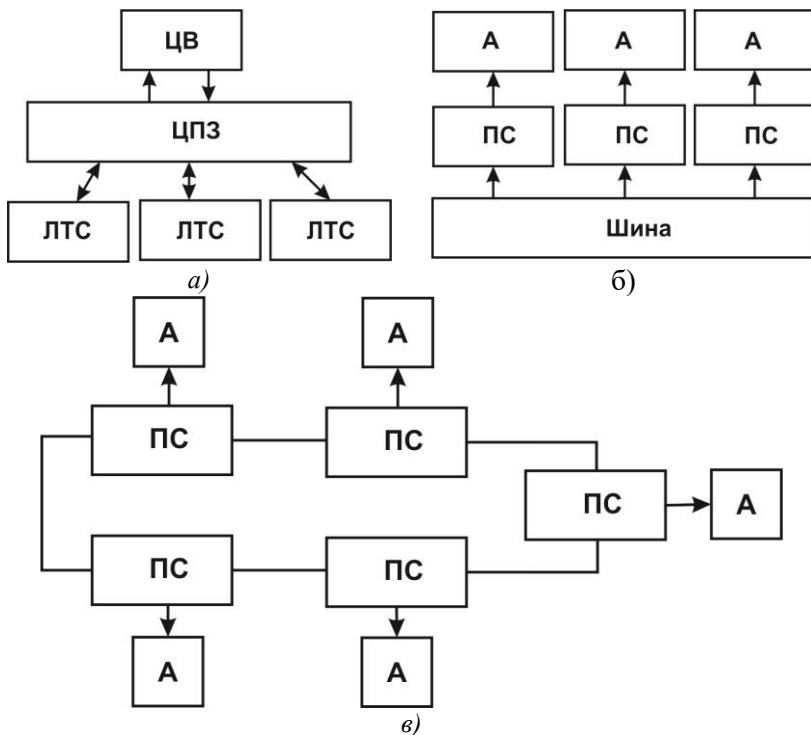


Рисунок 2.4 – Структури розподіленої АСК ТП:
а) радіальна; б) магістральна; в) кільцева.

Радіальна структура АСК ТП характеризується наявністю центрального вузла (ЦВ) і центрального пристрою зв'язку (ЦПЗ), що дозволяють приєднувати за допомогою мережевих засобів локальні технологічні станції (ЛТС), які не мають прямих інформаційних зв'язків між собою, повідомлення від однієї станції до іншої проходять через ЦВ. АСК ТП з радіальної

структурою застосовують для автоматизації складних ОК, для яких може бути проведена декомпозиція розглянутих задач на ряд локальних задач оптимізації та одну задачу координації, при цьому локальні завдання розподіляються за відповідними ЛТС, а завдання координації вирішується в центральній КОМ.

Розподілена АСК ТП з магістральною структурою характеризується наявністю єдиного зв'язку або шини, до якої за допомогою пристроїв сполучення (ПС) паралельно під'єднуються різні абоненти (А) – станції, мікроконтролери і т.п. Через шину в будь-який момент часу передається інформація тільки від одного абонента, одержувачем при цьому може бути будь-яка кількість станцій.

При кільцевій структурі розподіленої АСК ТП кожен абонент (А) підключений до пристрою сполучення (ПС), останні об'єднані один з одним шинами, утворюючи, таким чином, кільцеву мережу. Інформація циркулює цією мережею лише в одному напрямку. АСК ТП з централізованою структурою відходять у минуле. Сучасні АСК ТП будуються за принципом розподілених систем, для яких характерна світова інтеграція й уніфікація технічних рішень, відкритість системи, використання промислових обчислювальних мереж. В даний час розвиток концепції побудови автоматизованих систем керування технологічними процесами в різних галузях народного господарства відбувається під дією суперечливих факторів. Це пов'язано з тим, що принципи побудови АСК ТП регламентуються стандартами, методичними і керівними документами. У зв'язку з цим використовуються загальні та приватні принципи побудови систем управління.

Загальні принципи побудови АСК ТП базуються на узагальнених принципах побудови безлічі систем одного і того ж класу і відрізняються формулюванням головних особливостей конкретного об'єкта автоматизації, визначенням загального обсягу автоматизації, а також визначенням основних завдань управління та контролю.

Приватні принципи побудови АСК ТП враховують специфіку конкретного об'єкта автоматизації, а також головні особливості технічної реалізації, які відображаються в технічних специфікаціях на систему, що проектується. Як засоби автоматизації більшості технологічних процесів використовують програмно-технічні комплекси (ПТК), що представляють собою сукупність мікропроцесорних засобів автоматизації, дисплейних пультів оператора, серверів різного призначення, промислових мереж, які пов'язують компоненти АСК ТП, а також програмного забезпечення контролерів і дисплейних пультів оператора (рис. 2.5).

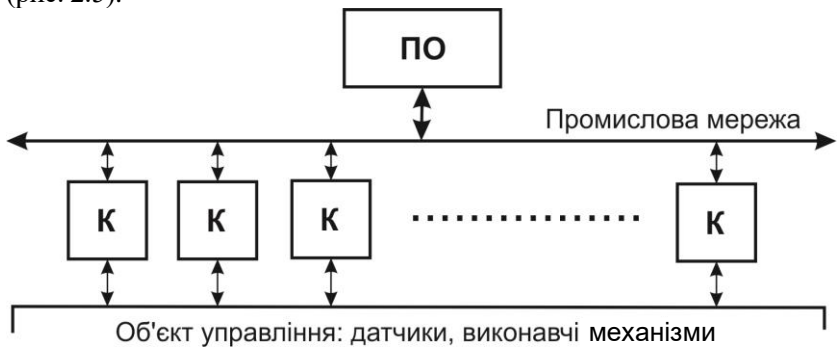


Рисунок 2.5 – Структура ПТК

Структура ПТК визначається мережевими можливостями (контролери, пульти операторів, віддаленість блоків введення-виведення), гнучкість і різноманітність залежить від числа наявних мережеских рівнів, можливих технологій на кожному рівні мережі (шина, зірка, кільце), параметрів мережі кожного рівня (типи кабелів, припустимі відстані, максимальна кількість вузлів, швидкість передачі інформації, методи доступу компонентів до мережі). Найбільш простою є структура ПТК, в якій всі функціональні можливості системи розділені на два рівні: перший – контролери, другий – пульт оператора (робоча станція або промисловий комп'ютер). Контролери (К) виконують збір інформації від датчиків, встановлених на об'єкті управління,

проводять попередню обробку сигналів (масштабування, фільтрацію), реалізацію алгоритмів управління та формування керуючих сигналів для виконавчих механізмів об'єкта управління; передачу і прийом інформації з промислової мережі.

Пульт оператора (ПО) формує мережеві запити до контролерів нижнього рівня, отримує від них оперативну інформацію про хід технологічного процесу, відображає на екрані монітора хід технологічного процесу в зручному для оператора вигляді, здійснює довгострокове зберігання динамічної інформації про хід процесу (архів), автоматично встановлює оптимальні налаштування необхідних параметрів алгоритмів управління та уставок регуляторів в контролерах нижнього рівня. Останнім часом в АСК ТП все ширше використовується архітектура ПТК «клієнт-сервер», що дозволяє підвищити ефективність і швидкість роботи всієї системи, надійність і живучість за рахунок резервування серверів, робочих станцій, а також за рахунок територіального розподілу вирішуваних завдань. Сервери виконуються на базі промислових комп'ютерів і резервуються. На практиці розрізняють сервер бази даних реального часу, сервер оперативної і архівної бази даних, сервер вводу-виводу і ін. Основними функціями сервера є: збір та обробка оперативних даних від УСО і контролерів; передача управлінь контролерам з верхнього рівня; зберігання і відображення інформації про задані змінних; уявлення необхідної інформації клієнтських робочих станцій; архівація трендів, друкованих документів, проколів подій. Багато сучасних ПТК включають станції інжинірингу, виконані на базі персональних комп'ютерів в офісному виконанні, які дозволяють здійснювати інженерне обслуговування контролерів - програмування, налагодження, а в деяких випадках і інженерне обслуговування робочих станцій. Характерною рисою сучасних ПТК є активне проникнення INTERNET-технологій на рівень промислової автоматизації, це, як правило, публікації на Web-серверах інформації про хід технологічного процесу і всіляких зведених звітів. Структуру більш складного ПТК наведено на рис. 2.6.

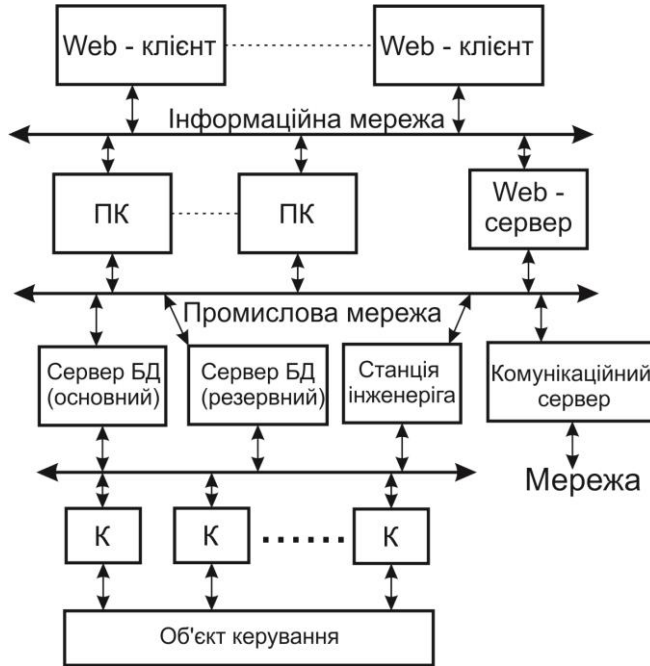


Рисунок 2.6 – Структура складного ПТК

Сучасні розподілені АСК ТП характеризуються наявністю промислової мережі, що складається з багатьох складових, обмін між якими проводиться цифровим способом. Використання промислової мережі дозволяє розташувати вузли, в якості яких виступають контролери та інтелектуальні пристрої введення-виведення, максимально наблизивши їх до датчиків і виконавчих механізмів, скоротивши до мінімуму довжину аналогових ліній.

Кожен вузол промислової мережі виконує такі функції, як прийом команд і даних від інших вузлів промислової мережі, зчитування даних з підключенням датчиків, перетворення отриманих даних в цифрову форму, відпрацювання запрограмованого технологічного алгоритму, видачу керуючих впливів на підключення виконавчих механізмів по команді іншого вузла або згідно з технологічним алгоритмом, передачу інформації на інші

вузли мережі. АСК ТП на базі промислових мереж дозволяють істотно економити кабельну продукцію, підвищити надійність системи управління, гнучкість і придатність до модифікації, використовувати принципи відкритих систем. Найбільш поширеними промисловими мережами, застосовуваними в системах автоматизації, є Modbus, PROFIBUS, Intabus, Fieldbus, Ethernet і ін. З метою взаємодії відкритих систем з різними видами обчислювального обладнання і стандартах протоколів запропонована «Описова модель взаємозв'язку відкритих систем (OSI-модель, ISO / OSI Model)». Відповідно до моделі ISO | OSI мережеві функції розподіляють на сім рівнів: фізичний, каналний, мережевий, транспортний, сеансовий, рівень уявлення, прикладний. На фізичному рівні визначаються фізичні характеристики каналу зв'язку і параметри сигналів. Канальний рівень визначає правила спільного використання фізичного рівня вузлами мережі. Мережевий рівень відповідає за адресацію і доставку пакета за оптимальним маршрутом. Транспортний рівень розбирається з вмістом пакетів, робить розподіл і складання пакетів. Сеансовий рівень координує взаємодію між вузлами мережі. Рівень представлення займається при необхідності перетворенням формату даних. Прикладний рівень забезпечує безпосередню підтримку прикладних процесів і програм кінцевого користувача, а також управління взаємодією цих програм з різними об'єктами мережі передачі даних. Більшість промислових мереж мають три рівні: фізичний, каналний, прикладний. Вибір типу промислової мережі визначається рівнем автоматизації, місцем мережі в ієрархії промислового підприємства, вимогами до її функціональних характеристик. Для ефективної роботи АСК ТП необхідно реалізувати такі види забезпечення:

1. Оперативний персонал. До складу оперативного персоналу входять технологи-оператори автоматизованого технологічного комплексу, які здійснюють управління технологічним об'єктом, і експлуатаційний персонал АСК ТП, що забезпечує функціонування системи. Оперативний персонал може працювати в контурі управління і поза ним. У першому випадку реалізуються

функції управління за рекомендаціями, що видаються КТС. У другому випадку оперативний персонал задає системі режим роботи, контролює роботу системи і при необхідності приймає на себе управління технологічним об'єктом.

2. Організаційне забезпечення включає опис функціональної, технічної, організаційної структур системи, інструкції та регламенти для оперативного персоналу, для роботи АСК ТП. Воно містить сукупність правил, приписів, що забезпечують необхідну взаємодію оперативного персоналу між собою і комплексом засобів. Таким чином, організаційна структура управління – це зв'язки між людьми, зайнятими експлуатацією об'єкта. Персонал, зайнятий оперативним управлінням, підтримує технологічний процес в заданих нормах, забезпечує виконання виробничого плану, контролює роботу технологічного обладнання, стежить за умовами безпечного ведення процесу. Експлуатаційний персонал АСК ТП забезпечує правильність функціонування КТС АСК ТП, веде облік і звітність. АСК ТП отримує від вищого рівня управління виробничі завдання, критерії реалізації цих завдань, передає на вищі рівні управління відомості про виконання завдань, кількісні й якісні показники продукції і функціонування автоматизованого технологічного комплексу.

Для аналізу організаційної структури та визначення оптимальної побудови внутрішніх взаємозв'язків використовують методи групової динаміки. При цьому зазвичай застосовують методику і прийоми соціальної психології. Проведені дослідження дали можливість сформулювати вимоги, необхідні для організації групи оперативного технологічного персоналу:

- вся виробнича інформація повинна передаватися тільки через керівника;
- у одного підлеглого має бути не більше одного безпосереднього керівника;
- у виробничому циклі інформаційно взаємодіють один з одним тільки підлеглі одного керівника.

Підрозділи технічного обслуговування виконують роботи на всіх стадіях створення АСК ТП (проектування, впровадження, експлуатація), їх основними функціями є:

- забезпечення експлуатації систем відповідно до правил і вимог технічної документації;
- забезпечення поточного і планового ремонту технічних засобів АСК ТП;
- проведення спільно з розробниками випробувань АСК ТП;
- проведення досліджень з визначення економічної ефективності системи;
- розробка та реалізація заходів щодо подальшого розвитку системи;
- підвищення кваліфікації працівників служби АСК ТП, вивчення й узагальнення досвіду експлуатації.

Для виконання функцій технологу-оператора необхідно надати технічні та програмні засоби, що забезпечують в залежності від особливостей технологічного процесу необхідні набори з таких інформаційних повідомлень:

- індикація вимірних значень параметрів за викликом;
- індикація та зміна заданих меж контролю параметрів процесу;
- звукова сигналізація й індикація відхилень параметрів за регламентні кордони;
- звукова сигналізація й індикація відхилень швидкості зміни параметрів від заданих значень;
- відображення стану технологічного процесу і обладнання на схемі об'єкта управління;
- реєстрація тенденцій зміни параметрів;
- оперативна реєстрація порушень технологічного процесу і дій оператора.

3. Інформаційне забезпечення (ІЗ) включає систему кодування технологічної та техніко-економічної інформації, довідкову і оперативну інформацію, містить опис всіх сигналів і кодів, які використовуються для зв'язку технічних засобів. Застосовувані

коди повинні включати мінімальне число знаків, мати логічну структуру і відповідати іншим вимогам кодування.

Форми вихідних документів і подань інформації не повинні викликати труднощів при їх використанні. При розробці і впровадженні системи ІЗ АСК ТП необхідно враховувати принципи організації управління технологічним процесом, яким відповідають такі етапи:

- визначення підсистем АСК ТП і типів управлінських рішень, за якими необхідно забезпечення науково-технічною інформацією. Результати цього етапу використовуються для визначення оптимальної структури масивів інформації, для виявлення характеристик очікуваного потоку запитів;
- визначення основних груп споживачів інформації. Споживачі інформації класифікуються залежно від їх участі в підготовці та прийнятті управлінських рішень, пов'язаних з організацією технологічного процесу. Накопичення інформації здійснюється з урахуванням видів завдань, що вирішуються при управлінні процесами. Споживач може отримати інформацію про зв'язані технологічні ділянки, також створюються умови для перерозподілу інформації при зміні потреб;
- вивчення інформаційних потреб;
- вивчення потоків науково-технічної інформації, необхідної при управлінні процесами, базується на результатах аналізу управлінських завдань. Поряд з потоками документальної інформації аналізуються факти, що відображають досвід даного й аналогічних підприємств;
- розробка інформаційно-пошукових систем для управління технологічним процесом.

Для автоматизованих систем характерні процеси обробки інформації – перетворення, передача, зберігання, сприйняття. При управлінні технологічним процесом відбувається передача інформації та переробка керуючою системою вхідної інформації у вихідну інформацію. При цьому необхідні контроль і регулю-

вання, які полягають в порівнянні інформації про результати попереднього етапу діяльності з інформацією, що відповідає умовам досягнення мети, в оцінці неузгодженості між ними і виробленні коригуючого вихідного сигналу. Неузгодженість викликається внутрішніми та зовнішніми збуджуючими впливами випадкового характеру. Процес передавання інформації передбачає наявність джерела інформації і приймача. Для забезпечення участі людини в управлінні технологічним процесом необхідне документування інформації. Для подальших аналізів потрібно накопичення статистичних вихідних даних за допомогою реєстрації станів і значень параметрів процесу в часі. На основі цього перевіряється дотримання технологічного процесу, якість продукції, контролюються дії персоналу в аварійних ситуаціях, здійснюється пошук напрямків вдосконалення процесу. При розробці інформаційного забезпечення АСК ТП, пов'язаного з документуванням і реєстрацією, необхідно:

- визначити вид реєстрованих параметрів, місце і форму реєстрації;
- вибрати часовий чинник реєстрації;
- мінімізувати кількість реєстрованих параметрів виходячи з необхідності та достатності для оперативних дій і аналізу;
- уніфікувати формати документів, їх структуру;
- ввести спеціальні реквізити;
- вирішити питання класифікації документів і маршрутів їх руху;
- визначити обсяги інформації в документах, встановити місце і терміни зберігання документів.

Потоки інформації в каналах зв'язку АСК ТП система повинна передавати з необхідною якістю інформації від місця її утворення до місця її прийому та використання. Для цього повинні задовольнятися такі вимоги:

- своєчасність доставки інформації;
- правильність передачі – відсутність спотворень, втрат;
- надійність функціонування;
- єдність часу в системі;

- можливість технічної реалізації;
- забезпечення економічної прийнятності інформаційних вимог.

Крім того, система повинна передбачати:

- регулювання інформаційних потоків;
- можливість здійснення зовнішніх зв'язків;
- можливість розширення АСК ТП;
- зручність участі людини в аналізі й управлінні процесом.

До основних характеристик потоку інформації відносяться:

- об'єкт управління (джерело інформації);
- мета інформації;
- формат інформації;
- об'ємно-часові характеристики потоку;
- періодичність виникнення інформації;
- об'єкт, який використовує інформацію.

При необхідності характеристики потоку деталізуються зазначенням:

- виду інформації;
- найменування контрольованого параметра;
- діапазону зміни параметра в часі;
- числа однойменних параметрів на об'єкті;
- умов відображення інформації;
- швидкості генерації інформації.

До основних інформаційних характеристик каналу зв'язку відносяться:

- місце розташування початку і кінця каналу зв'язку;
- форма переданої інформації;
- структура каналу передачі – давач, кодер, модулятор, лінія зв'язку, демодулятор, декодер, пристрій відображення;
- вид каналу зв'язку – телефонний, механічний;
- швидкість передачі й обсяг інформації;
- способи перетворення інформації;
- пропускна здатність каналу;
- обсяг сигналу і ємність каналу зв'язку;
- стійкість;

- інформаційна та апаратурна надмірність каналу;
- надійність зв'язку та передачі через канал;
- рівень загасання сигналу в каналі;
- інформаційне узгодження ланок каналу;
- мобільність каналу передачі.

В АСК ТП може бути внесена тимчасова ознака інформації, яка передбачає єдину систему часу з централізованою шкалою відліку. Для інформаційних зв'язків АСК ТП характерною рисою є дія в реальному масштабі часу. Застосування єдиної системи відліку часу забезпечує виконання таких завдань:

- документування часу прийому, передачі інформації;
- протоколювання відбуваються в АСК ТП подій;
- аналіз виробничих ситуацій з тимчасовою ознакою (черговість надходження, тривалість);
- облік часу проходження інформації через канали зв'язку і час обробки інформації;
- управління черговою прийому, передачі, обробки інформації;
- завдання послідовності дій, що управляють в межах єдиної шкали часу;
- відображення єдиного часу в межах зони дії АСК ТП.

При створенні АСК ТП основна увага приділяється сигналам, пов'язаним із взаємодією окремих елементів. Вивченню підлягають сигнали взаємодії людини з технічними засобами, а також одних технічних засобів з іншими технічними засобами. У зв'язку з цим розглядаються такі групи сигналів і кодів.

Перша група – це стилізовані мови, які забезпечують економічне введення даних в технічні засоби і виведення їх оператору. За характером інформації виділяють технічні та економічні дані.

Друга група вирішує завдання передачі даних і стикування технічних засобів. Тут основною проблемою є правильність передачі повідомлення, для чого використовують перешкодостійкі коди. Інформаційна сумісність технічних засобів забезпечується

установкою додаткової апаратури для узгодження, використан-
ням допоміжних програм перекодування даних.

Третя група – це машинні мови. Зазвичай використовують
двійкові коди з елементами захисту даних за цифровим модулем,
з доповненням коду перевірочним розрядом.

Загальні технічні вимоги до АСК ТП з інформаційного
забезпечення:

- максимальне спрощення кодування інформації за рахунок
кодових позначень і кодів повторення;
- забезпечення простоти декодування вихідних документів і
форм;
- інформаційна сумісність АСК ТП із суміжними системами
за змістом, кодування, формою представлення інформації;
- можливість внесення змін в раніше передану інформацію;
- забезпечення надійності виконання системою своїх функ-
цій за рахунок перешкодозахищеності інформації.

Персонал АСК ТП взаємодіє з КТС, сприймаючи і вводячи
технологічну й економічну інформацію. Крім цього оператор
взаємодіє з іншими операторами та вищим персоналом. Для по-
легшення цих зв'язків використовуються заходи з формалізації
потоків інформації, їх стиснення та впорядкування. ЕОМ пере-
дає оператору інформацію у вигляді світлових сигналів, зобра-
жень, друкованих документів, звукових сигналів. При взаємодії
оператора з ЕОМ необхідно забезпечити:

- наочним відображенням функціонально-технологічної
схеми об'єкта управління, інформацію про його стан в
обсязі функцій, покладених на оператора;
- відображенням зв'язку та характеру взаємодії об'єкта
управління з зовнішнім середовищем;
- сигналізацією про порушення в роботі об'єкта;
- швидким виявленням і ліквідацією несправностей.

Окремі групи елементів, найбільш суттєві для контролю й
управління об'єктом, зазвичай виділяють розмірами, формою,
кольором.

Технічні засоби, що використовуються для автоматизації управління, дозволяють вводити інформацію тільки в певній заздалегідь обумовленій формі. Це призводить до необхідності кодування інформації. Обмін даними між функціональними блоками системи управління повинен здійснюватися закінченими змістовими повідомленнями. Повідомлення передаються двома розділними потоками даних: інформаційних і керуючих. Сигнали інформаційного потоку поділяються на групи:

- вимірюваного параметра;
- допустимі межі;
- стану функціональних блоків системи;
- адреси (приналежність вимірюваного параметра певного блоку);
- часу;
- службовий.

Для захисту від помилок при обміні інформацією через канали зв'язку на вході і виході апаратури слід використовувати надлишкові коди з їх перевіркою на парність, циклічність, ітеративність, повторюваність. Питання захисту інформації пов'язані із забезпеченням надійності роботи системи управління, формами подання інформації. Інформацію необхідно захищати від спотворення та використання її не за призначенням. Методи захисту інформації залежать від вироблених операцій, від використовуваного обладнання.

4. Програмне забезпечення. До складу програмного забезпечення входять загальне програмне забезпечення, що постачається із засобами обчислювальної техніки, в тому числі, які організують програми, програми-диспетчери, які транслюють програми, операційні системи, бібліотеки стандартних програм, а також спеціальне програмне забезпечення, яке реалізує функції конкретної системи, забезпечує функціонування КТС, в тому числі апаратним шляхом.

5. Математичне, алгоритмічне забезпечення. Як відомо, модель – це образ об'єкту дослідження, що відображає істотні властивості, характеристики, параметри, взаємозв'язки об'єкта.

Одним з методів дослідження процесів або явищ в АСУ ТП є метод математичного моделювання, тобто шляхом побудови їх математичних моделей і аналізу цих моделей. Різновидом математичного моделювання є імітаційне моделювання, при якому використовується пряма підстановка чисел, що імітують зовнішні впливи, параметри і змінні процесів за допомогою ЕОМ. Для проведення імітаційних досліджень необхідно розробити алгоритм. Алгоритми, що використовуються в АСУ ТП, характеризуються такими особливостями:

- тимчасовий зв'язок алгоритму з керованим процесом;
- зберігання робочих програм в оперативній пам'яті ЕОМ для доступу до них в будь-який момент часу;
- перевищення питомої ваги логічних операцій;
- поділ алгоритмів на функціональні частини;
- реалізація на ЕОМ алгоритмів в режимі поділу часу. Облік тимчасового чинника в алгоритмах управління зводиться до необхідності фіксації часу прийому інформації в систему, часу видачі повідомлень оператором для формування керуючих впливів, прогнозування стану об'єкта управління. Необхідно забезпечити своєчасну обробку сигналів ЕОМ, пов'язаної з керованим об'єктом. Це досягається складанням найбільш ефективних за швидкістю алгоритмів, що реалізуються на швидкодіючих ЕОМ. З другої особливості алгоритмів АСК ТП випливають жорсткі вимоги до обсягу пам'яті, необхідної для реалізації алгоритму, до пов'язаності алгоритму. Третя особливість алгоритмів обумовлена тим, що технологічні процеси керуються на основі рішень, прийнятих за результатами зіставлення різних подій, порівняння значень параметрів об'єкта, перевірки виконання різних умов і обмежень.

Використання четвертої особливості алгоритмів АСК ТП дає можливість розробнику сформулювати кілька завдань системи, а потім об'єднати розроблені алгоритми цих завдань в єдину систему. Ступінь взаємозв'язку завдань АСК ТП може бути різною і залежить від конкретного об'єкта управління.

Для обліку п'ятої особливості алгоритмів управління необхідно розробити операційні системи реального часу і планувати черговість завантаження модулів, що реалізують алгоритми задач АСК ТП, їх виконання в залежності від пріоритетів. На етапі розробки АСК ТП створюються вимірювальні інформаційні системи, які забезпечують повний і своєчасний контроль режиму роботи агрегатів, що дозволяють аналізувати хід технологічного процесу і прискорити вирішення завдань оптимального управління. Функції систем централізованого контролю зводяться до вирішення таких завдань:

- визначення поточних і прогнозованих значень величин;
- визначення показників, що залежать від ряду вимірюваних величин;
- виявлення подій, що є порушеннями та несправностями на виробництві.

Загальна модель задачі при оцінці поточних значень вимірюваних величин і який обчислюється за ним ТЕП в системі централізованого контролю може бути представлена таким чином: задається сукупність величин і показників, які необхідно визначати в об'єкті контролю, вказується необхідна точність їх оцінки, є сукупність давачів, які встановлені на об'єкті, що автоматизується. Тоді загальна задача оцінки значення окремої величини формулюється так: для кожної окремої величини потрібно знайти групу давачів, частоту їх опитування і алгоритм переробки одержуваних від них сигналів, в результаті якого значення цієї величини визначається із заданою точністю. Для вирішення завдань в умовах АСК ТП використовуються такі математичні методи, як лінійне програмування, динамічне програмування, методи оптимізації, опукле програмування, комбінаторне програмування, нелінійне програмування. Методами побудови математичного опису об'єкта є метод Монте-Карло, математична статистика, теорія планування експерименту, теорія масового обслуговування, теорія графів, системи алгебраїчних і диференціальних рівнянь.

б. Технічне забезпечення. Комплекс технічних засобів повинен забезпечити таку систему вимірювань в умовах функціонування АСК ТП, які, в свою чергу, забезпечують необхідну точність, швидкодію, чутливість і надійність відповідно до заданих метрологічних, експлуатаційних та економічних характеристик. Технічні засоби можна групувати за експлуатаційними характеристиками, функціями управління, інформаційними характеристиками, конструктивною подібністю. Найбільш зручно вважається класифікація технічних засобів за інформаційними характеристиками. Виходячи з вищеприведеного комплекс технічних засобів повинен містити:

- засоби отримання інформації про стан об'єкта управління і засоби введення в систему (вхідні перетворювачі, давачі), що забезпечують перетворення вхідної інформації в стандартні сигнали та коди;
- засоби проміжного перетворення інформації, що забезпечують взаємозв'язок між пристроями з різними сигналами;
- вихідні перетворювачі, засоби виведення інформації та управління, що перетворюють машинну інформацію в різні форми, необхідні для управління технологічним процесом;
- способи формування і передачі інформації, що забезпечують переміщення інформації в просторі;
- засоби фіксації інформації, що забезпечують переміщення інформації в часі;
- засоби переробки інформації;
- засоби локального регулювання і управління;
- засоби обчислювальної техніки;
- засоби представлення інформації оперативному персоналу;
- виконавчі пристрої;
- засоби передачі інформації в суміжні АСК і АСК інших рівнів;
- прилади, пристрої для налагодження і перевірки працездатності системи;

- документаційна техніка, що включає засоби створення і знищення документів;
- конторсько-архівна техніка;
- допоміжне обладнання;
- матеріали й інструмент.

Допоміжні технічні засоби забезпечують виконання дургорядних процесів управління: копіювання, друк, обробку кореспонденції, створення умов нормальної роботи управлінського персоналу, підтримання технічних засобів в робочому стані й їх функціонування. Створення типових АСК ТП в даний час неможливо через значне розходження організаційних систем управління підприємствами. Технічні засоби АСК ТП повинні відповідати вимогам стандартів, які спрямовані на забезпечення різної сумісності об'єкта автоматизації. Ці вимоги поділяються на групи:

- Інформаційні. Забезпечують інформаційну сумісність технічних засобів між собою і з персоналом, що обслуговує.
- Організаційні. Структура управління технологічним процесом, технологія управління, технічні засоби повинні відповідати один одному до і після впровадження АСК ТП, для чого необхідно забезпечити:
 - відповідність структур КТС – структурі управління об'єктом;
 - автоматизоване виконання основних функцій, виділення інформації, її передачу, обробку, виведення даних;
 - можливість модифікації КТС;
 - можливість створення організаційних систем контролю роботи КТС;
 - можливість створення систем контролю персоналу.
- Математичні. Згладжування невідповідностей роботи технічних засобів з інформацією може бути виконано за допомогою програм перекодування, переведення, перекладання макетів. Це обумовлює такі вимоги до математичного забезпечення:
 - швидке вирішення основних завдань АСК ТП;

- спрощення спілкування персоналу з КТС;
- можливість інформаційного узгодження різних технічних засобів.
- Технічні вимоги:
 - необхідна продуктивність для своєчасного вирішення завдань АСК ТП;
 - пристосованість до умов зовнішнього середовища підприємства;
 - надійність і ремонтпридатність;
 - використання уніфікованих, блоків, що серійно випускаються;
 - простота експлуатації й обслуговування;
 - технічна сумісність засобів, заснована на загальній елементній і конструкторській базі;
 - вимоги ергономіки, технічної естетики.
- Економічні вимоги до технічних засобів:
 - мінімальні капіталовкладення на створення КТС;
 - мінімальні виробничі площі для розміщення КТС;
 - мінімальні витрати на допоміжне обладнання.

При виборі технічних засобів АСК ТП необхідно керуватися вимогами за групами видів технічних засобів:

1. Вхідні перетворювачі (давачі) застосовуються в комплекті з вторинними приладами, регуляторами, пристроями централізованого контролю, системами управління. При виборі давачів технологічних параметрів враховується ряд факторів метрологічного і режимного характеру, найбільш суттєві з яких такі:

- допустима для АСК ТП похибка, яка визначає клас точності давача;
- інерційність давача, що характеризується його постійною часу;
- межі вимірювання з гарантованою точністю;
- вплив фізичних параметрів контрольованої і навколишнього середовища на роботу давачів;
- руйнівний вплив на давач різних середовищ, внаслідок абразивних властивостей, хімічного впливу;

- наявність в місці встановлення датчика неприпустимих для його функціонування полів і випромінювань;
- можливість застосування давача з точки зору пожежної безпеки;
- відстань, на яку може передаватися інформація з давача;
- граничні значення вимірюваної величини. Давачі вибираються в два етапи. На першому етапі задається різновид давача, на другому - визначається типорозмір.

2. Проміжні перетворювачі є внутрішньосистемними, призначені для перетворення сигналу одного виду в інший без зміни кількості інформації. Їх використовують для узгодження вхідних і вихідних сигналів КТС. Попередньо вхідні перетворювачі вибирають за класифікаційними ознаками: уніфікація вхідного сигналу, характер інформації, що перетворюється, вид носіїв інформації. Потім за технічними характеристиками остаточно вибирають тип перетворювача, враховуючи необхідний клас точності, ймовірність безвідмовної роботи. Аналогічним чином вибирається і вихідний перетворювач.

3. Засоби виведення інформації управління призначені для виведення інформації, що управляє з обчислювальних пристроїв оператору або на виконавчі механізми. Вони перетворюють машинні сигнали і сигнали оператора в форму, яка сприймається людиною і виконавчими механізмами. Для кращого сприйняття оператором інформація виводиться у вигляді сигналізації, контролю, реєстрації, діалогу. Форма подання визначається технологією роботи оператора з виведеною інформацією, її важливістю, значенням. Засоби виведення інформації вибирають за класифікаційними ознаками і за технічними характеристиками, з урахуванням конструктивних і технічних особливостей.

4. Побічні прилади служать для перетворення контрольованих параметрів та подання їх оператору, крім того, вони можуть містити пристрої, що дозволяють вводити інформацію в інші елементи технічних засобів АСК ТП, здійснювати управління технологічними об'єктами.

5. Автоматичні регулятори і виконавчі пристрої. Автоматичний регулятор виконує завдання, яке визначається елементом, його робота визначається законом регулювання. Вибір автоматичних регуляторів здійснюється в такому порядку:

- оцінка можливості вибору регулятора по відстані, на яке передається регулюючий вплив;
- за пожежобезпекою, наявністю вібрацій і полів, за руйнівним впливом середовища;
- за оцінкою інерційності регулятора. Пристрої, призначені для переміщення регулюючих органів в системах дистанційного і автоматичного керування називаються виконавчими механізмами.

Вибір виконавчих пристроїв заснований на виконанні таких вимог:

- відповідність принципу дії і конструкції виконавчих пристроїв задачі автоматизації;
- відповідність категорії виробничого приміщення;
- відповідність властивостям і значенням регулюючого середовища;
- забезпечення необхідної надійності та технічного ресурсу;
- безвідмовна робота в передбачуваному місці установки;
- забезпечення швидкості регулювання;
- лінійність ходової характеристики.

6. Засоби передачі інформації здійснюють переміщення інформації в просторі і призначені для обміну інформацією між розподіленим виробничим персоналом, а також між технічними засобами через мережі зв'язку різного призначення:

- засоби зв'язку і сигналізації, призначені для обміну мовною та документальною інформацією між персоналом АСК ТП, суміжними і підрозділами, що стоять вище, централізованого візуального контролю ходу технологічного процесу, автоматизації процесів контролю за станом зовнішнього середовища, видачі персоналу інформації про поточний час і інших сигналів;

- засоби передачі даних (канали зв'язку), за якими здійснюється передача даних з гарантованим рівнем достовірності; сумісність забезпечується реалізацією вимог до параметрів електричних сигналів, алгоритмам взаємодії з кінцевими пристроями і каналами зв'язку;
- кошти переміщення даних включають пристрої переміщення в просторі документів і носіїв інформації, при їх виборі перевага віддається засобам, здатним поєднувати кінцеві пристрої та лінії зв'язку, а також багатофункціональним установкам.

7. Засоби фіксації інформації. Під фіксацією інформації розуміється процес запам'ятовування відомостей на певний термін з обов'язковою можливістю їх зчитування. До пристроїв фіксації інформації відносяться пристрої, що запам'ятовують, рахункові реле, електричний годинник, автоматизовані картотеки, інформаційні довідкові машини. Вибір засобів фіксації інформації здійснюється в такій послідовності:

- за характером інформації, що фіксується;
- за кількістю інформації, що запам'ятовується, швидкості її надходження;
- методикою використання інформації.

8. Пристрої обробки технологічної інформації призначені для прийому, редагування, переробки інформації та видачі результатів користувачеві в зручній для нього формі. Для вибору моделей і числа ЕОМ використовують такі дані, як характеристики вирішуваних завдань, вимоги до часу вирішення і надійності роботи ЕОМ, технічні та експлуатаційні характеристики ЕОМ.

9. Допоміжне обладнання включає засоби організаційної техніки, які вибирають на основі організаційних проектів. Розрізняють кошти складання текстової документації, кошти розмноження і копіювання документів, засоби обробки документів, засоби зберігання, пошуку і транспортування документів, засоби для креслярських робіт і рахункових операцій, засоби сигналізації. Для конкретних АСК ТП засоби оргтехніки вибираються з

урахуванням вимог системи і технічних характеристик застосовуваних засобів.

До основних технічних засобів АСК ТП відносяться системні комплекси, які випускаються у вигляді окремих пристроїв, що реалізують один вид робіт з інформацією, або у вигляді комплексів, які здійснюють ряд функцій, кілька видів робіт з інформацією. Блочно-модульні вироби комплексів дозволяють будувати різні системи автоматизації технологічними процесами методом агрегування.

Технічні засоби агрегатних комплексів вибираються в кілька етапів:

- вибір типу комплексу і його модифікація в залежності від статичних і динамічних характеристик об'єкта та завдань автоматизації;
- вибір приладів, що входять в комплекс, в залежності від характеру переробки інформації та призначення модулів;
- вибір типорозміру апаратури відповідно до технологічних параметрів.

При виборі апаратури необхідно враховувати зовнішнє середовище, джерела живлення, вимоги до надійності:

- апаратура для локальних систем управління (ЛСУ), призначена для управління окремими, не пов'язаними між собою об'єктами. Ці системи утворюють нижній рівень управління, є одноконтурні та однофакторний, мають часткову ступінь автоматизації. ЛСУ виконують функції автоматичного контролю, обліку, захисту, блокування, логічного управління;
- апаратура для централізованих систем управління, що здійснює контроль, взаємозалежне автоматичне керування, регулювання складних багатфакторних об'єктів. Такі системи є багатоконтурними, для їх проектування потрібно провести розрахунок і координацію установок регуляторів. Для оперативного управління виробництвом організуються диспетчерські пункти, оснащені автоматизованими технічними засобами;

- апаратура для комплексних систем управління, яка крім функцій, які виконуються локальними і централізованими системами, передбачає оптимальне управління, адаптивне управління, розрахунок ТЕП, оперативне планування і облік, рішення інформаційних завдань для фахівців.

10. Надійність АСК ТП. При розгляді технічного забезпечення розглядається і питання надійності АСК ТП. При цьому необхідно провести дослідження АСК ТП, виділивши такі моменти:

- складність (велику кількість різних технічних засобів і персоналу);
- багатфункціональність;
- багатоспрямованість використання елементів в системі;
- множинність видів відмов (причини виникнення, наслідки);
- взаємозв'язок надійності і економічної ефективності;
- залежність надійності від технічної експлуатації;
- залежність надійності від КТС і структури алгоритмів.

2.2. Сучасні давачі й їх характеристики для отримання даних технологічного процесу

Автоматизація різних технологічних процесів, ефективне управління різними агрегатами, машинами, механізмами вимагають численних вимірювань різноманітних фізичних величин. **Давачі (датчики)** (в літературі часто звані також вимірювальними перетворювачами), або по-іншому, **сенсори**, є елементами багатьох систем автоматики – з їх допомогою отримують інформацію про параметри контрольованої системи або пристрою.

Давач – це елемент вимірювального, сигнального, регулюючого або керуючого пристрою, що перетворює контрольовану величину (температуру, тиск, частоту, силу світла, електрична напруга, струм і т.д.) в сигнал, зручний для вимірювання, передачі, зберігання, обробки, реєстрації, а іноді і для впливу ним на керовані процеси. Або простіше, **давач** – це пристрій, що

перетворює вхідний вплив будь-якої фізичної величини в сигнал, зручний для подальшого використання.

Використовувані датчики досить різноманітні і можуть бути **класифіковані за різними ознаками:**

- в залежності від виду вхідної (вимірюваної) величини розрізняють: датчики механічних переміщень (лінійних і кутових), пневматичні, електричні, витратоміри, датчики швидкості, прискорення, зусилля, температури, тиску та ін.

В даний час існує приблизно такий розподіл частки вимірювань різних фізичних величин у промисловості: температура - 50%, витрата (масова і об'ємна) - 15%, тиск - 10%, рівень - 5%, кількість (маса, об'єм) - 5%, час - 4%, електричні і магнітні величини - менш 4%.

- за видом вихідної величини, в яку перетворюється вхідна величина, розрізняють неелектричні та електричні: датчики постійного струму (ЕРС або напруги), датчики амплітуди змінного струму (ЕРС або напруги), датчики частоти змінного струму (ЕРС або напруги), датчики опору (активного, індуктивного або ємнісного) і ін.

Більшість датчиків є електричними. Це обумовлено такими перевагами електричних вимірювань:

- електричні величини зручно передавати на відстань, причому передача здійснюється з високою швидкістю;
- електричні величини універсальні в тому сенсі, що будь-які інші величини можуть бути перетворені в електричні і навпаки;
- вони точно перетворюються в цифровий код і дозволяють досягти високої точності, чутливості і швидкодії засобів вимірювань.

За принципом дії датчики можна розділити на два класи: генераторні і параметричні (датчики-модулятори). Генераторні датчики здійснюють безпосереднє перетворення вхідної величини в електричний сигнал. Параметричні датчики вхідну величину перетворюють в зміну будь-якого електричного параметра (R, L або C) датчика. Датчики також можна розділити на омичні,

реостатні, фотоелектричні (оптико-електронні), індуктивні, ємнісні і т.д.

Розрізняють три класи датчиків:

- аналогові датчики, тобто датчики, що виробляють аналоговий сигнал, пропорційно зміні вхідної величини;
- цифрові датчики, які генерують послідовність імпульсів або двійкове слово;
- бінарні (двійкові) датчики, які виробляють сигнал тільки двох рівнів: «ввімкнено / вимкнено» (0 або 1); набули широкого поширення завдяки своїй простоті.

Вимоги до датчиків:

- однозначна залежність вихідної величини від вхідних;
- стабільність характеристик у часі;
- висока чутливість;
- малі розміри і маса;
- відсутність зворотного впливу на контрольований процес і на контрольований параметр;
- робота при різних умовах експлуатації;
- різні варіанти монтажу.

Параметричні датчики (датчики-модулятори) вхідну величину X перетворюють в зміну будь-якого електричного параметра (R , L або C) датчика. Передати на відстань зміну перерахованих параметрів датчика без сигналу, що несе енергію, (напруги або струму) неможливо. Виявити зміну відповідного параметра датчика тільки і можна за реакцією датчика на струм або напругу, оскільки перераховані параметри і характеризують цю реакцію. Тому параметричні датчики вимагають застосування спеціальних вимірювальних ланцюгів з живленням постійним або змінним струмом.

Омічні (резистивні) датчики – принцип дії заснований на зміні їх активного опору при зміні довжини, площі перетину S або питомого опору ρ :

$$R = \rho / S$$

Крім того, використовується залежність величини активного опору від контактного тиску і освітленості фотоелементів. Відповідно до цього омичні датчики діляться на:

- контактні;
- потенціометричні (реостатні);
- терморезисторні;
- фоторезисторні.

Контактні датчики – це найпростіший вид резисторних датчиків, які перетворюють переміщення первинного елемента в стрибкоподібну зміну опору електричного кола. За допомогою контактних датчиків вимірюють і контролюють зусилля, переміщення, температуру, розміри об'єктів, їх форму і т.д. До контактних датчиків належать шляхові й кінцеві вимикачі, контактні термометри і так звані електродні датчики, які використовуються в основному для вимірювання граничних рівнів електропровідних рідин.

Контактні датчики можуть працювати як на постійному, так і на змінному струмі. Залежно від меж виміру контактні датчики можуть бути односторонніми та багаторічними. Останні використовують для вимірювання величин, що змінюються в значних межах, при цьому частини резистора, включеного в електричний ланцюг, послідовно закорочуються. Недолік контактних датчиків – складність здійснення безперервного контролю і обмежений термін служби контактної системи. Але завдяки граничній простоті цих датчиків їх широко застосовують в системах автоматики.

Реостатні датчики являють собою резистор із змінним активним опором. Вхідною величиною датчика є переміщення контакту, а вихідною – зміна його опору. Рухомий контакт механічно пов'язаний з об'єктом, переміщення (кутове або лінійне) якого необхідно перетворити. Найбільшого поширення набула потенціометрична схема включення реостатного датчика, в якій реостат включають за схемою подільника напруги. Нагадаємо, що дільником напруги називають електротехнічний пристрій для ділення постійної або змінної напруги на частини; дільник

напруги дозволяє знімати (використовувати) тільки частина наявної напруги за допомогою елементів електричного кола, що складається з резисторів, конденсаторів або котушок індуктивності. Змінний резистор, що включається за схемою подільника напруги, називають потенціометром.

Зазвичай реостатні датчики застосовують в механічних вимірювальних приладах для перетворення їх показань в електричні величини (струм або напруга), наприклад, в поплавцевих вимір рівня рідин, різних манометрах і т.д.

Датчик у вигляді простого реостата майже не використовується внаслідок значної нелінійності його статичної характеристики:

$$I_n = f(x),$$

де I_n – струм в напрузі.

Вихідною величиною такого датчика є падіння напруги $U_{вих}$ між рухомим і одним з нерухомих контактів. Залежність вихідної напруги від переміщення x контакту $U_{вих} = f(x)$ відповідає закону зміни опору вздовж потенціометра. Закон розподілу опору по довжині потенціометра, який визначається його конструкцією, може бути лінійним або нелінійним.

Потенціометричні датчики, конструктивно представляють собою змінні резистори, виконують з різних матеріалів – обмотувального дроту, металевих плівок, напівпровідників і т.д.

Тензорезистори (тензометричні датчики) служать для вимірювання механічних напружень, невеликих деформацій, вібрації. Дія тензорезисторів базується на тензоефекті, що полягає в зміні активного опору провідникових і напівпровідникових матеріалів під впливом прикладених до них зусиль.

Термометричні датчики (*терморезистори*) – опір залежить від температури. Терморезистор як датчик використовують двома способами:

- температура терморезистора визначається навколишнім середовищем; струм, що проходить через терморезистор, настільки малий, що не викликає нагрівання терморезистора. При цій умові терморезистор використовується як датчик температури і часто називається «*термометром опору*»;
- температура терморезистора визначається ступенем нагрівання постійним за величиною струмом та умовами охолодження. В цьому випадку встановлена температура визначається умовами тепловіддачі поверхні терморезистора (швидкістю руху навколишнього середовища – газу або рідини – щодо терморезистора, її щільністю, в'язкістю і температурою), тому терморезистор може бути використаний як датчик швидкості потоку, теплопровідності навколишнього середовища, щільності газів і т.д. У датчиках такого роду відбувається якби двоступеневе перетворення: вимірювана величина спочатку перетвориться в зміну температури терморезистора, яка потім перетвориться в зміну опору.

Терморезистори виготовляють як з чистих металів, так і з напівпровідників. Матеріал, з якого виготовляється такі датчики, повинен володіти високим температурним коефіцієнтом опору, по можливості лінійною залежністю опору від температури, хорошою відтворюваністю властивостей і інертністю до впливів навколишнього середовища. Найбільшою мірою всіма зазначеними властивостями задовольняє платина; в трохи меншій – мідь і нікель.

У порівнянні з металевими терморезисторами більш високою чутливістю володіють напівпровідникові терморезистори (термістори).

Індуктивні датчики служать для безконтактного отримання інформації про переміщення робочих органів машин, механізмів, роботів і т.п. та перетворення цієї інформації в електричний сигнал.

Принцип дії індуктивного датчика заснований на зміні індуктивності обмотки на магнітопроводі в залежності від положення окремих елементів магнітопроводу (якоря, сердечника і ін.). У таких датчиках лінійне або кутове переміщення X (вхідна величина) перетворюється в зміну індуктивності (L) датчика. Застосовуються для вимірювання кутових і лінійних переміщень, деформацій, контролю розмірів і т.д.

У найпростішому випадку індуктивний датчик являє собою котушку індуктивності з магнітопроводом, рухливий елемент якого (якір) переміщається під дією вимірюваної величини.

Індуктивний датчик розпізнає і відповідно реагує на всі струмопровідні предмети. Індуктивний датчик є безконтактним, не вимагає механічного впливу, працює безконтактно за рахунок зміни електромагнітного поля.

Дані датчики мають ряд переваг:

- немає механічного зносу, відсутні відмови, пов'язані зі станом контактів;
- відсутній брязкіт контактів і помилкові спрацьовування;
- висока частота перемикань до 3000 Hz;
- стійкий до механічних впливів.

До недоліків можна віднести такі – порівняно мала чутливість, залежність індуктивного опору від частоти напруги живлення, значний зворотний вплив датчика на вимірювану величину (за рахунок тяжіння якоря до сердечника).

Ємнісні датчики – принцип дії заснований на залежності електричної ємності конденсатора від розмірів, взаємного розташування його обкладок і від діелектричної проникності середовища між ними.

Для двохобкладочного плоского конденсатора електрична ємність визначається виразом:

$$C = \frac{\epsilon_0 \epsilon S}{h};$$

де ϵ_0 – діелектрична постійна;

ϵ – відносна діелектрична проникність середовища між обкладками;

S – активна площа обкладок;

h – відстань між обкладками конденсатора.

Залежності $C(S)$ та $C(h)$ використовують для перетворення механічних переміщень в зміну ємності.

Ємнісні датчики, також як і індуктивні, харчуються змінною напругою (зазвичай підвищеної частоти – до десятків мегагерц). В якості вимірювальних схем зазвичай застосовують мостові схеми і схеми з використанням резонансних контурів. В останньому випадку, як правило, використовують залежність частоти коливань генератора від ємності резонансного контуру, тобто датчик має частотний вихід.

Переваги ємнісних датчиків – простота, висока чутливість і мала інерційність. Недоліки – вплив зовнішніх електричних полів, відносна складність вимірювальних пристроїв.

Ємнісні датчики застосовують для вимірювання кутових переміщень, дуже малих лінійних переміщень, вібрацій, швидкості руху і т.д., А також для відтворення заданих функцій (гармонійних, пілообразних, прямокутних і т.д.).

Ємнісні перетворювачі, діелектрична проникність яких змінюється за рахунок переміщення, деформації або зміни складу діелектрика, застосовують в якості датчиків рівня непроводячих рідин, сипучих і порошкоподібних матеріалів, товщини шару матеріалів, що не проводять електричний струм, (товщині), а також контролю вологості і складу речовини.

Генераторні датчики здійснюють безпосереднє перетворення вхідної величини в електричний сигнал. Такі датчики перетворюють енергію джерела вхідної (вимірюваної) величини відразу в електричний сигнал, тобто вони є якби генераторами електроенергії (звідки і назва таких датчиків – вони генерують електричний сигнал).

Додаткові джерела електроенергії для роботи таких датчиків принципово не потрібні (проте додаткова електроенергія може знадобитися для посилення вихідного сигналу датчика,

його перетворення в інші види сигналів і інших цілей). Генераторними є термоелектричні, п'єзоелектричні, індукційні, фотоелектричні і багато інших типів датчиків.

Індукційні давачі перетворюють вимірювану неелектричну величину в ЕРС індукції. Принцип дії давачів заснований на законі електромагнітної індукції. До цих датчиків належать тахогенератори постійного і змінного струму, що представляють собою невеликі електромашинні генератори, у яких вихідна напруга пропорційно кутовій швидкості обертання вала генератора. Тахогенератори використовуються як датчики кутової швидкості.

Тахогенератор є електричною машиною, що працює в генераторному режимі. При цьому виробляється ЕРС пропорційна швидкості обертання і величині магнітного потоку. Крім того, зі зміною швидкості обертання змінюється частота ЕРС. Застосовуються як датчики швидкості (частоти обертання).

Температурні датчики. У сучасному промисловому виробництві найбільш поширеним є вимірювання температури (так, на атомній електростанції середнього розміру є близько 1500 точок, в яких проводиться такий вимір, а на великому підприємстві хімічної промисловості подібних точок присутній понад 20000). Широкий діапазон вимірюваних температур, різноманітність умов використання засобів вимірювальної техніки та вимог до них визначають розмаїття застосовуваних засобів вимірювання температури.

Якщо розглядати датчики температури для промислового застосування, то можна виділити їх основні класи: кремнієві датчики температури, біметалічні датчики, рідинні і газові термометри, термоіндикатори, термістори, термопари, термоперетворювачі опору, інфрачервоні датчики.

Кремнієві датчики температури використовують залежність опору напівпровідникового кремнію від температури. Діапазон вимірюваних температур $-50 \dots + 150^{\circ}\text{C}$. Застосовуються в основному для вимірювання температури всередині електронних приладів.

Біметалічний датчик зроблений з двох різнорідних металевих пластин, скріплених між собою. Різні метали мають різний температурний коефіцієнт розширення. Якщо з'єднані в пластину метали нагріти або охолодити, то вона зігнеться, при цьому замкне (розімкнеться) електричні контакти або переведе стрілку індикатора. Діапазон роботи біметалевих датчиків $-40 \dots + 550^\circ\text{C}$. Використовуються для вимірювання поверхні твердих тіл і температури рідин. Основні області застосування – автомобільна промисловість, системи опалення та нагріву води.

Термоіндикатори – це особливі речовини, що змінюють свій колір під впливом температури. Зміна кольору може бути оборотною і необоротною. Виробляються у вигляді плівок.

Термоперетворювачі опору. Принцип дії термоперетворювачів опору (терморезисторів) заснований на зміні електричного опору провідників і напівпровідників в залежності від температури (розглянути раніше).

Платинові терморезистори призначені для вимірювання температур в межах від -260 до 1100°C . Широке поширення на практиці отримали більш дешеві мідні терморезистори, які мають лінійну залежність опору від температури.

Недоліком міді є невеликий питомий опір і легка окислюваність при високих температурах, внаслідок чого кінцева межа застосування мідних термометрів опору обмежується температурою 180°C . За стабільністю і відтворюваністю характеристик мідні терморезистори поступаються платиновим. Нікель використовується в недорогих датчиках для вимірювання в діапазоні кімнатних температур.

Напівпровідникові терморезистори (термістори) мають негативний або позитивний температурний коефіцієнт опору, значення якого при 20°C становить $(2 \dots 8) \cdot 10^{-2} (C^0)^{-1}$, тобто на порядок більше, ніж у міді і платини. Напівпровідникові терморезистори при досить малих розмірах мають високі значення опору (до 1 МОм). В якості напівпровідникового матеріалу використовуються оксиди металів: напівпровідникові терморезистори типів КМТ – суміш оксидів кобальту і марганцю та

ММТ – міді і марганцю. Напівпровідникові датчики температури мають високу стабільність характеристик в часі і застосовуються для зміни температур в діапазоні від 100 до 200°C.

Термоелектричні перетворювачі (термопари) – принцип дії термопар заснований на термоелектричному ефекті, який полягає в тому, що при наявності різниці температур місць з'єднань (спаїв) двох різнорідних металів або напівпровідників в контурі виникає електрорушійна сила, яка називається термоелектрорушійною (скорочено термо-ЕРС). У певному інтервалі температур можна вважати, що термо-ЕРС прямо пропорційна різниці температур $\Delta T = T_1 - T_0$ між спаєм і кінцями термопари.

Сполучені між собою кінці термопари, що занурюються в середу, температура якої вимірюється, називають робочим кінцем термопари. Кінці, які знаходяться в навколишньому середовищі, і які зазвичай приєднують проводами до вимірювальної схеми, називають вільними кінцями. Температуру цих кінців необхідно підтримувати постійною. За цих умов термо-ЕРС E_T буде залежати лише від температури T_1 робочого кінця.

$$U_{\text{вих}} = E_T = C(T_1 - T_0),$$

де C – коефіцієнт, що залежить від матеріалу провідників термопари.

Створювана термопарами ЕРС порівняно невелика: вона не перевищує 8 мВ на кожному 100°C і зазвичай не перевищує за абсолютною величиною 70 мВ. Термопари дозволяють вимірювати температуру в діапазоні від -200 до 2200°C.

Найбільшого поширення для виготовлення термоелектричних перетворювачів отримали платина, платинородій, хромель, алюмель.

Термопари мають такі переваги: простота виготовлення і надійність в експлуатації, дешевизна, відсутність джерел живлення і можливість вимірювань у великому діапазоні температур.

Поряд з цим термопарам властиві і деякі недоліки – менша, ніж у терморезисторів, точність вимірювання, наявність значної теплової інерційності, необхідність введення поправки на температуру вільних кінців і необхідність в застосуванні спеціальних сполучних дротів.

Інфрачервоні давачі (пірометри) – використовують енергію випромінювання нагрітих тіл, що дозволяє вимірювати температуру поверхні на відстані. Пірометри діляться на радіаційні, яскравості та колірні. Радіаційні пірометри використовуються для вимірювання температури від 20 до 2500°C, причому прилад вимірює інтегральну інтенсивність випромінювання реального об'єкта.

Яскраві (оптичні) пірометри використовуються для вимірювання температур від 500 до 4000°C. Вони засновані на порівнянні в вузькій ділянці спектру яскравості досліджуваного об'єкта з яскравістю зразкового випромінювача (фотометричної лампи).

Кольорові пірометри засновані на вимірі відношення інтенсивностей випромінювання на двох довжинах хвиль, які обирають зазвичай в червоній або синій частині спектра; вони використовуються для вимірювання температури в діапазоні від 800°C.

Пірометри дозволяють вимірювати температуру у важкодоступних місцях і температуру рухомих об'єктів, високі температури, де інші датчики вже не працюють.

Кварцові термоперетворювачі застосовуються для вимірювання температур від – 80 до 250°C часто використовуються так звані кварцові термоперетворювачі, що використовують залежність власної частоти кварцового елемента від температури. Робота даних датчиків заснована на тому, що залежність частоти перетворювача від температури і лінійність функції перетворення змінюються в залежності від орієнтації зрізу щодо осей кристала кварцу. Дані датчики широко використовуються в цифрових термометрах.

П'єзоелектричні датчики. Дія п'єзоелектричних датчиків базується на використанні п'єзоелектричного ефекту (п'єзо-ефекту), що полягає в тому, що при стисненні або розтягуванні деяких кристалів на їх гранях з'являється електричний заряд, величина якого пропорційна силі, що діє.

П'єзоэффект є зворотнім, тобто прикладена електричне напруга викликає деформацію п'єзоелектричного зразка – стиснення або розтягнення його відповідно знаку прикладеної напруги. Це явище, зване зворотним п'єзоэффектом, використовується для збудження та прийому акустичних коливань звукової та ультразвукової частоти. Використовуються для вимірювання сил, тиску, вібрації і т.д.

Оптичні (фотоелектричні) датчики. Розрізняють аналогові й дискретні оптичні датчики. У аналогових датчиків вихідний сигнал змінюється пропорційно зовнішньої освітленості. Основна область застосування – автоматизовані системи управління освітленням.

Датчики дискретного типу змінюють вихідний стан на протилежний при досягненні заданого значення освітленості.

Фотоелектричні датчики можуть бути застосовані практично у всіх галузях промисловості. Датчики дискретної дії використовуються як своєрідні безконтактні вимикачі для підрахунку, виявлення, позиціонування та інших завдань на будь-якої технологічної лінії.

Оптичний безконтактний датчик реєструє зміну світлового потоку в контрольованій області, що пов'язана зі зміною положення в просторі будь-яких рухомих частин механізмів і машин, відсутністю або присутністю об'єктів. Завдяки великим відстаням спрацьовування оптичні безконтактні датчики знайшли широке застосування в промисловості і не тільки.

Оптичний безконтактний датчик складається з двох функціональних вузлів, приймача і випромінювача. Дані вузли можуть бути виконані як в одному корпусі, так і в різних корпусах.

За методом виявлення об'єкта фотоелектричні датчики поділяються на 4 групи:

- 1) перетин променя – в цьому методі передавач і приймач розділені по різних корпусах, що дозволяє встановлювати їх навпроти один одного на робочій відстані. Принцип роботи заснований на тому, що передавач постійно посиляє світловий промінь, який приймає приймач. Якщо світловий сигнал датчика припиняється через перекриття стороннім об'єктом, приймач негайно реагує змінюючи стан виходу.
- 2) відбиття від рефлектора – в цьому методі приймач і передавач датчика знаходяться в одному корпусі. Навпроти датчика встановлюється рефлектор (відбивач). Датчики з рефлектором влаштовані так, що завдяки поляризаційному фільтру вони сприймають відображення тільки від рефлектора. Це рефлектори, які працюють за принципом подвійного відображення. Вибір відповідного рефлектора визначається необхідною відстанню і монтажними можливостями;
- 3) відображення від об'єкта – в цьому методі приймач і передавач датчика знаходяться в одному корпусі. Під час робочого стану датчика всі об'єкти, що потрапляють в його робочу зону, стають своєрідними рефлекторами. Як тільки світловий промінь відбившись від об'єкта потрапляє на приймач датчика, той негайно реагує, змінюючи стан виходу;
- 4) фіксоване відображення від об'єкта – принцип дії датчика такий же як і у «відображення від об'єкта», але більш чутливо реагує на відхилення від настройки на об'єкт. Наприклад, можливо детектування роздутої пробки на пляшці з кефіром, неповне наповнення вакуумної упаковки з продуктами і т.д.

За своїм призначенням фотодатчики діляться на дві основні групи: датчики загального застосування і спеціальні датчики. До спеціальних, відносяться типи датчиків, призначені

для вирішення вузького кола завдань. Наприклад, виявлення кольорової мітки на об'єкті, виявлення контрастним межі, наявність етикетки на прозорій упаковці і т.д. Завдання датчика виявити об'єкт на відстані. Це відстань варіюється в межах 0,3 мм...50 м, в залежності від обраного типу датчика і методу виявлення.

Мікрохвильові датчики. На зміну кнопково-релейним пультам приходять мікропроцесорні АСК ТП високої продуктивності і надійності, датчики оснащуються цифровими інтерфейсами зв'язку, проте це не завжди призводить до підвищення загальної надійності системи і достовірності її роботи. Причина полягає в тому, що самі принципи дії більшості відомих типів датчиків накладають жорсткі обмеження на умови, в яких вони можуть використовуватися.

Наприклад, для спостереження за швидкістю руху промислових механізмів широко застосовуються безконтактні (ємнісні й індуктивні), а також тахогенераторні пристрої контролю швидкості (ПКШ). Тахогенераторні ПКШ мають механічний зв'язок з об'єктом, що рухається, а зона чутливості безконтактних приладів не перевищує декількох сантиметрів.

Все це не тільки створює незручності при монтажі датчиків, а й істотно ускладнює використання цих приладів в умовах пилу, яка налипає на робочі поверхні, викликаючи помилкові спрацьовування. Перераховані типи датчиків не здатні безпосередньо контролювати об'єкт (наприклад, стрічку конвеєра) – вони налаштовуються на рух роликів, крильчаток, натяжних барабанів і т.д. Вихідні сигнали деяких приладів настільки слабкі, що лежать нижче рівня промислових перешкод від роботи потужних електричних машин.

Аналогічні труднощі виникають при використанні традиційних сигналізаторів рівня – датчиків наявності сипучого продукту. Такі пристрої необхідні для своєчасного відключення подачі сировини в виробничі ємності. До помилкових спрацьовувань призводить не тільки налипання і пил, але і дотик потоку продукту при його надходженні в бункер. У

неопалюваних приміщеннях на роботу датчиків впливає навколишня температура. Помилкові спрацьовування сигналізаторів викликають часті зупинки і запуски навантаженого технологічного обладнання – основну причину його аварій, призводять до завалів, обриву конвеєрів, виникнення пожежо- та вибухонебезпечних ситуацій.

Зазначені проблеми кілька років тому привели до розробки принципово нових типів приладів – радіолокаційних датчиків контролю швидкості, датчиків руху і підпору, робота яких заснована на взаємодії контрольованого об'єкта з радіосигналом частотою 10^{10} Гц.

Використання мікрохвильових методів контролю за станом технологічного обладнання дозволяє повністю позбутися від недоліків датчиків традиційних типів.

Відмінними рисами цих пристроїв є:

- відсутність механічного та електричного контакту з об'єктом (середовищем), відстань від датчика до об'єкта може становити кілька метрів;
- безпосередній контроль об'єкта (транспортної стрічки, ланцюга), а не їх приводів, натяжних барабанів і т.д.;
- мале енергоспоживання;
- нечутливість до налипання продукту за рахунок великих робочих відстаней;
- висока стійкість і спрямованість дії;
- разова настройка на весь термін служби;
- висока надійність, безпека, відсутність іонізуючих випромінювань.

Принцип дії датчика заснований на зміні частоти радіосигналу, відбитого від рухомого об'єкта. Це явище («ефект Доплера») широко використовується в радіолокаційних системах для дистанційного вимірювання швидкості. Рухомий об'єкт викликає появу електричного сигналу на виході мікрохвильового приймально-передавального модуля.

Так як рівень сигналу залежить від властивостей об'єкта, що відображає, датчики руху можуть використовуватися для

того, щоб сигналізувати про обрив ланцюга (стрічки), наявності на конвеєрній стрічці будь-яких предметів чи матеріалів. Стрічка має гладку поверхню і низький коефіцієнт відображення. Коли повз датчика, встановленого над робочою гілкою транспортера, починає рухатися продукт, збільшуючи коефіцієнт відображення, прилад сигналізує про рух, тобто, фактично про те, що стрічка не порожня. За тривалістю вихідного імпульсу можна на значній відстані судити про розмір переміщуваних предметів, виробляти селекцію і т.д.

При необхідності заповнити будь-яку ємність (від бункера до шахти) можна точно визначити момент закінчення засипки – опущений на певну глибину датчик буде показувати рух наповнювача до тих пір, поки не буде засипаний.

Конкретні приклади використання мікрохвильових датчиків руху в різних галузях промисловості визначаються її специфікою, але в цілому вони здатні вирішувати найрізноманітніші завдання безаварійної експлуатації обладнання та підвищити інформативність автоматизованих систем керування.

Розглянемо приклад датчиків різного призначення, їх особливості, для більш детального вивчення можна скористатися DataSheet фірми GmbH & Co.KG (Німеччина), сайт jumo.net. Вибір типу і виду датчиків буде залежати від конкретного завдання і вихідних, і вихідних умов їх експлуатації та необхідних діапазонів вимірювань.

Термоелемент, що угвинчується, з приєднувальних головною форми В тип 901020/10 і 901020/10. Даний термоелемент використовується для вимірювання температури в рідких і газоподібних середовищах. Надійна герметичність такої конструктивної форми як при вакуумі, так і надмірному тиску є важливим критерієм при виборі. Галузю застосування, зокрема, є нагрівальна техніка, будівництва печей, машинобудування, а також технологічні процеси.

Приєднувальна головка може використовуватися для температур до +100°C. Вимірювання можуть вестися в діапазоні

– 200...+800°C. Разом із стандартною формою приєднувальної головки, в комплект поставляються виконавча форма BUZ, BUZH, ВВК. Захисні трубки з нержавіючої сталі захищають вимірювальну частину від хімічного впливу і від механічних пошкоджень. У вимірювальній частині використовується термопара згідно з DIN EN 60 584 класу 2 і DIN 43 710. Передбачена можливість використовувати з двома термопарами. Технічні характеристики наведено в таблиці 2.1.

Таблиця 2.1

Технічні характеристики термоелемент, що угвинчується, з приєднувальних головою форми В

Параметр	Дані
1	2
Приєднувальна головка	Форма В DIN 43 729, лиття АІ, М 20х1,5,ІР 65, температура навколишнього середовища -40...+100°C; Форма BUZ, лиття АІ, М 20х1,5,ІР 65, температура навколишнього середовища - 40...+100 °C; Форма BUZH, лиття АІ, М 20х1,5,ІР 65, температура навколишнього середовища - 40...+100 °C; Форма ВВК, пластмаса, М 20х1,5,ІР 54, температура навколишнього середовища - 30...+130 °C
Трубка горловини	Нержавіюча сталь 1.4571, довжина 130мм
Підмикання до процесу	Різьба, нержавіюча сталь 1.4571
Захисна трубка	Нержавіюча сталь 1.4571, діаметр 9мм
Вимірювальна частина	Змінний, ізольований монтаж: 1хFe-CuNi “L”, DIN 43 710, клас 2, робоча температура -200...+600 °C; 1хFe-CuNi “J”, DIN 60 584, клас 2, робоча температура -200...+600 °C; 1хNiCr-Ni “K”, DIN EN 60 584, клас 2, робоча температура -200...+800 °C; 2хFe-CuNi “L”, DIN 43 710, клас 2, робоча температура -200...+600 °C;

2xFe-CuNi “J”, DIN 60 584, клас 2,
робоча температура -200...+600 °C;
2xNiCr-Ni “K”, DIN EN 60 584, клас 2,
робоча температура -200...+800 °C

Продовження табл. 2.1

1	2
Вимірювальний перетворювач	Програмований вимірювальний перетворювач, вихід 4...20мА/20...4мА, WtransВ, програмований перетворювач з передаванням даних радіоканалом
Приналежності	Захисна гільза

Вигляд термоелемента, що угвинчується, з приєднувальною головкою форми В представлений на рис. 2.7, а його габаритні розміри на рис. 2.8.

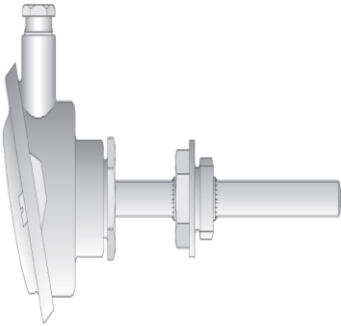


Рисунок 2.7 – Загальний вигляд термоелемента, що угвинчується, з приєднувальною головкою форми В В

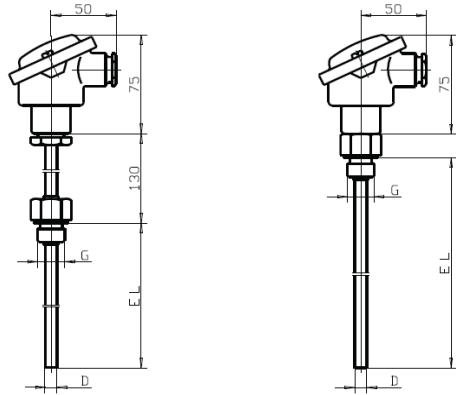


Рисунок 2.8 – Габаритні розміри термоелемента, що угвинчується, з приєднувальною головкою форми В: а) з головкою; б) без головки

Вимірювальні перетворювачі відносної вологості (з ємнісним сенсором) і температури для систем вентиляції та кондиціонування. Вимірювальні перетворювачі вологості серії 907020 були спеціально розроблені для вирішення завдань систем вентиляції та кондиціонування і представлені рис. 2.8.



Рисунок 2.8 – Вимірювальні перетворювачі вологості серії 907020

Основним елементом вимірювального перетворювача є ємнісний сенсор вологочутливості, величина електричної ємності якого залежить від вологості. Цей ємнісний чутливий елемент складається з несучої підкладки, покритої гігроскопічною полімерною плівкою. Холодильні камери пов'язують (абсорбують) або звільняють молекули води з газового середовища, де проводиться вимір (повітря), при цьому змінюється електрична ємність сенсора. Електронна схема перетворює зміну ємності, пов'язану з вологістю, в нормований електричний сигнал постійного струму або напруги.

Перетворювач розроблений для вимірювання вологості повітря при нормальному атмосферному тиску і не призначений для вимірювань в агресивних газах. У стандартному виконанні

чутливий елемент захищений фільтром у вигляді металевої сітки (крім кімнатного виконання). Як аксесуари можуть бути поставлені інші фільтри, що дозволяють застосовувати перетворювачі в більш жорстких умовах.

Перетворювач надійно вимірює вологість в діапазоні 5-95%. Доступні версії з розширеним температурним діапазоном. На додаток до типової версії з нормованими вихідними сигналами 4 ... 20 мА і 0 ... 10В випускаються виконання з вихідним сигналом 0 ... 20 мА і пасивним / резистивним вихідним сигналом Pt100 через канал температури.

Різні виконання перетворювача дозволяють обрати простий і відповідний до застосування й особливостей місця установки та монтажу приладу. Габаритні розміри на рис. 2.9.

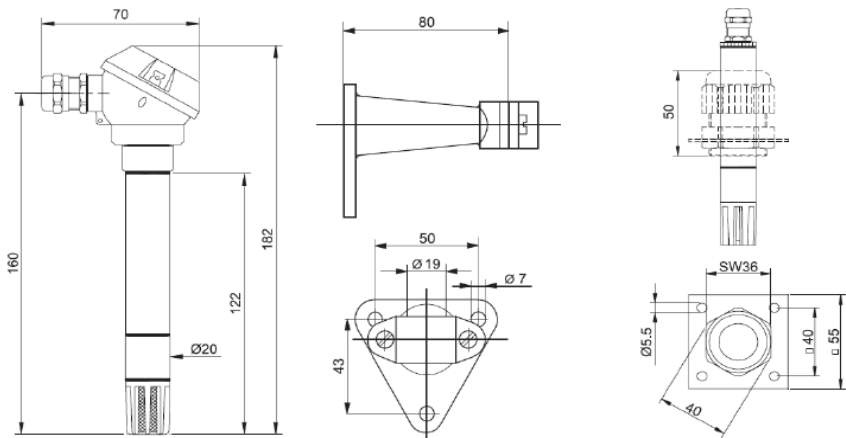


Рисунок 2.9 - Габаритні розміри перетворювачі вологості серії 907020

Розглянемо технічні характеристики, представлені в таблиці 2.2

Таблиця 2.2

Технічні характеристики перетворювачі вологості серії 907020

Параметри	Значення
1	2
Канал вологості	
Чуттєвий елемент	Ємнісний (тонкоплівковий, з низькою гігроскопічністю)
Діапазон вимірювань відносної вологості	0...100% гН (відносна вологість)
Робочий діапазон вологості	стрижневе і каналне виконання: 5...95% гН
Точність вимірювання вологості при 20 °С	± 2% гН (в діапазоні температур 10...40 °С, додаткова похибка менше 0,1% гН / К), для стрижневого виконання ± 2% гН (при 23 С і в діапазоні 40...60 % гН) ± 2,5% гН (при інших умовах), для каналного виконання ± 3,5% гН, для кімнатного виконання
Вплив температури	стрижневе виконання: менше 0,1% гН / С (при температурах не менше 10 °С і більше 40 °С каналне виконання: менше 0,15 гН / С кімнатна виконання: 0,05% гН / С (при 20 °С при 50% гН)
Робоче середовище	Повітря при атмосферному тиску, без агресивних газів, без утворення конденсату.
Вихідні сигнали	0 ... 20 мА (тільки для кімнатного і каналного виконань) 0 ... 1 В (тільки для стрижневого виконання) 0 ... 10 В 4 ... 20 мА (двухпроводная схема)
Канал температури	
Чутливий елемент	Тонкоплівковий сенсор Pt 100, клас В за DIN EN 60 751
Діапазон вимірювань	-30 ... + 70 °С, стрижневе виконання 0 ... 50 °С, кімнатне і каналне виконання 0 ... 100 °С, каналне виконання

Продовження табл. 2.2

1	2
Робочий діапазон	-10 ... + 60 ° С, кімнатне виконання -30 ... + 80 ° С, каналне виконання
Точність вимірювання температур	± 0,3° С, стрижневе і каналне виконання ± 0,8° С, кімнатне виконання
Електричні характеристики	
Напруга живлення	6...30В постійного струму, стрижневе виконання з вихідним сигналом 0...1В 15 ... 24В постійного струму, кімнатне виконання 15...30В постійного струму, стрижневе і каналне виконання 24В постійного струму, кімнатна і каналне виконання з вихідним сигналом 0...10В
Навантаження (для входу струму)	$R_n = (U_{жив} - 10В)/0,02А ± 50Ом$
Навантаження (для виходу за напругою)	не менше 10кОм (для сигналу 0 ... 10В) не менше 2кОм (для сигналу 0 ... 1В)
Електромагнітна сумісність	за EN 61326
Конструктивні характеристики	
Корпус	кімнатна виконання: ударміцне пластмаса каналне виконання: ABS пластик, алюмінієвий стрижень, пофарбований стрижневе виконання: алюмінієва головка форми J, стрижень, як в каналному виконанні, в якості опції нероз'ємний кабель 1,5 м
Габаритні розміри	кімнатна виконання: 155 x 70 x 43 мм каналне виконання: 248 x 120 x 80 мм (довжина вставної частини 200 мм, діаметр 20 мм) стрижневе виконання: довжина вставної частини 122 мм, діаметр 20 мм
Маса	кімнатна виконання: близько 200 г каналне виконання: близько 400 г штирьове виконання: близько 140 г штирьове виконання з головкою форми J: близько 340 г

Розглянемо датчик рівня JUMO MAERA S28 тип 404392. Зонд рівня призначений для гідростатичного вимірювання рівня заповнення резервуара або, наприклад, для визначення рівня води у відкритих водоймах. Зонд рівня розрахований на висоту наповнення від 2,5 м до 250 м вод. ст. (водяного стовпа). Завдяки міцній конструкції і заданій для виготовлення матеріалу можливий як внутрішній, так і зовнішній монтаж. Для варіанту зовнішнього монтажу передбачений захист від перенапруги, що запобігає руйнуванню зонда рівня при попаданні блискавки у водойму, в яку поміщений зонд. При визначенні висоти наповнення необхідно враховувати, що щільність будь-якої рідини залежить від температури, тобто Показник щільності при температурі рідини 5°C буде відрізнятися від показника при температурі 30°C. Ці дані представлені в зведених таблицях. Можливість відображення температурного режиму рідини під час виконання вимірювань надає опціональний варіант виконання з інтегрованим датчиком температури Pt100. На рис. 2.10 представлений загальний вигляд датчика рівня JUMO MAERA S28.



Рисунок 2.10 – Загальний вид датчика рівня JUMO MAERA S28

Основні характеристики датчика рівня зведено в таблицю 2.3.

Таблиця 2.3

Характеристики давача рівня JUMO MAERA S28

Параметри	Значення
Діапазон вимірів	від 0... 2,50 бар до 0...25 бар 0°C...50°C
Тип сенсора	Кремнієвий сенсор з розділовою мембраною з нержавіючої сталі синтетичне масло
Припустимий цикл навантаження	> 10 мільйонів, 0 ... 100 % діапазона
Положення при монтажі	вертикально/вішається на кабелі
Номінальні умови	згідно DIN 16086 і DIN EN 60770
Діапазони припустимих температур середи, що змінюється	0 ... 50 °C Не допускати вмрзання пристрою в середовище, що вимірюється! Можливе обмеження в залежності від вимірювального середовища.
Ступінь захисту	IP68, на глибині до 400м
Макс. Витрата струму	≤ 30 мА

Розглянемо термоелемент, що угвинчується. Для вимірювання температури розплаву використовуються переважно в полімерній промисловості з метою контролю температури маси. Загострені або плоскі вимірювальні наконечники забезпечують оптимальний вимір температури. Компенсаційний провід розрахований на робочий температурний діапазон від -40 до +600 ° C.

У вимірювальній частині використані термопари (елементи), що відповідають стандарту DIN EN 60 584 або DIN 43 710. Загальний вигляд датчика наведено на рис. 2.11.

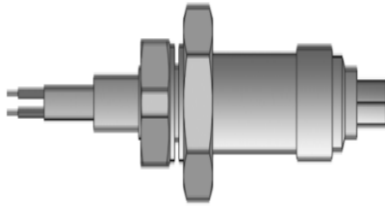


Рисунок 2.11 – Термоелемент, що угвинчується, для виміру температури розплаву

Основні технічні характеристики зведено в таблицю 2.4.

Таблиця 2.4

Основні характеристики термоелемента, що угвинчується, для виміру температури розплаву

Параметри	Значення
Приєднання	3 кінців проводів знята ізоляція, в комплект поставки входить кабельний затиск або багатополоснове рознімне приєднання, IP50
Приєднувальний провід	PTFE, температура довкілля від -40 ... +260 °C Металеве обплетення, температура навколишнього середовища від -20 ... +400 °C
Підключення до процесу	Різьба, нержавіюча сталь 1.4571
Захисна трубка	Нержавіюча сталь 1.4571
Вимірювальна частина	Ізольований монтаж 1x Fe-CuNi “J”, DIN EN 60 584, клас 2, робоча температура від -40 до +400 °C (+600 °C) 1x Fe-CuNi “L”, DIN EN 60 584, клас 2, робоча температура від -40 до +400 °C (+600 °C) 1x NiCr-Ni “K”, DIN EN 60 584, клас 2, робоча температура від -40 до +400 °C (+600 °C)

На міжнародному ринку існує багато фірм, які виробляють великий асортимент датчиків, вибір датчика для вирішення того чи іншого завдання залежить від вхідних параметрів і точності вимірювань та способів захисту і підключення до ПЛК.

2.3 Програмовані логічні контролери їх структура та характеристика

ПЛК – програмований логічний контролер – це мікропроцесорний пристрій, призначений для збору, перетворення, обробки, зберігання інформації та вироблення команд управління, що має кінцеву кількість входів і виходів, підключених до них датчиків, ключів, виконавчих механізмів до об'єкта управління, і призначений для роботи в режимах реального часу. Спрощено структуру застосування ПЛК представлено на рис. 2.12.



Рисунок 2.12 – Спрощена структурна використання ПЛК

Принцип роботи ПЛК дещо відрізняється від «звичайних» мікропроцесорних пристроїв. Програмне забезпечення універсальних контролерів складається з двох частин. Перша частина – це системне програмне забезпечення. Проводячи аналогію з комп'ютером можна сказати, що це операційна система, тобто управляє роботою вузлів контролера, взаємозв'язком складових частин, внутрішньою діагностикою. Системне програмне забезпечення ПЛК розташоване в постійній пам'яті центрального процесора і завжди готове до роботи. При вмиканні живлення ПЛК готовий взяти на себе управління

системою вже через кілька мілісекунд. ПЛК працюють циклічно за методом періодичного опитування вхідних даних. Робочий цикл ПЛК складається з таких фаз:

- а) опитування входів;
- б) виконання програми;
- в) встановлення значень виходів;
- г) допоміжні операції (діагностика, підготовка даних для відладчика, візуалізація, і т.д).

Виконання фази «а» забезпечується системним програмним забезпеченням. Після чого управління передається прикладній програмі, тій програмі, яка записана в пам'ять програмістом, за цією програмою контролер виконує всі необхідні операції для виконання вимог технологічного процесу, а по її завершенню управління знову передається системному рівню. За рахунок цього забезпечується максимальна простота побудови прикладної програми – її розробник не повинен знати, як здійснюється управління апаратними ресурсами. Необхідно знати, з якого входу приходить сигнал і як на нього реагувати на виходах. Час реакції на подію буде залежати від часу виконання одного циклу прикладної програми. Визначення часу реакції – часу від моменту події до моменту видачі відповідного керуючого сигналу – пояснюється на рис. 2.13.

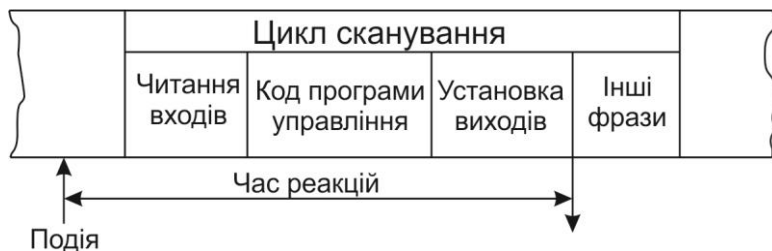


Рисунок 2.13 – Визначення часу реакції ПЛК

Маючи пам'ять, в якій знаходиться програмний код, ПЛК в залежності від вхідних значень подій, здатний реагувати по-

різному в залежності від алгоритму роботи. Можливості перепрограмування, управління за часом, розвинені обчислювальні можливості, включаючи цифрову обробку сигналів, піднімають ПЛК на більш високий рівень на відміну від простих комбінаційних автоматів.

Розглянемо вхід і вихід ПЛК. Існує три види входів: дискретні, аналогові і спеціальні:

- один дискретний вхід ПЛК здатний приймати один бінарний електричний сигнал, що описується двома станами – включений або виключений;
- дискретні входи (загального виконання) контролерів зазвичай розраховані на прийом стандартних сигналів з рівнем 24В постійного струму. Типове значення струму одного дискретного входу (при вхідній напрузі 24В) становить близько 10мА;
- аналоговий електричний сигнал відображає рівень напруги або струму, відповідний деякій фізичній величині, в кожен момент часу. Це може бути температура, тиск, вага, положення, швидкість, частота і т.д.

Використання даних входів і виходів залежить від типу розв'язуваної задачі, специфіки технологічного процесу та в першу чергу від типу датчиків і виконуючого обладнання.

Оскільки ПЛК є цифровою обчислювальною машиною, аналогові вхідні сигнали обов'язково піддаються аналого-цифровому перетворенню (АЦП). В результаті, утворюється дискретна змінна певної розрядності. Як правило, в ПЛК застосовуються 8-12 розрядні перетворювачі, що в більшості випадків, виходячи з сучасних вимог до точності управління технологічними процесами, є достатнім. Крім цього АЦП вищої розрядності не виправдовують себе, в першу чергу через високого рівня індустріальні перешкоди, притаманні умовам роботи контролерів.

Практично всі модулі аналогового вводу є багатоканальними. Вхідний комутатор підключає вхід АЦП до необхідного входу модуля.

Стандартні дискретні і аналогові входи ПЛК здатні задовольнити більшість потреб систем промислової автоматики. Необхідність застосування спеціалізованих входів виникає у випадках, коли безпосередня обробка деякого сигналу програмно складна, наприклад, вимагає багато часу.

Найбільш часто ПЛК оснащуються спеціалізованими рахунковими входами для вимірювання тривалості, фіксації фронтів і підрахунку імпульсів.

Наприклад, при вимірі положення та швидкості обертання валу дуже поширені пристрої, що формують певну кількість імпульсів за один оборот – поворотні шифратори. Частота проходження імпульсів може досягати декількох мегагерц. Навіть якщо процесор ПЛК має достатню швидкодію, безпосередній підрахунок імпульсів в призначеній для користувача програмі буде марнотратним за часом. Тут бажано мати спеціалізований апаратний вхідний блок, здатний провести первинну обробку і сформувати необхідні для прикладної задачі величини. Другим поширеним типом спеціалізованих входів є входи, здатні дуже швидко запускати задані користувальницькі завдання з перериванням виконання основної програми – входи переривань.

Дискретний вихід також має два стани – включений і виключений. Вони потрібні для управління електромагнітними клапанами, котушками, пускачами, світловими сигналізаторами і т.д. Загалом сфера їх застосування величезна і охоплює майже всю промислову автоматику.

Конструктивно ПЛК підрозділяються на моноблочні, модульні і розподілені. Моноблочні мають фіксований набір входів виходів, приклад наведено на рис. 2.14.



Рисунок 2.14 – Моноблочний ПЛК фірми Овен модель ПЛК63

У модульних ПЛК модулі входів-виходів встановлюються в різному складі та кількості в залежності від поставленого завдання, яке необхідно вирішити. Приклад модульної ПЛК наведено на рис. 2.15.



Рисунок 2.15 – Модульний ПЛК

У розподілених системах модулі або навіть окремі входи-виходи, які утворюють єдину систему управління, можуть бути рознесені на значні відстані. Дане рішення дозволяє зробити автоматизацію великих технологічних процесів, які можуть перебувати в різних виробничих ділянках на великих відстанях.

Архітектурою контролера називають набір його основних компонентів і зв'язків між ними. Типовий склад ПЛК включає центральний процесор, пам'ять, мережеві інтерфейси і пристрої введення-виведення, що наведено на рис. 2.16.



Рисунок 2.16 – Типова структура ПЛК

Іноді ця конфігурація доповнюється пристроєм для програмування і пультом оператора, пристроями індикації, рідше – принтером, клавіатурою, мишею або трекболом.

Процесорний модуль включає в себе мікропроцесор (центральный процесорний пристрій – ЦПП), пристрої, що запам'ятовують, годинник реального часу і сторожовий таймер. Терміни «мікропроцесор» і «процесор» в даний час стали синонімами, оскільки всі процесори, що зараз випускаються, виконуються у вигляді СБИС, тобто є мікропроцесорами.

Основними характеристиками мікропроцесора є розрядність (в ПЛК використовуються 8-ми, 16-ти і 32-розрядні мікропроцесори), тактова частота, архітектура, наявність операцій з плаваючою точкою, типи підтримуваних портів введення-виведення, температурний діапазон працездатності і споживана потужність.

Продуктивність мікропроцесорів з однієї і тієї ж архітектурою пропорційна тактовій частоті. Більшість контролерів використовують мікропроцесори зі скороченим набором команд (RISC - Reduced Instruction Set Computing), в яких використовується невелика кількість команд однакової довжини і велика кількість регістрів. Скорочений набір команд

дозволяє будувати більш ефективні компілятори і конвеєр процесора, здатний за кожен такт видавати результат виконання чергової команди.

Для контролерів, які виконують інтенсивну математичну обробку даних, важлива наявність математичного співпроцесора (допоміжного процесора, що виконує операції з плаваючою точкою) або сигнальних процесорів, в яких операції типу виконуються за один такт. Сигнальні процесори дозволяють прискорити виконання операцій згортки або швидкого перетворення Фур'є.

Ємність пам'яті визначає кількість змінних (тегів), які можуть бути оброблені в процесі функціонування ПЛК. У мікропроцесорах час доступу до пам'яті є одним з істотних факторів, що обмежують швидкодію. Тому пам'ять поділяють на кілька рівнів ієрархії, залежно від частоти використання даних, що зберігаються в ній, та швидкодії. Ієрархія пам'яті відноситься до істотних характеристик архітектури процесора, оскільки вона дозволяє знизити негативний вплив повільної пам'яті на швидкодію мікропроцесора. Основними типами пам'яті є постійний запам'ятовуючий пристрій (ПЗУ), оперативний запам'ятовуючий пристрій (ОЗП) і набір регістрів. Регістри є найбільш швидкодіючими елементами пам'яті, оскільки вони використовуються арифметико-логічним пристроєм (АЛП) для виконання елементарних команд процесора. ПЗУ використовують для зберігання інформацією, що рідко змінюється, такої, як операційна система, драйвери пристроїв, завантажувач, виконуваний модуль програми користувача. ОЗП використовується для зберігання даних, які багаторазово змінюються в процесі роботи контролера, наприклад, значення тегів, результати проміжних обчислень, діагностична інформація, масиви, виведені на графіки, дані для відображення на дисплеї.

Як ПЗУ (або ROM – «Read Only Memory») зазвичай використовується пам'ять, що електрично стирається та перепрограмується (EEPROM – «Electrically Erasable

Programmable ROM»)). Різновидом EEPROM є флеш-пам'ять, принцип дії якої базується на зберіганні заряду в конденсаторі, утвореному плаваючим затвором і підкладкою МОП-транзистора. Особливістю флеш-пам'яті є її енергонезалежність, тобто збереженість даних при вимкненому живленні. Стирання і перезапис у флеш-пам'яті виконується не окремими осередками, а великими блоками, тому вона отримала назву, що походить від англійського «flash» – «спалах». Недоліком всіх ПЗУ є низька швидкодія.

Кількість циклів запису інформації у флеш-пам'ять обмежена і складає кілька десятків тисяч разів. За конструктивним виконанням і інтерфейсів флеш-пам'ять поділяється на Compact Flash (CF), Memory Stick, Secure Digital (SD), MultiMediaCard (MMC), RS-MMC, SmartMedia Card (SMC), USB-flash. Флеш-пам'ять може бути впаяна в друковану плату або бути знімною.

Як ОЗП сучасні мікропроцесори використовують статичну пам'ять (SRAM – Static Random Access Memory) і динамічну (DRAM – «Dynamic Random Access Memory»), SDRAM («Synchronous DRAM»). SRAM виконується на тригерах, інформація в яких зберігається необмежено довго при наявності живлення. У динамічній пам'яті інформація зберігається на конденсаторах і тому DRAM вимагає періодичної регенерації (перезарядки конденсаторів). До недоліків триггерної пам'яті відноситься її висока вартість, пов'язана з низькою щільністю компоновки тригерів на кристалі, і мале відношення ємності до ціни. Перевагою є висока швидкодія, що досягає гігагерц, в той час як пам'ять на конденсаторах не може працювати на частотах вище сотень герц. Обидва типи пам'яті (DRAM і SRAM) не можуть зберігати інформацію при відключенні живлення ПЛК. Тому деякі типи ПЛК використовують батарейне живлення пам'яті для збереження працездатності системи автоматизації після короткочасного відключення живлення.

Моноблочні і модульні контролери використовують, як правило, паралельну шину для обміну даними з модулями

вводу-виводу, що дозволяє на порядок підвищити швидкодію їх опитування у порівнянні з послідовною шиною. Паралельні шини можуть бути стандартними (ISA, PC/104, PCI, ComactPCI, VME, CXM). Послідовна шина контролера (на основі інтерфейсу RS-485) використовується для підключення до нього віддалених (розподілених) модулів вводу-виводу.

Програмування контролерів малої потужності виконується за допомогою кнопок, розташованих на лицьовій панелі або за допомогою переносного пульта для програмування. В якості пульта останнім часом використовується комп'ютер формату «ноутбук». Програмування потужних контролерів виконується за допомогою персонального комп'ютера, на якому встановлюється спеціальне програмне забезпечення, наприклад CoDeSys або ISaGRAF, яке виконує трансляцію технологічного мови стандарту МЕК 61131-3 у виконуваний код процесора, який завантажується в ПЗУ ПЛК, наприклад, через порт Ethernet.

Сторожовий таймер (Watchdog Timer – WDT) являє собою лічильник, який рахує імпульси тактового генератора і в нормальному режимі періодично скидається (перезапускається) працюючим процесором. Якщо процесор «зависає», то сигнали скидання не надходять в лічильник, він продовжує вважати і при досягненні деякого порога виробляє сигнал «Скидання» для перезапуску процесора, що «завис».

Годинники реального часу (РЧ) – це кварцові годинники, які живляться від батарейки і тому продовжують йти при вимкненому ПЛК. Годинники РЧ використовуються, наприклад, для управління вуличним освітленням в залежності від часу доби, в системах охорони об'єктів та інших випадках, коли необхідна прив'язка даних або подій до астрономічного часу.

Процесорний модуль ПЛК виконує такі завдання:

- збирає дані з модулів введення в пам'ять і відсилає дані з пам'яті в модулі виводу;

- виконує обмін даними з пристроєм для програмування контролера;
- видає мітки годин реального часу;
- здійснює обмін даними з промислової мережею;
- реалізує стек протоколів промислової мережі (для цієї мети можуть використовуватися допоміжні комунікаційні процесори);
- виконує початкове завантаження і виконання операційної системи;
- виконує завантажувальний модуль користувальницької програми системи автоматизації;
- управляє актами обміну з пам'яттю.

Однією з тенденцій у розвитку ПЛК є використання процесорних модулів різної потужності для одного конструктиву контролера. Це дозволяє отримати серію контролерів різної потужності і тим самим покрити більший сегмент ринку, а також виконати модернізацію (upgrade) контролерів, куплених споживачами, шляхом заміни всього одного модуля.

До основних характеристик процесорного модуля відносяться:

- тип операційної системи (Windows CE, Linux, DOS, OS-9, QNX і ін.);
- наявність виконавчого середовища для стандартної системи програмування на мовах MEK 61131-3;
- типи підтримуваних інтерфейсів (RS-232, RS-422, RS-485, CAN, USB, Ethernet і ін.);
- типи підтримуваних мереж (Modbus RTU, Modbus TCP, Ethernet, Profibus, CANopen, DeviceNet і ін.);
- можливість підключення пристроїв індикації або інтерфейсу оператора (світлодіодного або РКІ, клавіатури, миші, дисплея з інтерфейсами VGA, DVI або CMOS, LVDS, трекболу і ін.);
- розрядність (8, 16, 32 або 64 біта);
- тактова частота мікропроцесора та пам'яті;

- час виконання команд;
- обсяг, ієрархія і типи пам'яті (ОЗУ, кеш, ПЗУ – флеш, знімна флеш і ін.);
- типи вбудованих функцій (ПД-регулятор, лічильники, ШПМ, алгоритми позиціонування і управління рухом та ін.);
- бренд виробника (Intel, AMD, Atmel, Motorola, RealLab, і ін.).

Швидкодію процесорного модуля ПЛК зазвичай оцінюють за часом виконання логічних команд, оскільки вони найбільш поширені при реалізації алгоритмів керування.

Величезна різноманітність завдань, покладених на ПЛК, і сильна залежність ціни від потужності контролера призвели до великої різноманітності використовуваних мікропроцесорів, від простих і дешевих 8-розрядних Atmel і Microchip до найвисокопродуктивніших мікропроцесорів серії Intel Pentium, включаючи двоядерні і чотириядерні процесори.

Восьмирозрядні мікропроцесори користуються великим успіхом у автономних ПД-контролерах і мікро-ПЛК для нескладного алгоритмічного управління верстатами, теплицями, невеликими технологічними апаратами в яості міжмережевих шлюзів. Їх перевагою є висока надійність, пов'язана з граничною простотою програмного забезпечення.

Зазвичай мікропроцесори, використовувані в ПЛК, на кілька поколінь відстають від процесорів офісних персональних комп'ютерів (ПК) у зв'язку з відносно малим об'ємом ринку ПЛК, який не забезпечує окупність розробки нового контролера за період зміни поколінь мікропроцесорів.

Джерело живлення. Стандартними напругами живлення ПЛК є напруги 12В, 24В і 48В. Джерелом електричної енергії зазвичай є промислова мережа 220В, 50Гц. У разі розподілених систем автоматизації джерело живлення може бути розташовано далеко від ПЛК, тому напруга на клеммах ПЛК або модулях введення-виведення може сильно відрізнятися від напруги джерела живлення внаслідок падіння напруги на опорі кабелю. Для вирішення цієї проблеми кожен ПЛК або кожен модуль віддаленого вводу забезпечуються вбудованим стабілізатором

напруги, який забезпечує нормальне їх функціонування в діапазоні напруг від 10В до 30В.

Низька напруга живлення дозволяє жити контролери від акумуляторів бортових мереж транспортних засобів або переносних акумуляторів.

В ПЛК іноді використовують батарею для живлення годин реального часу (які повинні функціонувати при вимкненому ПЛК) та для збереження інформації в ПЗУ на час аварійних перерв живлення.

Продуктивність ПЛК оцінюється за такими параметрами:

- тривалість контролерного циклу (період зчитування значень з каналів введення, обробки в процесорі і запису в канали виведення);
- час виконання команд (окремо логічних, з фіксованою і з плаваючою точкою);
- пропускна здатність шини між контролером і модулями вводу-виводу;
- пропускна здатність промислової мережі;
- час циклу опитування всіх контролерів в одномастерній мережі або цикл звернення маркера для багатомастерних мереж з маркером;
- час реакції.

Контролер в системах автоматизації виконує циклічний алгоритм, що включає введення даних і розміщення їх в ОЗУ, обробку даних та виведення. Тривалість контролерного циклу (його ще називають робочим циклом) залежить від кількості модулів вводу-виводу і наявності в них замаскованих (виключених з процедури обміну) входів-виходів, тому розраховується для кожної конфігурації автоматизованої системи окремо. За умов інтенсивної математичної обробки даних (наприклад, при цифровій фільтрації, інтерполяції або ідентифікації об'єкта управління в режимі нормального функціонування системи) тривалість контролерного циклу істотно залежить від швидкодії процесорного модуля. У контролерний цикл входить також обслуговування апаратних

ресурсів ПЛК (забезпечення роботи системних таймерів, оперативне самотестування, індикація стану), контроль часу циклу, мережевий обмін, управління багатозадачністю, відображення процесу виконання програми на дисплеї і т. п.

Перед початком роботи ПЛК виконує завантаження операційної системи і програми користувача в ОЗП та ПЗУ, початкове тестування апаратури. ПЛК зазвичай може працювати в режимі налагодження, покрокового виконання програми, перегляду і редагування значень змінних і т.п.

В процесі функціонування ПЛК дані, введені з пристроїв введення, розташовуються в ОЗП і протягом робочого циклу контролера не змінюються. Пряме читання входу під час виконання одного циклу не виконується. Це прискорює процес обробки даних і виключає непередбачені ситуації.

У системах з розподіленими по об'єкту модулями вводу-виводу тривалість контролерного циклу може визначатися пропускнуою спроможністю промислової мережі, що в ряді випадків є обмеженням на граничну кількість модулів введення-виведення.

Вимоги до тривалості контролерних циклу істотно залежать від сфери застосування ПЛК. При управлінні тепловими процесами тривалість циклу може становити одиниці і десятки секунд, в задачах для управління верстатами вона вимірюється мілісекундами, при опитуванні датчиків температури на елеваторі контролерний цикл вимірюється цілодобово.

Час реакції контролера – це інтервал часу від моменту появи впливу на систему (з боку модулів введення або оператора) до моменту вироблення відповідної реакції. Час реакції залежить від тривалості робочого циклу контролера, який визначається швидкістю модулів вводу-виводу і продуктивністю процесора.

У контролерах для відповідальних застосувань можуть бути передбачені такі функції самодіагностики:

- виявлення помилок центрального процесора;

- сигналізація про спрацювання сторожового таймера;
- виявлення відмови батареї або джерела живлення;
- виявлення збою пам'яті;
- перевірка програми користувача;
- виявлення виходу з ладу запобіжника;
- виявлення обриву або к.з. в ланцюзі датчика і навантаження.

У контролерах для систем протиаварійного захисту (ПАЗ) та сигналізації, а також для небезпечних промислових об'єктів може бути передбачена можливість резервування окремих частин системи: промислової мережі, процесорного модуля або контролера, джерела живлення, мережевого сервера, замкнутих контурів автоматичного регулювання, модулів вводу-виводу. Об'єктом резервування зазвичай є найбільш відповідальна або найбільш ненадійна частина системи.

Можливість гарячої заміни елементів системи (тобто без відключення живлення) досягається одночасно апаратними та програмними засобами. Апаратно передбачається незалежність початкового стану пристрою від черговості подачі сигналів на його клеми в процесі заміни; програмно забезпечується можливість тимчасової відсутності компонента системи без її зависання або переходу в аварійні режими.

Надійність контролерів характеризується напрацюванням на відмову, яка визначається як відношення сумарного часу працездатного стану контролера до математичного сподівання числа його відмов протягом цього часу або часом від початку експлуатації до першої відмови. Надійність пов'язана з припустимими механічними перевантаженнями – амплітудою вібрації в необхідному діапазоні частот, допустимим прискоренням при ударі.

При створенні системи керування технологічним процесом, завжди існує проблема взаєморозуміння програміста і технологів. Що викликано неможливістю технолога формалізувати опис алгоритму. Внаслідок цього програмісту ПЛК необхідно знати всі особливості технологічного процесу, розробити алгоритм роботи, описати всі вхідні і вихідні

значення та реакцію ПЛК на їх значення, а лише потім писати програму. Найчастіше при такому підході програміст залишається єдиною людиною, здатною розуміти та пояснити логіку роботи програми. Така ситуація сприяла створенню технологічних мов програмування ПЛК, доступні інженерам-програмістам і технологам та максимально спрощує процес програмування.

За останній п'ять років з'явилося кілька технологічних мов. Внаслідок цього Міжнародною Електротехнічною Комісією розроблений стандарт МЕК-61131-3, який концентрує все передове в області мов програмування для систем автоматизації технологічних процесів. Цей стандарт вимагає від різних виробників ПЛК застосовувати єдиний підхід до синтаксису команд, які є однаковими і за зовнішнім виглядом, і за дією.

Даний стандарт специфікує 5 наступних мов програмування ПЛК:

- Sequential Function Chart (SFC) – мова послідовних функціональних блоків;
- Function Block Diagram (FBD) – мова функціональних блокових діаграм;
- Ladder Diagrams (LAD) – мова релейних діаграм;
- Statement List (STL) – мова структурованого тексту, мова високого рівня, нагадує собою Паскаль;
- Instruction List (IL) – мова інструкцій, це типовий асемблер з акумулятором і переходах мітками.

Мови LAD або KOP (з німецького Kontaktplan) схожі на електричні схеми релейної логіки. Тому інженеру, який знає мови програмування, неважко написати програму. Мова FBD нагадує створення схем на логічних елементах. У кожній з цих мов є свої позитивні і негативні моменти. Тому при виборі фахівці ґрунтуються в основному на особистому досвіді. Хоча більшість програмних комплексів дають можливість переконвертувати вже написану програму з однієї мови в іншу.

Деякі завдання просто вирішуються на одній мові, а на іншій доведеться зіткнутися з деякими труднощами.

Найбільшого поширення в даний час отримали мови LAD, STL і FBD.

Більшість фірм виробники ПЛК традиційно мають власні фірмові напрацювання в області інструментального програмного забезпечення такі як «Concept» Schneider Electric, «Step 7» Siemens.

SCADA система Trace Mode 6 підтримує більш 2567 ПЛК та лічильників електроенергій, газу та води. У список підтримуваних системою ПЛК входять пристрої розроблені фірмами ABB, Accurate Technologies, Actaris, Action Instruments, Advantech, AEG, Airmar, Algodue, Albatros, Rockwell Automation, Allen Bradley, Alerton, АМЕТЕК Thermox, Alpha-T, April, DEP, EMIS, GE Fanuc і т.д.

Розглянемо як приклад можливість 761 Series однопозиційного MICRO Plus контролера фірми Foxboro (USA), патент USA 1296. Контролер 761 Series розширює можливості базового 760 Series і має вбудовану можливість сприяння високорозвиненої стратегії управління для складних замкнутих систем. Загальний вигляд ПЛК 761 Series наведено на рис. 2.17.



Рисунок 2.17 – Загальний вигляд ПЛК 761 Series

Основні функції 761 ПЛК Series:

- динамічна компенсація з Lead/Lag, Impulse, і функцій «мертвого часу»;
- сигнали, що призначаються;
- алгебраїчні розрахунки;
- більш гнучкі конфігурації блоків з внутрішньої маршрутизацією;
- розширені послідовні зв'язки;
- Pulse Driven Input;
- можливість використання булевої логіки;
- рН дисплей;
- можливе відображення незалежної змінної;
- просте пакетне управління;
- єдина станція управління;
- опція для відображення другої змінної.

Властивості, які сумісні з 760 Series:

- контроль ЕХАСТ;
- повністю інтерактивний дисплей;
- вхідний сигнал/зовнішнє живлення від 10мА до 50мА;
- вхідний/вихідний сигнал від 10мА до 50мА;
- стандартний комп'ютерний інтерфейс для роботи та програмування;
- незалежна пам'ять;
- помилка живлення і логіка відновлення;
- передня панель і клавіатура для програмування й інтерфейс оператора;
- корпус або монтажне рішення;
- легке фізичне дооснащення рішень, що вже існують;
- джерела внутрішнього сигналу для калібрування;
- passcode безпеки.

ПЛК використовує технологію ЕХАСТ управління, яка дозволяє зробити поточні коригування фактично в реальному масштабі часу без зупинки динаміки технологічного процесу. Це відміна від інших «самоналагоджувальних» ПЛК. Під час роботи ПЛК, постійно скануючи змінні процесу, ЕХАСТ управління, ініціює коригувальні дії негайно при виявленні збою

логіки програми. Користувач може налаштувати пріоритет реакцій ПЛК відповіді вказавши бажане демпфірування і перегулювання до навантаження чверть амплітуди загасання. Значення одержувані за технологією EXACT можуть бути оброблені інженером, і за допомогою клавіатури внесені зміни в логічні функції ПЛК, що дозволяє використовувати його як передовий PID. Забезпечення даного підходу і точного контролю дозволяє параметрам для виконання технологічного процесу уникнути пошкодження при збої.

ПЛК 761 Series оснащений RS-485 послідовним портом для передачі даних з хост-комп'ютерів, або безпосередньо, або через RS-485/RS-232, або еквівалентним ним. Якщо в системі використовувати конвертер модель Foxboro F6501A, він дозволить розмістити до 90 ПЛК з одним зв'язком порту, що приймає.

Функція копіювання дозволяє конфігурацію одного ПЛК дублювати для використання в іншому. Це досягається використанням двох NOVRAMS (модуль незалежної пам'яті довільного доступу). Тобто після виключення живлення, дана процедура, працює таким чином:

- видалити налаштований NOVRAM з ПЛК;
- встановити копію;
- підключити конфігурований NOVRAM і другий NOVRAM (буде сконфігурований) в копію приналежності;
- потім увімкнути живлення, а другий NOVRAM скопіюється для використання в перший ПЛК.

Внаслідок використання даного підходу може бути налаштовані основні параметри в лінійки ПЛК, а потім індивідуально для кожного внесені зміни в залежності від специфіки технологічного процесу.

2.4. Контрольні питання

1. Дайте пояснення терміну автоматизована системи керування технологічними процесами.

2. Що є об'єктом керування в АСК ТП?
3. Які основні потоки інформації в АСК ТП?
4. Наведіть приклад узагальненої структури АСК ТП.
5. У чому особливість АСК ТП?
6. Інформаційні функції АСК ТП?
7. Наведіть приклад структури централізованої АСК ТП.
8. Наведіть приклад структури супервізорної АСК ТП.
9. На базі яких об'єктів будуються розподілена АСК ТП?
10. Які існують структури розподіленої АСК ТП?
11. Дайте пояснення терміну програмно-технічні комплекси (ПТК).
12. Наведіть приклад структури ПТК.
13. На базі яких об'єктів будується структура складного ПТК?
14. Наведіть приклад промислових мереж застосовуваними в системах автоматизації.
15. Які види забезпечення необхідно реалізувати для ефективної роботи АСК ТП?
16. Сформулювати вимоги, необхідні для організації групи оперативного технологічного персоналу.
17. Які основні функції підрозділа технічного обслуговування на стадії створення АСК ТП?
18. Дайте пояснення терміну інформаційне забезпечення.
19. Які принципи організації управління технологічним процесом необхідно враховувати?
20. Наведіть приклад процесів обробки інформації для автоматизованих систем?
21. Що відноситься до основних характеристик потоку інформації?
22. Що відноситься до основних характеристик каналу зв'язку?
23. Які загальні технічні вимоги ставляться до АСК ТП з інформаційного забезпечення?
24. Які ставляться вимоги до надійності АСК ТП?
25. Дайте пояснення терміну давач (сенсор).
26. Наведіть класифікацію давачів.

27. Які три класи датчиків ви знаєте?
28. Принцип дії параметричних датчиків та омичних (резистивних) датчиків?
29. Принцип дії контактних датчиків та реостатних датчиків.
30. Принцип дії термометричних й індуктивних датчиків.
31. Принцип дії ємнісних та генераторних датчиків.
32. Принцип дії температурних та кремнієвих датчиків.
33. Для чого використовують термоіндикатори?
34. Інфрачервоні давачі (пірометри).
35. На чому базується дія п'єзоелектричних датчиків?
36. Оптичні (фотоелектричні) датчики.
37. На базі яких об'єктів будуються спрощена структурна використання ПЛК та приведіть типову структуру ПЛК?
38. Як визначається час реакції ПЛК?

3. TRACE MODE 6.0 В РОЗВ'ЯЗАННІ ЗАДАЧ АСК ТП

3.1. Галузь задач, які розв'язуються в SCADA TRACE MODE 6

Застосування системи SCADA TRACE MODE можливо в будь-яких сферах промислової діяльності людини. Наведемо приклади розв'язання задач, в яких використовується дана система на прикладі реальних об'єктів, в умовах виробництва.

Компанія ТОВ «НТ-ПРОЕКТ» (Київ) здала в промислову експлуатацію автоматизовану систему комерційного обліку природного газу і стічних вод в цеху очищення промислових, і стічних вод на черкаському ПАТ «АЗОТ» (Черкаси, Україна). Черкаське відкрите акціонерне товариство «Азот» – одне з найбільших підприємств України з виробництва мінеральних добрив, іонообмінних смол, капролактаму та іншої хімічної продукції, яку споживають в Україні, країнах СНД, Європи, Південно-Східної Азії і Латинської Америки. На частку ВАТ «Азот» в Україні припадає 15% виробництва аміаку, 24% виробництва карбаміду, а за виробництвом капролактаму та іонообмінних смол підприємство є монополістом.

Розроблена автоматизована система комерційного обліку природного газу і стічних вод ПАТ «АЗОТ» (Черкаси, Україна) призначена для:

- автоматизованого контролю та комерційного обліку споживання природного газу;
- комерційного обліку міських і промислових стічних вод ПАТ «АЗОТ»
- автоматизованого контролю за технологічними параметрами хлораторної, котельної і насосно-повітродувної станції;
- сигналізації про порушення технологічних параметрів у вигляді екранних повідомлень і звукових сигналів та SMS-повідомлень;
- автоматичного генерування звітів про роботу установки.

Нова система являє собою робочу станцію на базі промислового ПК iROBO-2000a з встановленим на ньому серверним програмним забезпеченням GSM MPB + SCADA TRACE MODE 6. До промислового комп'ютера підключені витратоміри, інтелектуальні датчики, комунікаційне обладнання та лінії зв'язку. В системі передбачено додаткове клієнтське робоче місце начальника зміни на базі NetLink Light SCADA TRACE MODE 6, а також віддалені робочі місця Web-користувачів, що забезпечуються web-сервером TRACE MODE Data Center.

Вимірювальне обладнання системи комерційного обліку газу і стічних вод вже працювало на ПАТ «АЗОТ» (Черкаси) з 2003 року, тому метою робіт 2015 року була глибока модернізація системи комерційного обліку, тобто створення в SCADA TRACE MODE 6 сучасного операторського інтерфейсу, якісної системи сигналізації та документування відхилень від заданих параметрів технологічного процесу, а також заміна застарілого комунікаційного обладнання, робочої станції, системи безперебійного живлення і т.д.

Нова автоматизована система комерційного обліку природного газу і стічних вод на основі SCADA TRACE MODE 6 включена в систему менеджменту якості ПАТ «АЗОТ» і сертифікована на відповідність стандарту ISO 9001. На рис. 3.1 наведено приклад рішення автоматизації в SCADA TRACE MODE 6 контролю параметра випаровування.

Впровадження автоматизованої системи комерційного обліку природного газу і стічних вод ПАТ «АЗОТ» (Черкаси, Україна) дозволило:

- підвищити ефективність контролю та комерційного обліку природного газу і стічних вод;
- здійснювати збереження історії процесу в базу даних;
- формувати звітну документацію з можливістю їх публікації в web, подальшої обробки та аналізу;

- технологічному персоналу оперативно реагувати на виникаючі аварійні ситуації при прокачуванні промислових і міських стоків;
- аналізувати і визначати причину виникнення аварійних ситуацій для цілей їх запобігання;
- автоматизувати господарський і комерційний облік газу, води та енергоресурсів;
- забезпечити автоматичне введення даних технологічних параметрів в ERP-системи КАСКП «ІТ-Підприємство».

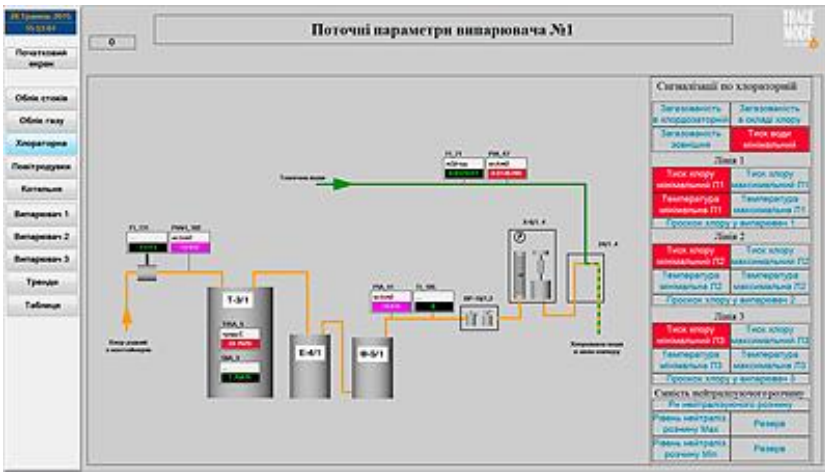


Рисунок 3.1 – Реалізація контролю випаровувань

Все це, за даними компанії ТОВ «НТ-ПРОЕКТ» (Київ), дозволило окупити витрати на модернізацію вже до кінця першого місяця експлуатації. Швидка окупність досягнута в основному за рахунок таких чинників:

- зниження числа аварій;
- скорочення технологічних простоїв;
- економії газу і води;
- зниження витрат на обслуговування;
- вивільнення матеріально-технічних ресурсів за рахунок оптимізації технології.

Фахівці компанії TEST TElemetry + STerowanie (Польща) застосували SCADA TRACE MODE (в системі управління виробництвом біогазового палива на підприємстві водопостачання та каналізації (Elblaskie Przedsiębiorstwo Wodociągów i Kanalizacji w Elblagu Sp. Z o.o.) в м. Ельблонг на північному сході Польщі.

Місто Ельблонг є польським лідером з виробництва енергії з поновлюваних джерел, включаючи біогаз і біомасу. На Ельблонгському підприємстві водопостачання і каналізації отриманий біогаз (метан) застосовується:

- для опалення;
- в якості джерела тепла в промислових процесах (опалення промисловий об'єктів);
- для виробництва електроенергії.

Місцеві енергетичні компанії охоче скуповують біопаливо у різноманітних виробників, серед постачальників екологічно чистого біопалива є і Elblaskie Przedsiębiorstwo Wodociągów i Kanalizacji w Elblagu Sp. z o.o.

Переробка відходів для виробництва біогазу відбувається в спеціальній герметичній ємності. Приклад реалізації системи моніторингу та управління переробки в SCADA TRACE MODE наведено на рис. 3.2

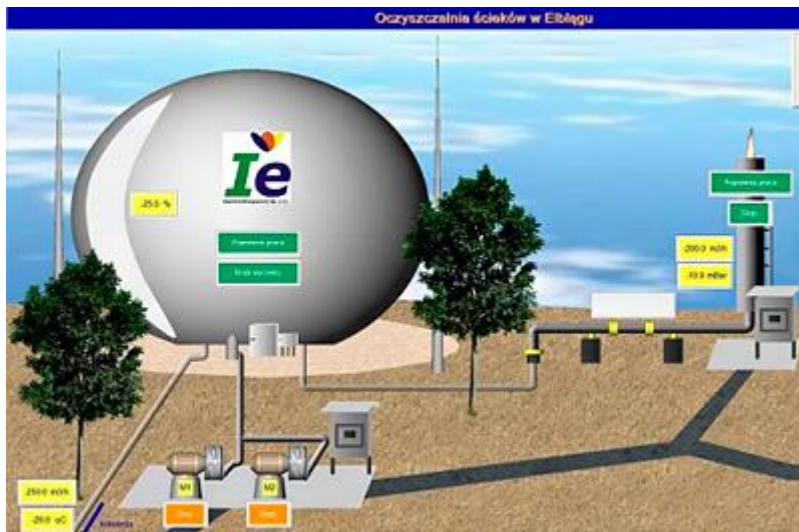


Рисунок 3.2 – SCADA TRACE MODE в рішеннях завдань моніторингу і управління виробництвом біопалива

Сировину змішують з іншими компонентами, наприклад, сухою травою, тирсою і органічними відходами. Суміш енергійно перемішують за допомогою спеціального обладнання. При розкладанні органічних речовин в результаті безповітряного бродіння, викликаного життєдіяльністю анаеробних бактерій, утворюється суміш газів. Значну частину суміші становить метан (60%), крім того утворюється вуглекислий газ (35%) і деякі інші хімічні сполуки. Відпрацьована органіка використовується на підприємстві в якості добрив.

Компанія TEST TElemetry + STerowanie є авторизованим системним інтегратором компанії Адастра в Польщі. Апаратний рівень системи управління виробництва біопалива представлений контролерами GE Intelligent Platforms і панелями Astraada.

На операторському рівні нової системи виробництва біопалива використовується SCADA TRACE MODE 6. Система дозволяє контролювати поточні значення:

- завантаження сировини,

- тиску або витоку газу,
- температури в ємності.

Нова система виробництва біопалива працює під управлінням SCADA TRACE MODE ДокМРВ+, що дозволяє вести власний архів значень, відображати необхідну інформацію на трендах і генерувати необхідну звітну документацію.

За заявою розробників системи виробництва біопалива, використання SCADA TRACE MODE знизило загальні витрати на реалізацію проекту в два-три рази (в порівнянні з аналогічними програмами рівня SCADA).

Як можемо бачити, застосування SCADA систем захоплює різні галузі виробництва і можуть використовуватися як система автоматизації та моніторингу за технологічними процесами, від видобутку природного газу, біопалива до важкої промисловості з складними системами АСК ТП.

3.2. Структура TRACE MODE 6, основні модулі й їх призначення

SCADA-система TRACE MODE розроблена і продовжує удосконалюватися фірмою-виробником AdAstra Research Group, LTD. Останній на даний момент продукт це 5-й реліз 6-ї версії TRACE MODE, який містить повний набір програмних засобів для створення АСК ТП і АСКП. SCADA-система TRACE MODE містить засоби розробки операторського інтерфейсу (SCADA / HMI), програмування контролерів (Softlogic), управління основними фондами (EAM), персоналом (HRM) і виробничими процесами (MES).

Для вивчення базових понять системи TRACE MODE, таких як проект, вузол, база каналів, шаблони екранів, FBD-програми, архіви та звіти тривоги зручніше скористатися більш ранньою версією, а потім перейти до версії, у якій більшість процедур реалізується в режимі автопобудови.

Всі програми, що входять в TRACE MODE, розділяються на дві групи (рис. 3.3): інструментальну систему розробки і виконавчі модулі (runtime). Як видно з рис. 3.3, інструментальна система розробки містить три редактори [1]: редактор бази каналів, редактор подання даних, редактор шаблонів.

У редакторі бази каналів створюється математична основа системи управління: описуються конфігурації всіх робочих станцій, контролерів і УСО, а також настроюються інформаційні потоки між ними. Тут же описуються вхідні і вихідні сигнали й їх зв'язок з пристроями збору даних і управління; задаються періоди опитування або формування сигналів, настроюються закони первинної обробки і управління, технологічні кордони, програми обробки даних та управління, здійснюється архівування технологічних параметрів, мережевий обмін, а також вирішуються деякі інші завдання.

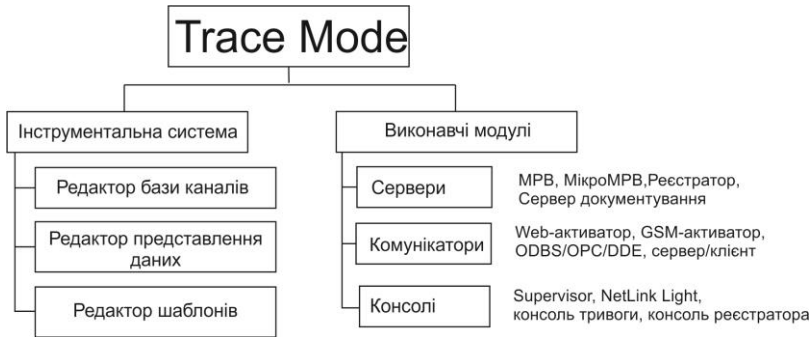


Рисунок 3.3 – Структура SCADA TRACE MODE 6.0

Результатами роботи в цьому редакторі є математична й інформаційна структури проекту АСК ТП, які включають в себе набір баз каналів і файлів конфігурації для всіх контролерів і операторських станцій (вузлів) проекту, а так само файл конфігурації всього проекту с розширенням smt (для версії 6 розширення – prtj). Всі інші файли проекту зберігаються в робочій директорії в каталозі, ім'я якого збігається з ім'ям файлу конфігурації.

У редакторі подання даних розробляється графічна частина проекту системи управління. Спочатку створюється статичний рисунок технологічного об'єкта, а потім поверх нього розміщуються динамічні форми відображення й управління. Серед цих форм присутні такі, як поля виведення числових значень, графіки, гістограми, кнопки, області введення значень і переходу до інших графічних фрагментів і т.д.

Крім стандартних форм відображення, TRACE MODE дозволяє вставляти в проекти графічні форми подання даних або управління, розроблені користувачами. Всі форми відображення інформації, управління й анімаційні ефекти зв'язуються з інформаційною структурою, розробленою в редакторі бази каналів. Для розробки шаблонів документів до складу інструментальної системи включений редактор шаблонів.

Виконавча система TRACE MODE включає в себе виконавчі модулі (монітори, МРВ) – програмні модулі різного

призначення, під керуванням яких в реальному часі виконуються складові частини проекту, що розміщуються на окремих комп'ютерах або в контролерах, призначені для роботи на всіх рівнях систем управління, про які говорилося вище.

Існує ряд програмних модулів, призначення яких чітко не прив'язане до функцій одного з перерахованих рівнів систем управління. До таких модулів відносяться (рис. 3.3):

- глобальний реєстратор;
- сервер документування;
- Web-активатор;
- GSM-активатор.

Вони можуть використовуватися для створення як оперативного, так і адміністративного рівнів систем управління.

Глобальний реєстратор служить для забезпечення надійного зберігання архівів ТП. Він архівує дані, що їх надсилали мережею моніторами реального часу (64000 параметрів з дискретністю 0,001с), забезпечує автоматичне відновлення даних після збою, а також може передавати архівні дані для перегляду моніторів SUPERVISOR. Глобальний реєстратор може також виступати як OPC-сервер і DDE-сервер та підтримує обмін з базами даних через ODBC.

Для документування технологічної інформації в TRACE MODE передбачений спеціальний модуль – сервер документування. Документування здійснюється за шаблонами, які створюються в редакторі шаблонів. Час або умова генерування документа, ім'я файлу шаблону, а також направлення виведення документа описуються в програмах документування – сценаріях.

Підготовка звітів (документів) найчастіше прив'язується до астрономічного часу. Наприклад, вони можуть генеруватися один раз на годину, один раз на добу, один раз на місяць або один раз в десять хвилин. Крім того, можна встановити режим підготовки документа один раз в зміну і потім описати розбивку доби на зміни.

Сервер документування NetLink Light використовується для вирішення завдання документування технологічної інформації. Він по команді МРВ, власним сценарієм або по команді оператора інтерпретує створені заздалегідь шаблони, запитує у МРВ необхідні дані та формує по них документи. Ці документи можуть бути роздруковані на принтері, відправлені по E-mail або опубліковані на Web-сервері. Утиліта консоль тривоги дозволяє переглядати звіт тривоги різних МРВ одного проекту. Для кожного звіту тривоги, що проглядається, створюється окреме вікно. У нього можна виводити інформацію з файлу звіту тривоги або повідомлення, що формуються МРВ.

Будь-яка робоча станція системи TRACE MODE може виступати в якості Web-сервера, що дозволяє управляти технологічним процесом через Інтернет (Internet) [1]. На віддаленому комп'ютері необхідно мати тільки доступ до мережі Інтернет і Web-браузер. Для реалізації даного режиму призначений модуль Web-активатора, який використовується в якості www-шлюзу для локальних систем АСК ТП на базі TRACE MODE або для додання функцій Web-сервера моніторів реального часу. Використання Web-активатора дозволяє швидко перетворити існуючі АСКТП і АСКП в Internet / Intranet-системи без переробки баз даних реального часу (баз каналів).

Доступ до даних реального часу через Web-активатор здійснюється за допомогою звичайного браузера, що працює під будь-якою операційною системою, що дозволяє запуск віртуальної Java-машини. Інформація про технологічний процес представляється користувачеві у вигляді анімованих мнемосхем, трендів і таблиць.

Зв'язок з серверами реального часу TRACE MODE може здійснюватися практично будь-якими доступними засобами, наприклад, через стільникову мережу стандарту GSM, інфрачервоний порт, мережу на основі інтерфейсу RS-232/485 або модем з використанням високонадійного протоколу TCP / IP. Можна здійснювати підключення і безпосередньо через

Internet. Для цього досить увійти в Internet і набрати IP-адресу сервера TRACE MODE – підключення відбудеться автоматично.

Для доступу до даних користувачеві досить набрати Web-адресу активатора і ввести пароль, тоді весь проект завантажується в віддалений комп'ютер у вигляді Java-аплету. Використання стандартної мови Java при написанні аплетів дозволяє реалізувати на віддалених комп'ютерах не тільки Windows, але і в інших операційних системах, наприклад, Unix, Linux, Mac OS і т.д., а так само ОС, що використовуються в кишенькових РС. Проект TRACE MODE надходить до користувача у вигляді Java-аплету, обсяг якого не перевищує 300 Кб, що дає можливість використовувати Web-активатор в мережах з низькою якістю зв'язку. Перевагою технології Java є також підвищена безпека.

При використанні Web-активатора не потрібна установка Web-серверів інших виробників (наприклад, MS IE), що вигідно відрізняє цю програму від рішень, застосованих в інших SCADA.

Для забезпечення мобільних користувачів АСК оперативною інформацією в режимі реального часу на базі TRACE MODE розроблений програмний продукт – GSM-активатор. Він призначений для дистанційного моніторингу та управління технологічними процесами, а також для отримання оперативної техніко-економічної інформації за допомогою понад портативних комп'ютерів handheld PC.

У реальному часі GSM-активатор може приймати інформацію від: 64000 датчиків, здійснювати супервізорне управління, отримувати техніко-економічну інформацію з баз даних через сервер, який використовує стандартні інтерфейси SQL/ODBC, OPC, DDE і т.д. Вся вхідна інформація відображається графічно в вигляді анімованих мнемосхем і трендів. GSM-активатор, що відноситься до нового класу систем оперативного управління, що відображають світову тенденцію до мініатюризації й автономізації комп'ютерних систем, може

бути використаний в якості персональної інформаційної системи керівника.

До GSM-активатора проявляють інтерес нафтові компанії, електричні та теплові мережі, комунальні й інші служби, що керують розподіленими у просторі об'єктами.

GSM-активатор придатний також до застосування в охоронних службах: отримання в реальному часі інформації про стан об'єкту, що охороняється, може стати основою успіху операції групи швидкого реагування, викликаній за тривоною.

Потрібно відзначити, що в останній версії TRACE MODE 6 всі редактори системи викликаються з однієї програми – Інтегрованого середовища розробки (IC). IC – єдина програмна оболонка, яка містить всі необхідні засоби для розробки проекту.

Всі змінні проекту, чого б вони не стосувалися – контролера, операторської станції, управління технічним обслуговуванням або виробництвом – зберігаються в єдиній базі даних проекту. Єдина база проекту усуває зайву роботу проектувальника із створення, підтримки та взаємного зв'язання багато в чому однакових баз змінних контролерів і ПК, характерну для систем попереднього покоління.

Логічна структура проекту повністю відокремлена від апаратної частини. Завдяки єдиному простору розподілених змінних, змінні з різних вузлів можуть зв'язуватися між собою також легко, як і в межах одного вузла, будь-які зміни, що вносяться до об'єкту, автоматично застосовуються скрізь, де він був задіяний.

3.3. Інсталяція TRACE MODE 6

Для інсталяцій SCADA системи TRACE MODE 6 необхідно зайти на сайт фірми AdAstra Research Group, LTD (офіційний сайт <http://www.adastra.ru>) і пройти реєстрацію. Після цього вам буде доступна для скачування версія TRACE MODE.

Після скачування буде доступний архів програми tm6b_ide_6.10.rar. Проведіть його розпакування на ваш розсуд і запустіть файл інсталяції setup.exe (версія файлу від 18.03.2003) або setup1.exe (версія файлу від 29.10.2015). На екрані з'явиться діалогове вікно установки, представлене рис. 3.4.

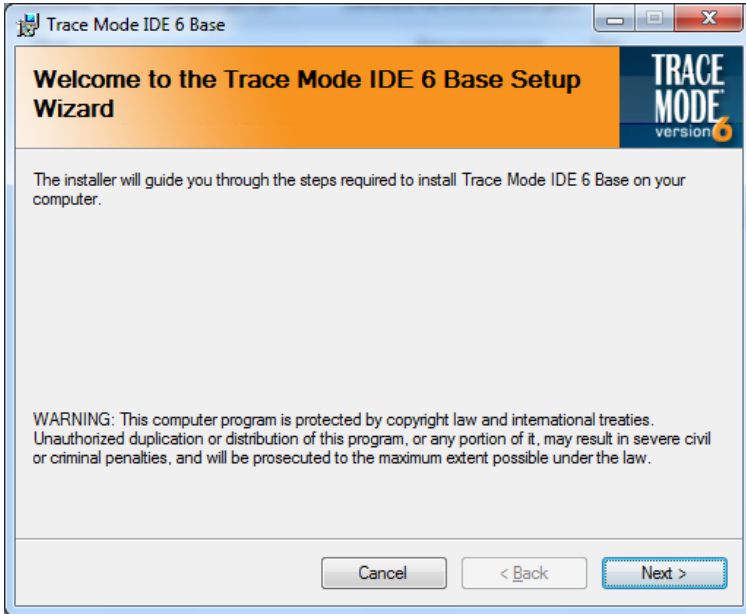
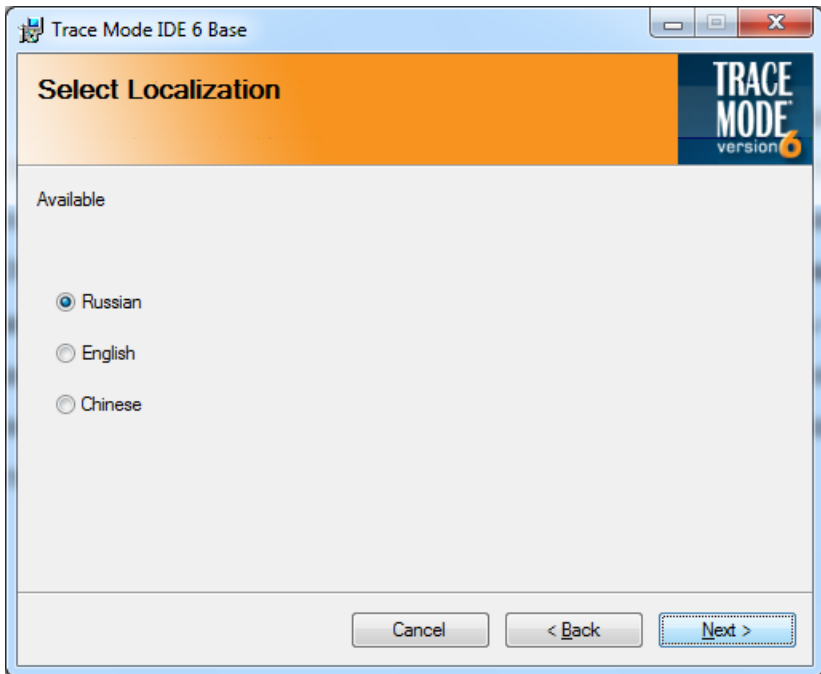


Рисунок 3.4 – Діалогове вікно інсталяції TRACE MODE 6

Натиснувши на кнопку «Next» ви перейдете у вікно ліцензійної угоди. Рекомендується прочитати ліцензійну угоду, в якій вказано, що ви згодні не використовувати освітню версію TRACE MODE 6 в комерційних цілях, а тільки для вивчення основних властивостей даної SCADA системи. Вказавши в вікні «Agree» ви погоджуєтеся з умовою ліцензійної угоди, і після цього вам буде доступна кнопка «Next», якщо ви не згодні з ліцензійною угодою, то інсталяція SCADA системи TRACE MODE 6 буде завершена. Після угоди ліцензійного договору вам

буде доступно вікно вибору мови інтерфейсу користувача представленого на рис. 3.5.



*Рисунок 3.5 – Вікно вибору мови інтерфейсу
SCADA системи TRACE MODE 6*

В даному випадку вибір мови інтерфейсу користувача залежить від знання тієї чи іншої мови. В даному навчальному посібнику буде обрана мова інтерфейсу користувача «Russian», так як цей посібник призначений для навчання іноземців основам SCADA системи TRACE MODE 6. Натисніть кнопку «Next» і вам буде доступно вікно вибору місця, куди буде інстальовано TRACE MODE 6, діалогове вікно представлено на рис. 3.6.

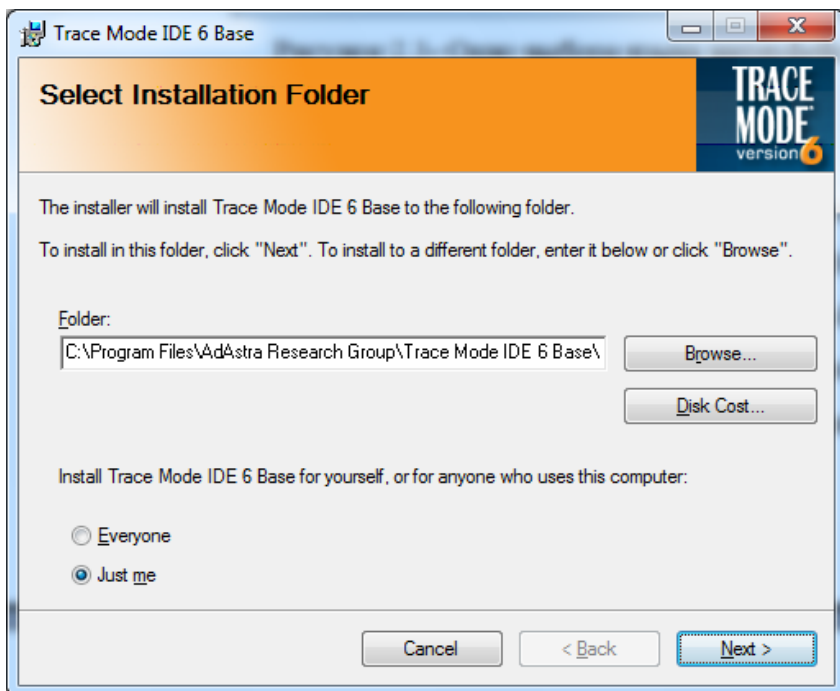


Рисунок 3.6 – Діалогове вікно вибору місця встановлення TRACE MODE 6

У вікні Folder, що редагується, вказується шлях до папки, куди буде встановлено дану систему, якщо у вас мало місця або ви хочете встановити систему в інше місце, то вам не обхідно натиснути на кнопку «Browse» і вибрати диск та папку куди вам необхідно поставити дану систему. Варто зазначити, що дане вікно містить вибір типу установки «Everyone» – для всіх користувачів даної OS системи або «Just me» – тільки для мене, що означає, що дана SCADA система буде доступна тільки даному у користувачеві облікового запису OS. Після вибору шляху установки і типу установки натисніть клавішу «Next» та вам стане доступним діалогове вікно представлене на рис. 3.7. У даному вікні підтверджується ваше бажання установки SCADA системи TRACE MODE 6 з вибраними параметрами, якщо

вибрані параметри вас не влаштовують, то ви можете повернутися в до вікна 3.6 натиснувши кнопку «Back», в іншому випадку натисніть клавішу «Next»

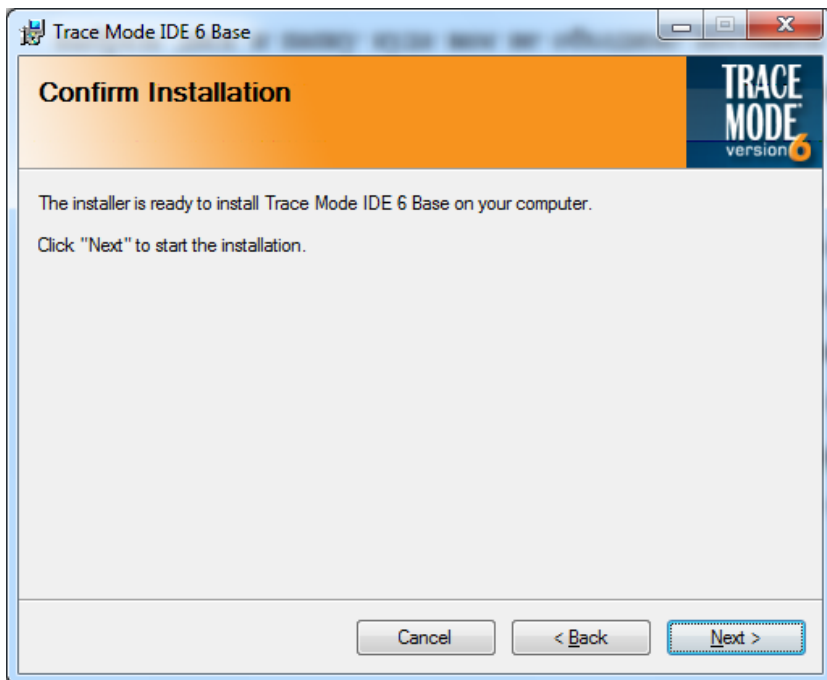


Рисунок 3.7 – Діалогове вікно підтвердження вибору установки SCADA системи TRACE MODE 6 з обраними параметрами

Внаслідок проведених дій вам стане доступним вікно інформацій про власника даного програмного продукту. Вікно представлено на рис. 3.8. В даному вікні необхідно вказати ваше ім'я і організацію, в якій буде використовуватися SCADA система TRACE MODE 6.

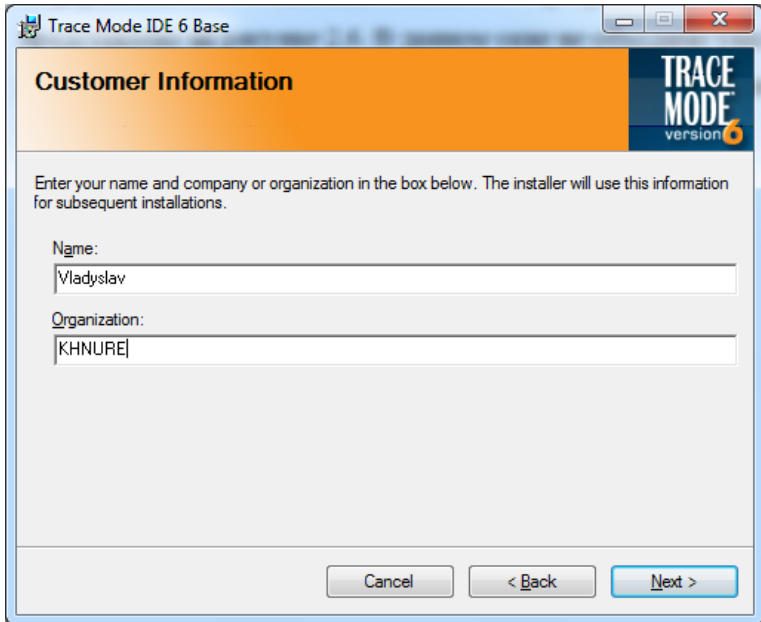


Рисунок 3.8. – Вікно введення інформації про власника

Заповнивши всі поля (Name (Ім'я) Organization (Організація), натисніть кнопку «Next». Ви перейдете у вікно установки та копіювання файлів SCADA системи TRACE MODE 6 на ваш диск. Це процедура триває 1-5 хвилин в залежності від потужності вашого комп'ютера. При успішній установці у вас з'явиться вікно підтвердження успішної установки, при натисканні клавіші «Close» діалогове вікно буде закрито: SCADA система TRACE MODE 6 успішно встановлена на ваш комп'ютер.

3.4. Основні визначення та поняття в SCADA TRACE MODE 6

ПРОЕКТ системи керування – це сукупність всіх математичних і графічних елементів системи, що функціонують на різних операторських станціях і контролерах однієї АСК ТП, об'єднаних інформаційними зв'язками й єдиною системою архівування. Проект може бути масштабним (сотні вузлів), а може містити тільки один контролер або одну операторську станцію. Під проектом в TRACE MODE 6 розуміється вся сукупність даних і алгоритмів функціонування розподіленої АСК (АСКТП і/або T-FACTORY), заданих засобами TRACE MODE.

Підсумком розробки проекту є створення файлів, що містять необхідну інформацію про алгоритми роботи АСК. Ці файли потім розміщуються на апаратних засобах (комп'ютерах і контролерах) та виконуються під управлінням виконавчих модулів TRACE MODE.

Складова частина проекту, що розміщується на окремому комп'ютері або в контролері та виконується під управлінням одного або декількох виконавчих модулів TRACE MODE, називається вузлом проекту.

Вузол – будь-який пристрій в рамках проекту, в якому запущено програмне забезпечення TRACE MODE, що реалізує серверні функції. Це може бути контролер, операторська станція або архівна станція. У проекті не може бути більше 128 вузлів. У загальному випадку розміщення вузла на тому ж апаратному засобі, на якому він повинен виконуватися під керуванням монітора, не є обов'язковим – монітори можуть завантажувати вузли з віддалених апаратних засобів.

База каналів – сукупність всіх каналів, математичних об'єктів, FBD-програм і ІL-програм, створених для кожного конкретного вузла.

Об'єкт бази каналів – сукупність будь-яких каналів, якій приписаний певний набір властивостей і атрибутів. Серед останніх можна назвати ім'я, графічний ідентифікатор, прапор підпорядкування: батько (предок, попередник), нащадок.

Оформлені групи каналів можуть бути підпорядковані один одному і створювати таким чином ієрархічні структури.

Драйвери обміну – драйвери, використовувані моніторами Trace Mode для взаємодії з пристроями, протоколи обміну з якими не вбудовані в монітори.

Канал (базове поняття системи) – це структура, що складається з набору змінних і процедур, що має налаштування на зовнішні дані, ідентифікатори і період перерахунку її змінних. Ідентифікаторами каналу є:

- ім'я, коментар і кодування. Наприклад, ім'я каналу, пов'язаного з п'ятим каналом плати аналогового введення, розташованої в першому посадковому місці контролера, буде AI_re01-0005.
- числовий ідентифікатор, який використовується всередині системи для посилань на цей канал. Серед змінних каналу виділяються чотири основних значення:
- вхідне (In),
- апаратне (A),
- реальне (R),
- вихідний (Q).

За допомогою налаштувань вхідне значення каналу зв'язується з джерелом даних, а вихідне – з приймачем.

Залежно від напрямку руху інформації, тобто від зовнішніх джерел (дані з контролерів, УСО або системні змінні) в канал або навпаки, канали підрозділяються на вхідні (тип **INPUT**) (рис. 3.9) і вихідні (тип **OUTPUT**) (рис.3.10).

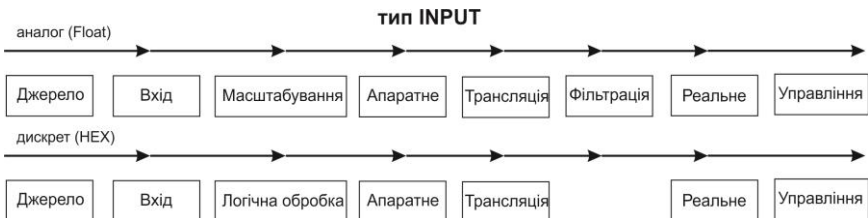


Рисунок 3.9 – Канали на вході (тип INPUT)

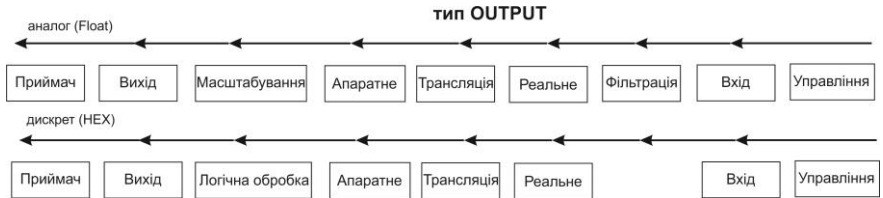


Рисунок 3.10 – Вихідні канали (типа OUTPUT)

Вхідний канал (рис. 3.9) запитує дані з зовнішнього джерела (контролер, інший MPB та ін.) або значення системних змінних (лічильник помилок, довжина архіву та ін.). Отримане значення надходить на вхід каналу і далі перераховується в апаратне та реальне значення. Апаратне значення у каналів типу INPUT формується масштабуванням (логічною обробкою для дискретних каналів) вхідних значень.

Використовувані процедури забезпечують первинну обробку даних (виправлення помилок датчиків, масштабування, корекція температури холодних спаїв термопар і т.д.). Вихідні значення в каналах типу INPUT не використовуються.

Вихідний канал (рис. 3.10) передає дані приймача. Приймач може бути зовнішнім (значення змінної в контролері, в іншому MPB та ін.) або внутрішнім – одна з системних змінних (номер звукового файлу, що програться, номер екрану, виведеного на монітор, і ін.). І зовнішні, і внутрішні приймачі даних зв'язуються з вихідними значеннями каналів. У каналів типу OUTPUT їх вхідне значення формується одним із таких способів:

- процедурою управління даного каналу;
- процедурами управління або трансляції інших каналів;
- метою програми на мові Техно ІІ;
- каналом віддаленого вузла (наприклад, через мережу);
- оператором за допомогою керуючих графічних форм.

У каналів типу OUTPUT апаратне значення виходить з реальної процедури трансляції. Апаратні значення каналів мають таку назву, оскільки з них зручно отримувати величини уніфікованих сигналів, з якими працює апаратура

введення/виведення (4-20мА, 0-10В і ін.). Реальні значення, призначені для зберігання значень контрольованих параметрів або сигналів управління в реальних одиницях (наприклад, кг/год, °С, % та ін.). Вихідне значення визначено лише для каналів типу OUTPUT. Воно перераховується з апаратного значення.

Дані з зовнішніх пристроїв записуються в канали, дані з каналів надсилаються на зовнішні пристрої. В канали оператор заносить керуючі сигнали. Значення з каналів записуються в архіви, операторські звіти і т.д. У каналах здійснюється перетворення даних. Змінюючи значення на системних каналах, можна управляти виведеною на екран інформацією, звуковими сигналами і т.д., тобто всією системою.

Процедури. Вхідне значення каналу перетвориться в апаратне, реальне і вихідне за допомогою процедур.

Процедурами каналу є:

- масштабування (множення і зміщення),
- фільтрація (придушення піків, апертура і згладжування),
- логічна обробка (попередня, інверсія, контроль сполучуваності),
- трансляція (виклик зовнішньої програми),
- управління (виклик зовнішньої програми).

Порядок проходження та зміст процедур може змінюватися в залежності від типу каналу (вхідний або вихідний, аналоговий або дискретний).

Процедура масштабування використовується тільки в каналах, які працюють з аналоговими змінними. Вона включає в себе дві операції: множення і зміщення. Послідовність цих операцій змінюється в залежності від типу каналу:

- у каналів типу INPUT вхідне значення множиться на заданий множник і до отриманого результату додається величина зсуву. Результат присвоюється апаратного значенням каналу;

- у каналів типу OUTPUT до апаратного значення додається величина зсуву, потім ця сума множиться на заданий множник, а результат присвоюється значенню каналу.

Процедура трансляції визначена для всіх каналів незалежно від їх типу та виду подання. Біля вхідних каналів процедура трансляції перетворює апаратне значення в реальне, а для вихідних – навпаки. Для цього викликається FBD-програма. Її викликає програма, що обирається при налаштуванні процедури. При налаштуванні процедури вхідні та вихідні аргументи обраної програми зв'язуються з атрибутами поточного каналу, а також будь-яких інших каналів з поточної бази. Тому процедура трансляції одного каналу може також використовуватися для формування значень інших каналів. Набір процедур в каналі залежить від формату даних. Канали, що працюють з аналоговими змінними, використовують такі процедури масштабування, трансляцію, фільтрацію й управління. У каналах, що обробляють дискретні параметри, використовується логічна обробка, трансляція й управління.

Фільтрація – процедура, яка присутня тільки в аналогових каналах. Набір виконуваних нею операцій відрізняється для вхідних і вихідних каналів. У каналів типу INPUT фільтрація виконується після процедури трансляції до формування реального значення. Фільтрація включає в себе такі операції:

- придушення випадкових сплесків в тракті вимірювання;
- придушення малих коливань значення каналу;
- експоненціальне згладжування;
- контроль шкали – відстеження виходу реального значення каналу за встановлені межі шкали.

У каналах типу OUTPUT дана процедура формує реальне значення за вхідним значенням. При цьому виконуються такі операції:

- обмеження швидкості зміни реального значення;
- придушення малих коливань значення каналу;
- експоненціальне згладжування;

- контроль шкали – обрізання величини керуючого впливу до меж шкали каналу.

Управління – процедура, яка визначена для всіх каналів. Вона реалізує функцію управління. З її допомогою можна викликати FBD-програму, в якій можна запрограмувати необхідні алгоритми управління. В якості аргументів програмі можуть передаватися значення й атрибути будь-яких каналів з поточної бази. Ці аргументи можуть бути як вхідними, так і формованими. Формально процедура управління пов'язана з каналом тільки циклом перерахунку. Вона може взагалі ніяк не брати участь у формуванні його значень, а управляти іншими каналами. Така ситуація часто спостерігається при використанні процедури «Управління» на каналах типу INPUT.

Крім основних значень, канал має додаткові змінні:

- шість меж;
- гістерезис настройки процедур обробки;
- початкові параметри;
- прапори архівування та ін. (рис. 3.11)

Змінні, налаштування й ідентифікатори каналу утворюють список його атрибутів.

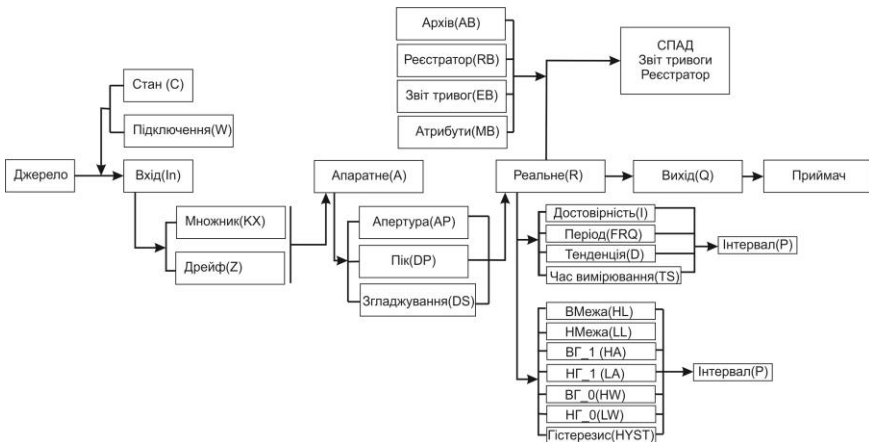


Рисунок 3.11 – Атрибути ідентифікаторів каналу

Частина з них задається в редакторі бази каналів і не може бути змінена в реальному часі. Інші можуть мати початкові значення і доступні для зміни.

Підтип каналу вказує клас джерел або приймачів даних, з якими буде зв'язуватися канал. Для каналів типу INPUT підтип характеризує одержувану ними інформацію (АНАЛОГ – значення АЦП, зчитане з плати УСО, СИСТЕМНИЙ – стан системи, ЗВ'ЯЗОК – дані з віддалених вузлів проекту та ін.). Канали OUTPUT мають той же набір підтипів, що і канали INPUT. Однак для них підтип визначає клас приймачів, а не джерел даних (АНАЛОГ – значення ЦАП, СИСТЕМНИЙ – стан системи, ЗВ'ЯЗОК – значення керованих каналів на віддалених вузлах проекту та ін.). Всього існує шістнадцять підтипів каналів. Всі вони можуть задаватися як для вхідних, так і для вихідних каналів. Підтип каналу задає клас джерел або приймачів даних. Крім того, підтип каналу визначає також кількість його додаткових налаштувань. Уточнення джерела або приймача в рамках заданого підтипом класу здійснюється за допомогою доповнення до підтипу.

Останній рівень адресації джерела або приймача даних здійснюється за допомогою налаштувань каналу.

Атрибути каналів межі шкали вказують можливий діапазон зміни контрольованого параметра. Наприклад, якщо датчик дозволяє вимірювати тиск в діапазоні від 0 до 10 кгс/см², то його свідчення, що лежать поза даного діапазону, є явно недостовірними. Якщо задати для каналу кордони шкали, то при виході за них його реального значення може автоматично формуватися ознака недостовірності даних. Ця інформація може бути доведена до оператора та зафіксована в архівах.

3.5 Механізм обміну даними та типи інтерфейсів

Послідовний інтерфейс. Обмін по всіх лініях, крім локальної мережі, реалізується через послідовний порт за протоколом M-Link. Вузли в мережі M-Link нерівноправні: один має статус Master, а решта - Slave. Такі мережі слід використовувати для зв'язку між операторськими станціями та контролерами. Монітор зі статусом Master є активним. Він посилає команди управління та запити на передачу інформації.

Монітор зі статусом Slave приймає послані йому команди та передає запитані дані. Команди управління містять вказівки на зміну значень атрибутів каналів віддаленого вузла.

Таким чином, запити, що посилаються монітором зі статусом Master, можуть бути двох типів:

- запит даних (використовується для отримання значень каналів або іншої інформації від монітора зі статусом Slave);
- запит на зміну (використовується для зміни значень атрибутів каналів на віддаленому моніторі). У запитах на зміну передаються нові значення коректованих атрибутів у віддаленій базі.

Слід зазначити, що в одній мережі M-Link не може бути двох моніторів, для яких встановлено статус Master. Щоб один монітор виступав і як Master, і як Slave, треба створити паралельні мережі, використовуючи при цьому по два послідовних порти на кожному вузлі. Тоді два монітори зможуть працювати в режимі Master.

Для обміну даними між моніторами TRACE MODE через послідовний інтерфейс використовується протокол M-Link. Він застосовується для обміну через інтерфейси RS-232, RS-485, RS-422, радіоканал, комутовані телефонні лінії та GSM мережі. Використовуючи протокол M-Link, в рамках TRACE MODE можна створювати мережеві комплекси на базі послідовного інтерфейсу RS-485. Такі комплекси можуть включати в себе до 128 вузлів (контролерів і операторських станцій). При цьому

зв'язок може здійснюватися за кількома послідовними портами. Для зв'язку двох моніторів можна використовувати інтерфейс RS-232. Щоб зв'язатися з кількома віддаленими вузлами через цей інтерфейс, потрібно мати відповідну кількість послідовних портів. Це дозволяє організувати зв'язок типу «зірка». Така конфігурація може зажадати додаткових витрат на багатоканальні плати. Однак вона дозволяє швидше передавати дані за рахунок розпаралелювання обміну з різними віддаленими вузлами. TRACE MODE підтримує обмін одночасно через 32 послідовних порти. Для зв'язку сильно рознесених в просторі моніторів можна використовувати радіоканал, виділені або комутовані телефонні лінії. У цих випадках потрібні додаткові пристрої – модеми. Вони узгоджують електричні характеристики послідовних портів і використовуваного середовища передачі.

Для обміну даними через послідовний інтерфейс між моніторами TRACE MODE застосовуються канали підтипу ЗВ'ЯЗОК.

Залежно від напрямку передачі інформації використовуються різні доповнення до підтипу цих каналів. Для запиту даних за протоколом M-Link призначені канали підтипу ЗВ'ЯЗОК з доповненням In M_Link і доповненням In M_Link (T). Для другого з них разом із значенням каналу передається час його останньої зміни. При цьому відображається час зміни значення каналу відповідає часу того МРВ, з якого зчитується канал. Воно копіюється в відповідний атрибут запитувача каналу, а також заноситься в архіви. Для передачі даних слід використовувати канали з доповненням OUT M_Link і доповненням OUT M_Link (T). В останньому випадку так само, як і при запиті, із значенням каналу передається час його формування. При зчитуванні значення каналу по M-Link (T) з МікроМРВ в МРВ відображається час зміни каналу відповідає часу МРВ. Зазначені канали мають такі настройки:

NN – номер послідовного порту;

NODE – ім'я віддаленого вузла;

CH – ім'я каналу на віддаленому вузлі;
ATR – копіюється атрибут віддаленого каналу;
OBJ – ім'я об'єкта в базі каналів віддаленого вузла.

Номер послідовного порту задається введенням з клавіатури у відповідному полі діалогу Канали об'єкта. Значення цієї настройки має бути на 1 менше номера відповідного порту (0 – COM1, 1 – COM2, ...). Інші налаштування вказуються в діалозі вибору каналу. Він виводиться на екран при натисненні лівої клавішею мишки в області завдання значення будь-якого з них.

Слід зазначити, що описані канали створюються тільки в базі монітора зі статусом Master. Канали видачі команди (OUT) через послідовний інтерфейс не працюють, якщо на той же COM-порт не налаштований хоча б 1 канал INPUT (навіть вимкнений). При відповіді на запит вузол зі статусом Slave аналізує апаратну недостовірність запитуваного каналу. Якщо значення недостовірне, то замість нього відсилається значення FFFF. Вузол зі статусом Master, отримавши таке значення, не змінює значення запитувача каналу, але виставляє йому прапор недостовірності.

Для обміну даними через протокол M_Link необхідно налаштувати відповідні параметри запуску вузла. До них відносяться статус вузла, а також фізичні параметри зв'язку. Параметри обміну через протокол M_Link налаштовуються в бланках «Основні» і «Параметри послідовних портів» діалогу «Параметри вузла». Для входу в цей діалог необхідно натиснути праву клавішу мишки на зображенні вузла, що настроюється, в редакторі бази каналів. Статус вузла при обміні через протокол M_Link задається в бланку «Основні» діалогу «Параметри вузла». Щоб вузол підтримував статус Master, необхідно встановити прапор M_Link в розділі Host Mode даного бланка, а для підтримки режиму SLAVE- той же прапор в розділі Slave Mode.

Крім статусу при обміні через M_Link необхідно налаштувати фізичні параметри порту, через який будуть

передаватися дані. Для обміну даними з контролерами через послідовні інтерфейси треба налаштувати використовувані порти. Це реалізується в бланку «Параметри послідовних портів» діалогу «Параметри вузла» редактора бази каналів. Для входу в нього треба виділити вузол, що настроюється, і натиснути праву клавішу мишки.

Цей бланк містить список послідовних портів (COM1 – порт 0, COM32 – порт 31) і сім полів налаштування параметрів виділеного в списку порту. Такими параметрами є:

- призначення порту;
- базова адреса порту;
- швидкість обміну;
- параметри зв'язку;
- таймаут на очікування відповіді;
- номер використовуваного переривання;
- режим управління передавачем.

Значення параметра «Призначення порту» формується з списку з чотирма наступними пунктами:

- зв'язок з контролерами;
- Slave M_Link;
- Modem;
- GSM_SMS.

За замовчуванням встановлюється значення «Зв'язок з контролерами». Це означає, що порт використовується для обміну з контролерами через зовнішній драйвер або через вбудовані протоколи зі статусом Master. Для обміну через протокол M_Link зі статусом Slave, в даному полі слід встановити призначення - Slave M_Link. Режим зв'язку Modem потрібно встановити для порту при його використанні для обміну через комутовані лінії, а GSM_SMS - при обміні через GSM мережу.

Два поля бланка «Параметри портів» такі як «Базова адреса порту» і «Номер використовуваного переривання» призначені для завдання базової адреси і номера переривання порту. Вони мають сенс при налаштуванні вузла, що

запускається під керуванням МікроМРВ. В інших випадках ці параметри портів налаштовуються засобами WINDOWS з Панелі управління (див. Довідкову систему TRACE MODE). У будь-якому випадку їх не можна залишати нульовими, бажано поставити їм реальні значення. Наприклад, базова адреса порту – 3f8, номер використовуваного преривання – 4.

Наступне поле «Швидкість обміну» заповнюється зі списку: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19.2k, 38.4k, 57.6k, 115.2k, 144k, 192k, 288k, 576k. Причому швидкість обміну через протокол M-LINK не повинна бути нижче 600 біт/с. Її величина при обміні через послідовний порт обмежується відстанню та наявністю перешкод в лінії. Чим нижче швидкість обміну, тим менше ймовірність збою. Наприклад, вона може бути призначена рівній 4800 біт/с.

В полі «Параметри зв'язку» задаються такі параметри обміну як кількість інформаційних біт в посилці; кількість стопових біт; наявність перевірки на парність. Значення всіх цих параметрів задається вибором зі списку. Кожен рядок цього списку містить одне з доступних сполучень цих трьох параметрів. Ці рядки мають такий вигляд:

$$k - m - x,$$

де k – кількість інформаційних біт;

m – кількість стопових біт;

x – наявність перевірки на парність (n – відсутність перевірки, e – перевірка на парність, o – перевірка на непарність).

Значення поля «Таймаут на очікування відповіді» вводиться безпосередньо з клавіатури. Воно задає час очікування відповіді від пристрою, якому був посланий запит з даного порту. Величина часу очікування задається в мілісекундах. Якщо величина таймаута не задана, то вона приймається рівною 100мс. Якщо протягом часу таймаута відповідь на запит від пристрою або МРВ не прийшов, то

каналу, що запросив ці дані, ставиться прапор апаратної недостовірності.

Крім того, щоб встановити тривалість затримки на включення передавача після завершення прийому в каналах на базі RS-485 і RS-232 використовується таймаут «RS-передача». Його значення в мілісекундах задається в бланку «Таймаут» того ж діалогу. В поле «Режим управління передавачем» вноситься «Ні», якщо не потрібно управляти передавачем. Інші пункти, крім першого, задають різні режими управління.

Можна помітити, що МікроМРВ підтримує до 4 зв'язків зі статусом Master по M-Link або по-іншому вбудованому протоколу (по 4-м COM-портів, що мають один і той же вектор переривання), а зі статусом Slave – тільки один зв'язок (на будь-якому перериванні).

В рамках завдань управління обміном по послідовних портах TRACE MODE дозволяє здійснювати такі операції:

- відключення обміну через порт;
- перемикання обміну на резервний порт;
- відключення групи каналів від обміну.

Одним з найбільш перспективних стандартів обміну даними між додатками WINDOWS при створенні систем управління є механізм OPC. OPC (OLE for Process Control) – стандартизовані інтерфейси для Microsoft технології COM, призначені для застосування в області автоматизації управління технологічними процесами. Стандарт OPC розроблений міжнародним фондом OPC Foundation, який був створений фірмами Fisher-Rosemount, Intellution, Intuitive Technology, Opto22, Rockwell і Siemens в 1995 р. У 1996 р з'явилася перша версія специфікації OPC.

OPC в даний час є стандартом, який визнаний розробниками, системними інтеграторами та користувачами АСК ТП. Сьогодні практично всі виробники програмного і апаратного забезпечення АСК ТП розробляють продукти, що відповідають цьому стандарту.

За останні кілька років OPC сервери повністю витіснили DDE (Dynamic Data Exchange) сервери та спеціалізовані драйвери для апаратних засобів автоматизації. DDE - найстаріший (час появи - 1989-1991pp.) і дуже повільний спосіб динамічного обміну даними між Windows додатками, був згодом замінений (перетворений) в OLE (Object Linking and Embedding). OLE з початку і до середини 90-х років використовувався виключно Microsoft для обміну даними між офісними додатками. Під час розробки Windows NT з'явилася технологія DCOM (Distributed Component Object Model) як продовження технології COM. DCOM була розроблена для розподілених клієнт-серверних додатків. Один клієнт міг одночасно використовувати кілька серверів, встановлених на різних комп'ютерах в мережі, і кожен сервер одночасно міг обслуговувати декілька клієнтів. В даний час OPC базується практично виключно на DCOM технології фірми Microsoft для розподілених систем. Головним поняттям DCOM є поняття інтерфейсу, за допомогою якого DCOM – об'єкти обслуговують клієнтів.

OPC-сервер NLogsc є програмною системою, що дозволяє підключити апаратуру до програмного забезпечення сторонніх виробників, якщо воно задовольняє стандарту OPC. Сервер NLogsc має такі відмінні риси:

- можливість надання різних прав доступу до тегів для різних клієнтів сервера;
- можливість додавання нових пристроїв в бібліотеку драйверів, що може розширюватися;
- містить вбудований редактор скриптів VBScript для опису призначених для користувача конверторів вхідних змінних;
- підтримує призначені для користувача DLL-бібліотеки для опису складних конверторів вхідних змінних;
- крім стандартного OPC-інтерфейсу, має додатковий спрощений COM інтерфейс Easy Access для управління пристроями;

- містить об'єкт для інтеграції сервера NLog і OPC-серверів сторонніх виробників з програмами, які не підтримують OPC, але підтримують OLE, наприклад, MS Excel, MATLAB.

Необхідний обсяг вільного місця на жорсткому диску становить п'ять мегабайт. OPC-сервер працює тільки з COM портами або їх емуляторами. MPB може виступати в якості OPC-сервера і OPC-клієнта. Як OPC-клієнт він підтримує такі режими:

- SYNC/CACHE – синхронне читання з CACHE;
- SYNC/DEVICE – синхронний обмін даними з пристроєм;
- ASYNC/DEVICE – асинхронний обмін даними з пристроєм;
- ADVISE – асинхронне читання даних при зміні їх значень.

У режимі ADVISE MPB приймає значення, що надсилаються через канал підписки. Вони зазвичай надсилаються сервером тільки при зміні значення параметра.

У режимі ASYNC MPB опитує OPC-сервер і приймає дані, що надсилаються через канал підписки в разі зміни значення параметра. При цьому підтримуються такі типи даних:

- VT_R4 (FLOAT, 4 байти) – для каналів типу Float;
- VT_I4 (INT, 4 байти) – для каналів типу Hex.

Для обміну даними через OPC між моніторами TRACE MODE використовуються канали підтипу ЗВ'ЯЗОК з доповненнями In OPC – прийом даних від MPB через OPC, Out OPC – передача даних MPB через OPC.

Під час налаштування зв'язку з OPC для кожного вузла необхідно вказати ім'я комп'ютера, на якому він буде запущений. Для цього в діалозі «Параметри вузла» на бланку «Основні» передбачено поле «Ім'я комп'ютера». Для доступу до віддаленого комп'ютера може знадобитися запуск утиліти DCOMCNFG.EXE і надання відповідних дозволів користувачам.

Канали для зв'язку з OPC-сервером створюються процедурою автопобудови. Щоб запустити її, слід, перебуваючи у вікні об'єктів вузла, що настроюється, виконати команду «Посилання на OPC-сервер» з меню «Вузол» або натиснути

клавіші «Alt»+«L». При цьому з'явиться екран «Вибір сервера OPC», на якому є три кнопки: «Додати», «Видалити», «Змінити». Натискання кнопки «Додати» виводить на екран «Вибір сервера OPC» – перелік серверів, зареєстрованих на локальній машині або на будь-якому комп'ютері, присутньому в мережі. Зазначений сервер додається в список попереднього діалогу.

При натисканні кнопки «Видалити» виділений в списку сервер видаляється з вікна. Кнопка «Змінити» використовується для заміни виділеного сервера. Вона виводить на екран той же діалог, що і кнопка «Додати». Обраний в ньому сервер замінює поточний.

Щоб створити канали TRACE MODE для обміну з виділеним в списку сервером, треба натиснути лівою клавішею на кнопці «Вибрати».

У лівому вікні з'явиться екран, слід вибрати канали OPC-сервера, які треба контролювати в MPB, і перенести їх в праве вікно натисканням лівої клавіші на кнопці «<>». Після виходу з цього діалогу в базі каналів з'явиться новий об'єкт, ім'я якого утворено з ідентифікатора OPC-сервера. У ньому створюються канали для обміну із зазначеними каналами сервера.

Для зв'язку з базами даних і бізнес-додатками в MPB вбудована підтримка інтерфейсу ODBC [1]. MPB може запитувати дані з зареєстрованих джерел даних ODBC і записувати в них значення каналів. При цьому передача значень каналів може здійснюватися як в режимі формування нових записів в базі (INSERT), так і в режимі оновлення існуючих (UPDATE).

Щоб зв'язатися з базами даних (БД) через ODBC в директорії проекту треба створити конфігураційний файл `odbc.cfg`. Цей файл має текстовий формат. У ньому описується база даних, ім'я користувача, що має доступ до неї, а також елементи запитів на мові SQL для управління обміном даними. При цьому з метою забезпечення обміну з будь-якими ODBC-

серверами фрагменти SQL-запитів слід записувати прописними буквами.

Перед тим як створити джерело даних, необхідно переконатися в наявності TRACE MODE драйвера ODBC driver, установка якого зазвичай проводиться автоматично при інсталяції системи. Якщо з якихось причин він не встановлений, необхідно виконати його установку вручну.

3.6. Метод розподілених комплексів

TRACE MODE має потужні засоби для створення розподілених АСК ТП, що включають в себе до трьох рівнів ієрархії:

- рівень контролерів – нижній рівень;
- рівень операторських станцій – верхній рівень;
- адміністративний рівень.

Розподіл на рівні іноді може бути досить умовним. У малих системах функції всіх рівнів часто реалізуються на одній операторській станції. У великих же на кожному рівні може бути виділена своя ієрархія. Проте в більшості випадків такий розподіл правомірний. Необхідно зауважити, що при розробці великих мережевих систем, що включають в себе десятки вузлів, лімітуючим елементом стають не характеристики пакета за кількістю одночасно працюючих в мережі вузлів, а пропускна спроможність ліній зв'язку. Виконавча система TRACE MODE включає в себе монітори, призначені для роботи на всіх рівнях систем управління. Розглянемо три рівні:

- рівень контролерів. На цьому рівні реалізується збір даних від датчиків і НЦУ. Для створення цього рівня передбачені монітори: МікроМРВ, МікроМРВ Модем+, МікроМРВ GSM+. Перший з них призначений для запуску в контролерах, пов'язаних з верхнім рівнем через локальну мережу або послідовний інтерфейс, другий – для зв'язку через комутовані лінії, а третій – через GSM-мережу. При використанні виділених телефонних ліній або радіоканалів слід застосовувати перший монітор. Ці монітори не мають графічного інтерфейсу. Однак за математичними функціями вони ідентичні моніторам верхнього рівня, а також мають ряд функцій, необхідних для роботи в контролерах (наприклад, підтримка сторожового таймера);
- оперативний рівень. Для верхнього рівня АСК ТП передбачені такі монітори, як МРВ, NetLink МРВ, NetLink Light. Вони дозволяють створювати робочі станції

оперативного персоналу, що управляє. MPB може обмінюватися даними з іншими моніторами TRACE MODE, а також з будь-якими контролерами через вбудовані протоколи або драйвер. Він запитує дані у нижнього рівня і передає йому команди управління. Отримані дані можуть відображатися, архівувати і передаватися іншим додаткам WINDOWS через протоколи ODBC, OPC та DDE. NetLink MPB – це мережева робоча станція. Цей монітор може обмінюватися даними з операторськими станціями (через послідовний інтерфейс або локальну мережу), а також з Мікро MPB, які працюють в PC-based контролерах. За функціями візуалізації, архівування, зв'язку з базами даних і документування NetLink MPB аналогічний MPB. На відміну від MPB, в ньому блоковані підтримка плат УСО, обмін з драйвером, обмін через вбудовані протоколи MODBUS і DCS, а також клієнтські функції OPC та DDE. NetLink Light – це мережевий графічний термінал. Він не має свого сервера математичної обробки, а зв'язується з сервером MPB або NetLink MPB, запущеним на іншому комп'ютері. NetLink Light дозволяє створювати додаткові робочі місця оператора;

- адміністративний рівень. Завданням даного рівня управління є контроль поточного стану виробничих процесів і аналіз функціонування виробництва за архівними даними.

Для вирішення завдань даного рівня передбачений монітор SUPERVISOR. Він є спеціалізованою графічною консоллю, яка може підключатися до сервера математичної обробки MPB, NetLink MPB або ГР. У перших двох випадках проглядається локальний СПАД архів, а в останньому - глобальний архів.

Крім того, SUPERVISOR можна перемкнути в режим реального часу. У цьому випадку він працює як консоль NetLink

Light, і може використовуватися для управління процесом. При роботі з архівами SUPERVISOR реалізує такі функції:

- відображення останніх змін значень каналів;
- перегляд архівів в режимі PLAYBACK;
- перегляд на заданий архівний час з покроковим переходом по часу.

Монітори реального часу TRACE MODE можуть обмінюватися даними за такими лініями: локальна мережа; послідовний інтерфейс RS-232, RS-485, RS-422; радіоканал; виділена телефонна лінія; комутовані телефонні лінії; мережі GSM. За цим носіям необхідно організувати інформаційні потоки всіх рівнів системи управління. При цьому можуть реалізуватися як вертикальні зв'язки (між рівнями), так і горизонтальні (між вузлами одного рівня).

3.7. Контрольні питання

1. Яка галузь задач розв'язується в SCADA TRACE MODE 6?
2. Наведіть приклад розв'язання задачі автоматизації в SCADA TRACE MODE 6?
3. Наведіть структуру TRACE MODE 6, основні модулі й їх призначення?
4. Які функції виконує редактор бази каналів та сервери?
5. Які функції виконує редактор подання даних та комунікатори?
6. Які функції виконує консоль і редактор шаблонів?
7. Що містить виконавча система TRACE MODE?
8. Для чого використовується сервер документування NetLink Light?
9. Поняття проект системи керування.
10. Дайте пояснення таким термінам: вузол, база каналів та об'єкт бази каналів.
11. Дайте пояснення терміну драйвери обміну.
12. Поясніть терміни канали на вході (тип INPUT) та вихідні канали (тип OUTPUT).
13. Дайте пояснення терміну процедури.
14. Які процедури ви знаєте?

15. Які атрибути ідентифікаторів каналу є в TRACE MODE?
16. Механізми обміну даними в TRACE MODE.
17. Метод розподілених комплексів.

4. ИНТЕРФЕЙС КОРИСТУВАЧА ТА ОСНОВНИЙ ФУНКЦІОНАЛ TRACE MODE 6

4.1. Інтерфейс і функціонал TRACE MODE 6

Для запуску SCADA системи TRACE MODE 6 необхідно зайти в меню «Пуск» -> Всі програми -> TRACE MODE 6 (base) -> та запустити TRACE MODE IDE 6 (base). Даний шлях може відрізнятись в залежності від версії OS, встановленого у користувача. Головна форма програми представлена на рис. 4.1.

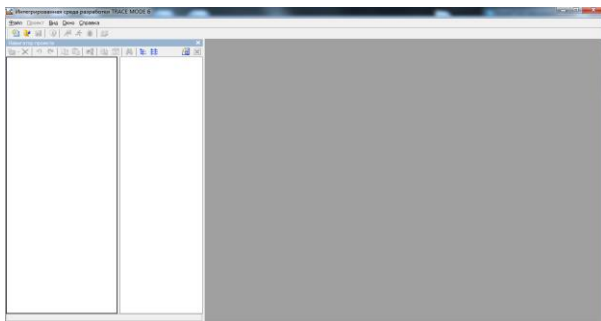


Рисунок 4.1 – Головна форма SCADA системи TRACE MODE 6

Для створення нового проекту необхідно натиснути на «Создать новый проект» в меню, що випадає, після чого відкривається діалогове вікно, представлене на рис. 4.2.

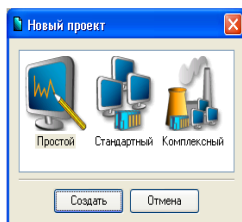


Рисунок 4.2 – Вікно створення нового проекту

4.1.1 Навігатор проекту

При створенні проекту АСК ТП необхідно використовувати навігатор проекту. На рис. 4.3 зображено вікно навігатора проекту. В лівій частині даного екрану відображається дерево проекту. В правій частині вікна навігатора відображаються під-групи, компоненти обраної групи. Виклик контекстного меню для виділеної групи або компонента дозволяє редагувати виділену одиницю, створити нову групу або компонент. Для того, щоб згорнути або розкрити шар (групу) необхідно двічі натиснути лівою клавішею миші по шару (групі).

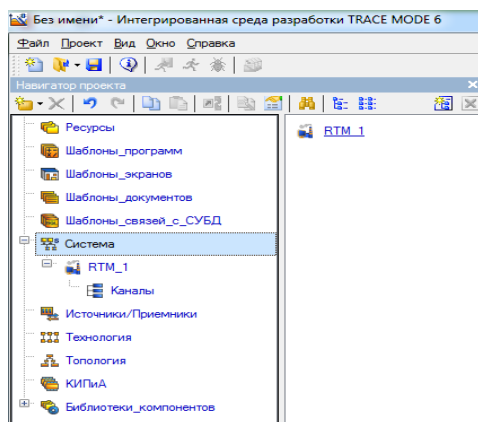


Рисунок 4.3 – Навігатор проекту

4.1.2. Редагування каналу

Крім створення каналу в багатьох випадках необхідно його редагування. Для редагування необхідно виділити канал, викликати контекстне меню. У меню слід вибрати редагування. Приклад зовнішнього вигляду вікна для каналу типу Float зображений на рис. 4.4. Зовнішній вигляд залежить від виду каналу (float, hex 16 і так далі). В даному вікні можна налаштувати системні властивості такі, як тип каналу, період перерахунку каналу, створення архівів і так далі. Канали для

Рисунок 4.4 - Вікно редагування для каналу типу Float

4.1.3. Прив'язка аргументів

При створенні проекту в TRACE MODE створюються численні канали, програми. Весь обмін даними відбувається через канали. Зв'язок між певними значеннями каналів, аргументами програми, екрана реалізується за допомогою механізму, який називається прив'язкою. При створенні програми або екрана необхідно створити аргументи. Створення аргументів виконується шляхом заповнення таблиці, зображеної на рис. 4.5.

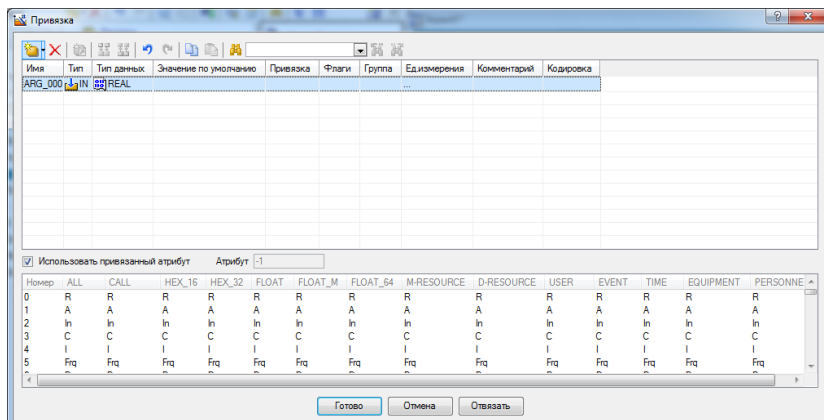


Рисунок 4.5 – Таблица створення прив'язки аргументів

Для створення аргумента необхідно зайти в «Свойства объекта». На рис. 4.6 представлено дане діалогове вікно для об'єкта типу «Текст» екранної форми. Для доступу до таблиці, представленої на рис. 4.5 необхідно натиснути на кнопку «Основная привязка».

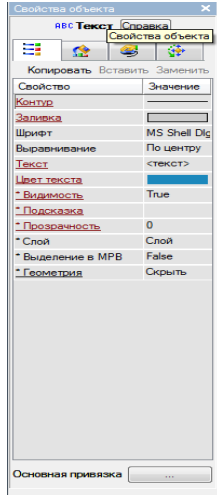



Рисунок 4.6 – Властивість об'єкта типу «Текст»

Іконка  на рис. 4.5 для створення нового аргументу. У таблиці кожна строка – аргумент. У стовпці «Ім'я» можна поставити довільне ім'я для даного аргументу. Бажано уникати прогалин в іменах аргументів. У стовпці «Тип» вказують тип аргументу: IN- для передачі в програму або виведення на екран, OUT- для передачі з програми або введення даних через інтерфейс людина-машина, IN/OUT- передача даних як в програму, так і з неї, виведення на екран і введення даних. У стовпці прив'язка проводиться прив'язка до певного значення каналу. Подвійним клацанням лівої кнопки миші в даному стовпці відкривається вікно, зображене на рис. 4.7, де вибирається необхідний канал і його значення (атрибут).

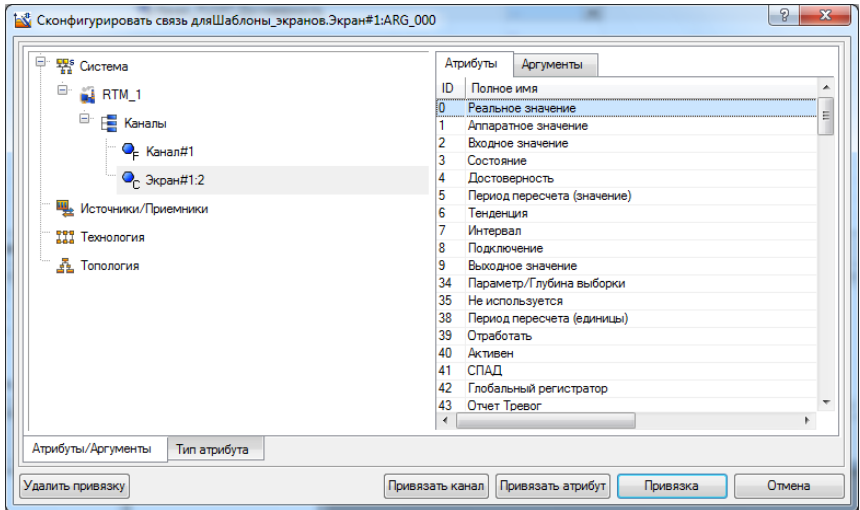



Рисунок 4.7 – Прив'язка до канала

4.2. Створення об'єктів екрана

Для виведення даних на екран, управління системою оператором необхідні об'єкти, такі як текст, стрілочний прилад, повзунок, кнопка, вимикач, тренд і так далі. Для розміщення об'єктів на екрані необхідно клацнути лівою клавішею миші по відповідній іконці інструменту (таблиця 4.1) на панелі інструментів. У ряді випадків замість нього може відобразитися інструмент для створення іншого об'єкта, тоді слід натиснути правою клавішею миші по іконці інструменту тієї ж групи, що і шуканий. Серед запропонованих інструментів слід вибрати необхідний. Після вибору інструменту слід клацанням лівої кнопки миші задати два протилежних кути об'єкта. Після створення всіх об'єктів обраним інструментом слід клацнути лівою клавішею миші по іконці  для переходу в режим редагування. У режимі редагування можна відкрити вікно властивостей об'єкта (рис. 4.6), виділивши сам об'єкт. Якщо вікно властивостей не відкрите, його можна відкрити подвійним



клацанням лівої клавіші по відповідному об'єкту. В даному вікні зліва розташовані імена полів, розділів (підкреслені). Для того, щоб розкрити або згорнути розділ досить двічі клацнути по ньому лівою клавішею миші. Заповнюючи поля і розділи можна задати зовнішній вигляд об'єкта, логіку його роботи.





Таблиця 4.1

Об'єкти TRACE MODE

Група об'єктів	Іконка об'єкта	Ім'я об'єкта
Прилади		Стрілковий прилад
		Повзунок
Тренди		Тренд
		Архівний тренд
		Тренд X-Y
		Архівна гістограма
Текст		Текст
Кнопки		Кнопка
		Група кнопок
		Картинки-кнопки
Вимикачі		Вимикач 0
		Вимикач 1
		Вимикач 2
		Вимикач 3
		Вимикач 4
		Вимикач 5
		Вимикач 6
		Вимикач 7

Найчастіше необхідно зробити прив'язку об'єкта до відповідного значення (атрибуту) каналу. Для прив'язки необхідно клацнути лівою клавішею миші в поле прив'язки або результат, джерело і так далі. Відкриється вікно, зображене на рис. 4.5. В даному вікні слід створити аргумент, зробити його прив'язку до бажаного (атрибуту) каналу. Якщо вже існує необхідний аргумент, то досить його вибрати у вікні на рис.4.5.

Ряд об'єктів не містить закладок. Інші об'єкти містять кілька закладок. Так тренди містять закладки: основні властивості  і криві  (сектори, стовпці). На закладці «Основні властивості» задається зовнішній вигляд, легенда, осі. На закладці «Криві» (сектори, стовпці) створюються і настроюються всі криві (сектори, стовпці), що виводяться об'єктом. Для створення кривої (сектора, стовпця) слід виділити рядок «Криві» (сектор, стовпець) і викликати контекстне меню. В меню буде запропоновано створити криву (сектор, стовпець).

Ряд об'єктів містять закладки динамічний контур , динамічна заливка , динамічна трансформація . Дані закладки дозволяють забезпечити динамічність зображення, яка докладніше розглядається при розгляді динамічного зображення. Розглянемо властивості об'єкта (рис. 4.6), який містить закладку дії , яка дозволяє налаштувати дії, що виконуються при натисканні (mousePressed) або відпуску (mouseReleased) лівої клавіші миші. В даній закладці можна створити і налаштувати дію аналогічно створенню кривої, викликаючи контекстне меню для mousePressed або mouseReleased.


Об'єкт вимикач дозволяє ввести і відобразити значення змінної, які беруть два значення; true (істина) і false (брехня). Для вимикача необхідно вибрати в полі «Прив'язка» «Ім'я аргументу», прив'язаного до необхідного значення параметра. Поля «Вид індикації», «Константа», «Код доступу» задають логіку роботи.

Для індикації Arg & Конст (& – побітове логічне І) характерно те, що вимикач переводиться в положення «вкл»,

коли аргумент & константа = true, в іншому випадку перемикач переводиться в положення «викл». При натисканні лівою клавшею миші по вимикачу аргументу присвоюється результат обчислення за формулою аргумент ^ Значення, де ^ – побітове виключне АБО. При індикації Arg>=Конст вимикач знаходиться в положенні «вкл», коли аргумент не менш ніж константа, якщо аргумент менше ніж константа вимикач переводиться в положення «викл». При індикації Arg == Константа вимикач знаходиться в положенні «вкл», якщо аргумент дорівнює значенню константа, в іншому випадку вимикач переводиться в положення «викл». При видах індикації Arg>=Конст, Arg == Константа і при натисканні лівою клавшею миші по вимикачу аргументу присвоюється значення, задане атрибутом.

4.2.1. Статичне зображення






Для створення зображення можна використовувати інструменти: «Лінія», «Ламані та криві», «Прямокутники», «Плоскі фігури», «Об’ємні фігури».

Для вибору інструменту «Лінія» слід клацнути лівою клавшею миші по іконці . Для створення лінії досить клацнути лівою клавшею миші мишкою там, де створювана лінія повинна починатися та закінчуватися.

Інструмент «Ламані та криві» дозволяє створювати різні криві та ламані, використовуючи об’єкти, наведені в таблиці 4.2. Інструмент на панелі інструментів відображається іконкою обраного інструменту. Якщо обраний необхідний інструмент, досить клацнути лівою клавшею миші на іконці інструменту. Під час відображення інструмент іконкою не того об’єкта, який потрібен, необхідно натиснути правою клавшею миші по іконці інструменту тієї ж групи, що і шуканий. Будуть запропоновані об’єкти, серед яких слід вибрати необхідний.

Таблиця 4.2

Ламані та криві

Іконка	Назва об'єкта
	Ламана лінія
	Багатокутник
	Ламана із заливкою
	Розімкнена крива
	Замкнена крива

Створення ламаних і кривих полягає в фіксації точок зламу або перегину клацанням лівої кнопки миші, коли курсор розташований в точці, де повинен бути злам або перегин. Положення останньої точки вигину або зламу слід фіксувати клацанням правої клавіші миші.





Інструмент «Прямокутники» дозволяє створити різні прямокутники, використовуючи стандартні об'єкти, зазначені в таблиці 4.3.

Іконка інструменту «Прямокутники» набуває вигляду іконки вибраного прямокутника. Вибір необхідного інструменту серед всіх прямокутників проводиться аналогічно вибору серед ламаних і кривих.

Для розміщення прямокутника необхідно клацанням лівої кнопки миші задати два протилежних кута створюваного прямокутника.

Таблиця 4.3






Прямокутники

Іконка	Назва об'єкта
	Контур
	Прямокутник
	Панель
	Рамка

Інструмент «Плоскі фігури» дозволяє створювати зображення клапана, стрілки, трикутника, овалу, сектора, еліпса. Стандартні об'єкти даного інструменту наведені в таблиці 4.4.

Таблиця 4.4

Плоскі фігури



Іконка	Назва об'єкта
	Плоский клапан
	Трикутник
	Стрілка
	Овал
	Еліпс сектор

Плоскі фігури створюються аналогічно прямокутнику. У плоских фігур, прямокутників, ламаних і кривих можна налаштувати статичний контур і заливку на закладці основні властивості.

Для статичного контуру можна задати його стиль, колір, товщину контуру. Для заливки можна задати тип заливки (колір або зображення). Якщо вибрано заповнення кольором, то можна задати колір, стиль заповнення. Інструмент «Об'ємні фігури» дозволяє створити різні об'єкти, зазначені в таблиці 4.5. У об'ємних фігур можна задати матеріал, ступінь прозорості (прозорість), текстуру, якість зображення, що визначає ступінь промальовування текстури, товщину стінок. Вибір краю об'ємної фігури дозволяє створювати зображення з різними краями: закруглені, скошені і так далі.

Таблиця 4.5

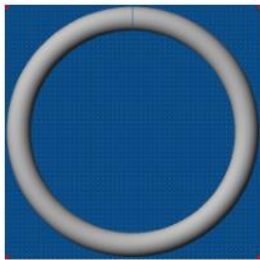
Об'ємні фігури

Іконка	Назва об'єкта
1	2
	Циліндр
	Сфера

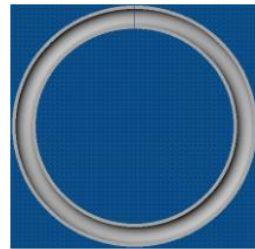
Продовження табл. 4.5

1	2
	Конус
	Тор
	Піраміда
	Ємність
	Клапан
	Насос
	Труба
	Рельєфний конус
	Криволінійний конус
	Гرادієнт

Розглянемо завдання товщини фігури. Якщо товщина стінок дорівнює 0, то на екрані зображується, як виглядає об'єкт ззовні. Коли задана товщина стінок не дорівнює 0 (товщина стінок більше 0) зображується розріз об'єкта. Приклад зображення тора при товщині стінок рівною 0 зображений на рис. 4.8 «а», при товщині стінок нерівній 0 на «б».



а)



б)

Рисунок 4.8 – Приклад зображення тора при різній товщині стінки

Розглянемо налаштування матеріалу. Для настройки матеріалу слід двічі клацнути лівою клавішею миші на підкресленому рядку «Матеріал», якщо цей розділ згорнутий. Якщо значення вибрати зі списку встановити рівним False, то матеріал задається як колір аналогічно іншим об'єктам. Якщо значення вибрати зі списку встановити рівним True, то з'являється можливість вибору матеріалу, з якого виготовлено виріб.

В останньому випадку можна вибрати такі матеріали, як хром, бронза, мідь, задати такі текстури, як шліфування або гравірування.

В якості текстури об'єкта можна вибрати зображення. Для цього необхідно імпортувати зображення:

- створити групу картини в розділі «Ресурси» навігатора проекту;
- створити бібліотеку зображень в групі «Картинки»;
- відкрити бібліотеку зображень і імпортувати необхідні зображення.


Після імпортування зображень необхідно вибрати зображення в полі «Текстура». У об'ємного прямокутника, ємності, труби можна змінювати вид верхнього або нижнього краю, змінюючи його зовнішній вигляд. У конуса можна задавати співвідношення основ у відсотках, у тора – його товщину, у клапана – його форму, а також форму і колір приводу. Насос дозволяє вибрати форму, у вигляді якої він буде зображений. Використовуючи зазначені об'єкти, розташовуючи їх певним чином, можна створити статичне зображення.

4.2.2. Динамічне зображення

Динамічне зображення відрізняється від статичного тим, що дане зображення змінюється, його стан визначається контрольованим процесом. Зображення може переміщатися, змінювати розміри, повертатися, може відбуватися переміщення


пунктирів по його контуру, об'єкт може бути заповнений до певного рівня.

Для отримання динамічного зображення завжди необхідна прив'язка до аргументу, значення якого відображається у той чи інший спосіб.

При динамічному контурі (зкладка «Динамічний контур» ) задаються два кольори: колір штрихів і проміжку між ними, довжина штриха, а також крок переміщення штрихів (рис. 4.9). Відбувається переміщення штрихів по контуру. Швидкість їх переміщення визначається прив'язаним аргументом. Якщо аргумент дорівнює 0, то переміщення відсутнє. Коли аргумент дорівнює 1 відбувається переміщення штрихів при кожному такті на один крок. Якщо аргумент дорівнює двом, наприклад, то штрихи переміщуються на один крок один раз за 2 такти.

Свойство	Значение
Привязка	ARG_000
Цвет штриха	
Цвет промежутка	
Длина штриха	5
Промежуток/штрих	1

Рисунок 4.9 – Налаштування динамічного контура

Розглянемо динамічну трансформацію (зкладка «Динамічна трансформація» ) . Можна виділити «Динамічне переміщення», «Масштабування», «Обертання», поставивши відповідний прапор. При динамічному переміщенні задається ламана лінія, уздовж якої відбувається переміщення. Для ряду точок (вузлів) задаються відповідні їм значення (рис. 4.10). Поточне положення об'єкта залежить від значення прив'язаного аргументу і значень, що відповідають вузлам, прапора «Переміщати плавно».

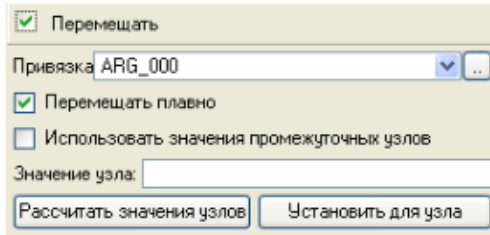


Рисунок 4.10 – Параметри динамічного переміщення

При динамізації масштабування задаються початковий і кінцевий розмір об'єкта та значення прив'язаного аргументу, що відповідають заданим значенням (рис. 4.11) і центр щодо якого буде відбуватися масштабування. Поточний розмір залежить від значення аргументу.



Рисунок 4.11 – Динамічне переміщення

При динамічному обертанні задається початковий і кінцевий кут, значення аргументу, що відповідають початковому і кінцевому кутку, центр щодо якого відбувається обертання. Кутове положення в даному випадку залежить від значення аргументу. Налаштування обертання відбувається аналогічно налаштуванню переміщення.

Динамічна заливка полягає в тому, що задається максимальне і мінімальне значення аргументу, що вводиться (рис. 4.12). Відбувається заливка об'єкта до рівня, який визначається прив'язаним аргументом. Заливка може бути одношаровою (відображається значення одного аргументу) і багатошаровою (відображення значень декількох аргументів).

Можна налаштувати зміну кольору динамічної заливки в залежності від стану технологічного процесу (попередження, аварія, поза межами). Для цього необхідно вибрати колір

заповнення і вибрати значення true в полі кольору для діапазонів.

Для настройки залежності кольору заливки об'єкта (об'єкт без динамічної заливки) від стану процесу на закладці «Основні властивості» в розділі «Заливка» слід розкрити розділ «Колір заливки», де можна вибрати вид індикації, вибрати кольори заливки (рис. 4.13), задати діапазон.

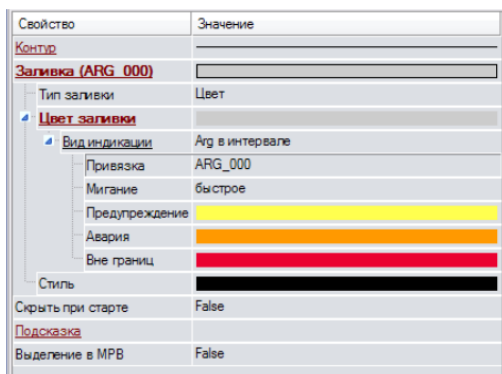


Рисунок 4.12 – Динамічна заливка

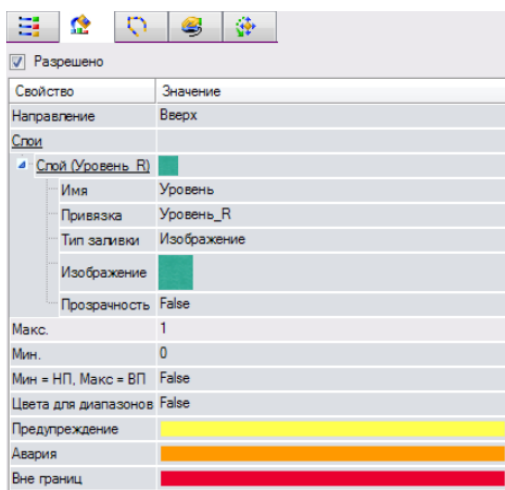


Рисунок 4.13 – Вибір кольора заливки

4.3. Тривоги та події

На багатьох виробництвах необхідно контролювати той чи інший параметр для виключення аварії, виходу з ладу обладнання. SCADA-системи дозволяють контролювати значення параметрів виробничого процесу. Одного тільки контролю параметрів недостатньо, необхідно в багатьох випадках повідомляти оператору про аварійну ситуацію, близькість значення параметра до аварійного значення, вести облік всіх аварійних ситуацій, що мали місце. Всі сучасні SCADA-системи дозволяють працювати з тривогами (аларми) і подіями.

Тривога (аларм) – повідомлення, яке попереджає оператора про виникнення ситуації близькою до критичної, яка може привести до аварії, виходу з ладу обладнання та вимагає уваги і, найчастіше, термінового втручання оператора.

Подія – повідомлення системи, яка не вимагає термінового втручання оператора. Можна виділити аналогові й дискретні тривоги. Дискретні тривоги відбуваються при відповідній зміні дискретного сигналу, тобто прийнятті значення істина або брехня. Аналогові тривоги полягають в тому, що відповідний параметр виходить за межі заданого діапазону.

Система TRACE MODE дозволяє працювати з аналоговими та дискретними тривогами. Система дозволяє встановити аналогові тривоги, задаючи різні діапазони значень контрольованого параметра. Дана система для представлення дискретних сигналів використовує спеціальні типи даних – HEX16 і HEX32. Для настройки дискретних тривог необхідно встановити відповідність між тривоною або подією та станом відповідного біта значення, що зберігається каналом HEX16 (32). Система дозволяє встановити відповідність між подією або тривоною і категорією. Категорія відображає ступінь важливості повідомлення для користувача. Як приклад можна навести такі категорії: «Без категорії», «Помилка», «Інформація», «Тривога», «Повідомлення», «Попередження».

Тривоги і події можуть відображатися за допомогою спеціального об'єкта – звіт тривог, який виводить на екран останні повідомлення про подію або тривогу. Інформація про подію або тривогу може зберігатися в текстовому файлі (звіт тривог). Відомості про тривоги та події можуть також виводитися на принтер, відправлятися на стільниковий телефон у вигляді SMS, викладатися в мережу. Також можна використовувати певний звуковий файл при генеруванні певної тривоги або події.

4.4. Тренди

Динаміку зміни технологічного параметра в часі зручно представити у вигляді залежності зміни даного параметра в часі. З цієї причини в SCADA – системах знайшли широке поширення об'єкти, які дозволяють подати зміну певного параметра в часі. Такі об'єкти називають трендами. Приклад тренда зображений на рис. 4.14. Можна виділити два типи трендів: тренди реального часу й архівні (історичні) тренди.

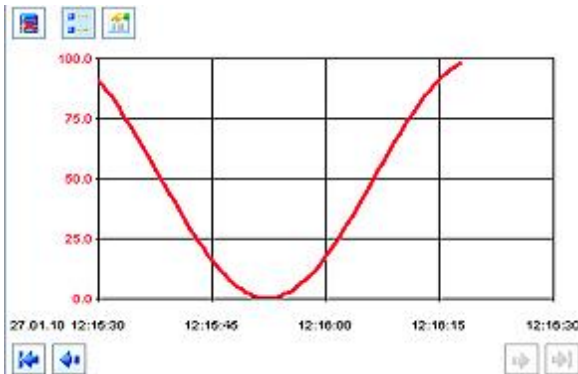


Рисунок 4.14 – Приклад тренда

Тренд реального часу відображає в реальному часі зміну параметра. Відбувається постійний зсув залежності вліво, нові

значення контрольованого параметра постійного додаються до побудованої залежності справа. Як правило, тренди містять смугу прокрутки, що дозволяє повернутися «назад» і подивитися, що відбувалося раніше на виробництві, або повернутися «в даний час».

SCADA-система дозволяє створювати архів даних. Під архівом слід розуміти бінарний файл, який містить інформацію про зміни контрольованих параметрів і часу, коли вони відбулися.

Архівний тренд дозволяє побудувати залежність обраного параметра від часу на основі даних, які збережені в архіві даних. Подібні тренди також містять смугу прокрутки, що дозволяє проаналізувати весь обсяг даних, що містяться в архіві.

SCADA-система TRACE MODE дозволяє будувати діаграми, тренди реального часу й архівні, кругову діаграму, гістограму реального часу й архівну. Кругова діаграма зручна, коли необхідно показати внесок кожної складової в суму всіх складових. Гістограма дозволяє відобразити значення декількох параметрів у вигляді стовпців, розташованих поруч один з одним. Діаграми зручні для порівняння декількох параметрів один з одним. Тренд X-Y, що дозволяє побудувати залежність параметра Y від параметра X. Середовище TRACE MODE не накладає обмеження на кількість трендів або діаграм, кількість графіків на кожному тренді, стовпців гістограми, секторів кругової діаграми.

4.5. Програмування в SCADA-системі TRACE MODE

Всі SCADA-системи містять вбудовані мови програмування. У кожній системі індивідуальна мова програмування. Все різноманіття мов програмування SCADA-систем в залежності від навичок програмування, які вимагаються від користувача для написання програми, можна розділити на дві великі групи:

- мови, орієнтовані на технологів;
- мови, орієнтовані на системних інтеграторів.

Оскільки у технологів на практиці немає досвіду програмування, то використовувати мови високого рівня вони не можуть. Для них розробляються особливі мови, які є спрощеними, що не вимагають спеціальних навичок. Дедалі більшого поширення набувають графічні мови програмування (рис. 4.15). У такій мові програмування користувач розміщує на робочому полі блоки, які виконують окремі функції. Кожен блок містить певні входи і виходи.

Користувачеві залишається з'єднати відповідні виходи з входами, задаючи напрямки передачі даних.

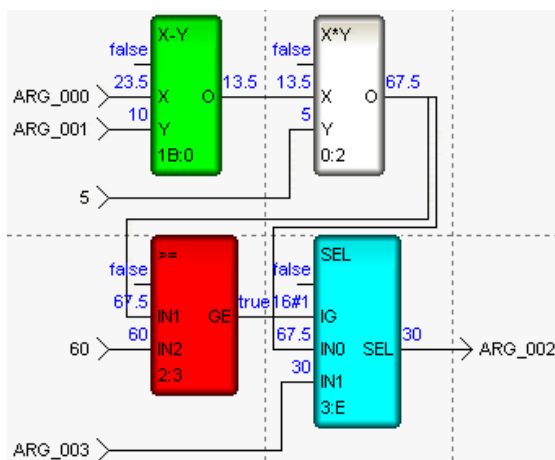


Рисунок 4.15 – Графічні мови програмування

SCADA-система TRACE MODE використовує модифіковані мови програмування, ST, IL, FBD, LD стандарту IEC61131-3. Мова TechnoST орієнтована на програмістів, її синтаксис схожий на Pascal і C. Мова дозволяє легко писати програми людям, знайомим з Pascal і з C. Це виражається в тому, що присвоєння, наприклад, можна виконувати оператором, який можна записати « \Leftarrow » і « $\Leftarrow =$ ». Дана мова дозволяє використовувати цикли, оператори безумовного переходу і з умовою, вибору. Є ряд стандартних функцій аналогічних C.

Мова Techno IL схожа на Асемблер. Базовим поняттям мови є акумулятор. Акумулятор – елемент пам'яті, що містить результат обчислень. До цієї комірки записуються результати всіх операцій, оператор може туди записати значення будь-якого операнда. Дана мова не містить циклів, операторів вибору. Мова надає користувачеві оператори для арифметичних операцій, порівняння, роботи з логічними змінними, звернення до акумулятора, безумовного переходу. Дана мова дозволяє писати програми для контролерів з низькою обчислювальною потужністю.

Мова Techno FBD – графічна мова програмування (рис. 4.15). У даній мові користувач розміщує на робочому полі блоки, які виконують різні функції, і з'єднує їх входи та виходи, задаючи напрямки передачі даних від блоку до блоку.

Мова Techno LD призначена для фахівців, які звикли складати схеми релейної логіки. Оператор розміщує на робочому столі котушки і контакти, блоки мови Techno FBD, з'єднує їх між собою, задаючи алгоритм обробки.

Мова Techno SFC дозволяє структурувати складну програму. Зовні програма нагадує блок-схему (рис. 4.16).

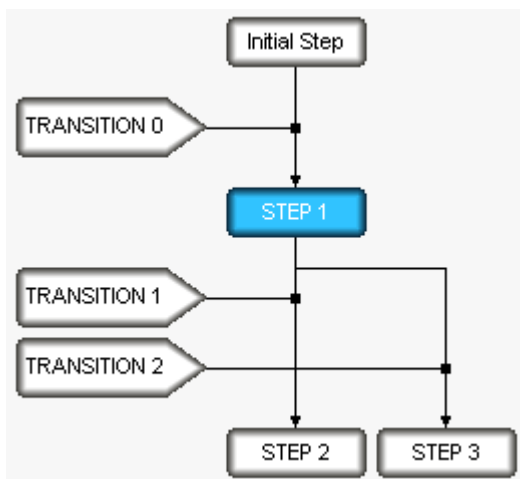


Рисунок 4.16 – Приклад програми на мові Techno SFC

Така програма містить кроки й умови. Кожен крок такої програми – підпрограма на одній з мов Trace Mode (TechnoST, Techno IL, Techno FBD, Techno LD), що реалізує ті чи інші дії. Перехід до наступного кроку в такій програмі проводиться, якщо вимога, що відповідає даному переходу, повертає значення «істина». Умова переходу до наступного кроку є програмою, написаною на одній з мов TRACE MODE (TechnoST, Techno IL, Techno FBD, Techno LD). Таким чином, дана мова дозволяє задати алгоритм виклику підпрограм (кроків), які можуть бути написані на різних мовах TRACE MODE. Розробку програм ми розглянемо в наступному розділі.

4.6. Звіти

Під звітом слід розуміти документ, який містить відомості про стан технологічних процесів, різного роду події та тривоги. Можна виділити два види звітів: звіт тривоги, який розглядається при розгляді тривоги, і звіт про стан виробництва. Розглянемо докладніше звіт про стан виробництва. Даний звіт може містити різні відомості про процеси, обладнання, тривоги, події. Це можуть бути як дані, одержувані з різних датчиків, так і результати обробки даних SCADA-системою.

Сам звіт може бути представлений по-різному: як таблиця або як html-сторінка, наприклад, SCADA-системи дозволяють зберегти створений звіт у вигляді книги MS Excel, pdf-файлу, html- сторінки.

Системи дозволяють такі дії зі створеним документом:

- відправити на друк;
- надіслати електронною поштою одержувачеві;
- опублікувати в інтернеті;
- відправляти по факсу;
- зберігати в заданій директорії.

Звіт може бути створений SCADA-системою в різних випадках:

- по команді оператора;
- періодично з заданим періодом генерації;
- за подією;
- при генеруванні тривоги;
- за результатами обчислення.

SCADA-система TRACE MODE дозволяє створювати звіт у вигляді html-сторінки. Як наслідок, звіт дозволяє представити різні дані: табличні дані, графіки, звіт тривог (вивести тривоги та події), текст, значення аргументів. Оскільки створюється html-сторінка, користувач може задати будь-який фон звіту, розмішувати малюнки і так далі.

Користувач створює шаблон документа, який задає правила оформлення, створення звіту SCADA-системою в подальшому. SCADA – система може формувати звіт як по команді оператора, так і з заданою періодичністю, при різних подіях. Користувач може написати програму, яка буде викликати генерування звіту в тих чи інших випадках. Створений звіт система TRACE MODE дозволяє вивести на друк, опублікувати на Web-сервері, зберегти в певній директорії. Звіт може бути створений на основі даних, отриманих як протягом останньої години, так протягом року. Приклад звіту системи TRACE MODE можна побачити на офіційному сайті.

Одним з поширених є звіт тривоги. Розглянемо його більш детально. Звіт тривог – текстовий документ, який містить повідомлення, що генеруються при різних ситуаціях в процесі роботи TRACE MODE.

У звіті можуть бути записані тривоги таких видів:

- системні повідомлення;
- повідомлення по каналах;
- повідомлення, що генеруються за допомогою системної змінної @Message;
- інтерактивні повідомлення оператора.

Рядок повідомлення в звіті тривоги містить поля, розділені пропуском:

Date Time Category Name Coding Text UserID T_ack N

Можна виділити такі поля:

Date – дата події;

Time – час події;

Category – категорія повідомлення. Можна виділити наступні категорії повідомлень:

1. <> – без категорії;
2. <M> – повідомлення;
3. <W> – попередження;
4. <E> – помилка;
5. <I> – інформація;
6. <A> – тривога;
7. <R> – зміна атрибутів;
8. <S> – призначене для користувача;
9. <_> – невидиме (не передається в графіку);
10. <!> – командне;
11. <?> – резерв.

Name – в разі повідомлень каналу – «Ім'я каналу»; при системному повідомленні – <ім'я файлу prj без дозволу> _<порядковий номер вузла>;

Coding – кодування каналу;

Text – текст повідомлення або призначений для користувача коментар;

UserID – ідентифікатор користувача;

T_ack – час квітування повідомлення в форматі DD_HH_MM:SS;

N – порядковий номер рядка.

Системними повідомленнями є:

1. Події, пов'язані з каналами:
 - Error – установка каналу ознаки апаратної недостовірності;
 - Login – коректна авторизація при старті;
 - Logout – коректне закінчення сеансу користувачем;

- Failed Login – некоректна авторизація при старті;
 - Failed Logout – некоректна авторизація при закінченні сеансу користувачем;
2. Події, не пов’язані з каналами:
 - Start – запуск монітора реального часу;
 - Continue – неможливість виконання файлової операції;
 - Stop – зупинка монітора реального часу.
 3. Інші події – невизначена подія.

Для створення звіту треба налаштувати аналіз меж каналу при його редагуванні. У будь-якого каналу, що містить числові дані, можна задати такі межі:

- ВП (HL) – значення верхньої межі;
- ВА (HA) – значення верхньої аварійної межі;
- ВГ (HW) – значення верхньої попереджувальної межі;
- НГ (LW) – значення нижньої попереджувальної межі;
- НА (LA) – значення нижньої аварійної межі;
- НП (LL) – значення нижньої межі;
- HYST – Гістерезис.

Якщо у вікні редагування каналу встановлено прапор «Використовувати» на панелі межі, то система TRACE MODE буде проводити аналіз меж каналів (рис.4.17).

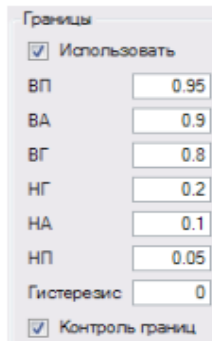


Рисунок 4.17 – Межі каналів

Прапор контроль кордонів на тій же панелі вікна редагування дозволяє установку ознаки програмної достовірності. Так якщо даний прапор встановлений і значення каналу лежить в діапазоні [НП; ВП], то атрибут даного каналу прийме значення 1, що вказує на достовірність даних, якщо значення каналу вийде за межі вказаного діапазону, атрибут «Програмна достовірність» прийме значення 0, що буде вказувати на недостовірність даних. Таким чином, діапазон (ВП; ∞) і $(-\infty; НП)$ задають недостовірні значення каналу, які можуть бути отримані при несправності датчикової апаратури або пристроїв узгодження. При недостовірності даних і встановленому прапорі «Програмна достовірність» в результаті кліпширування, канал буде повертати граничне значення НП або ВП.

При дозволеному аналізі, коли значення каналу лежить в діапазоні (ВА; ВП] або [НП; НА) генерується повідомлення про аварію. Якщо значення каналу лежить в діапазоні [ВА; ВГ) або (НГ; НА] генерується повідомлення, яке попереджає про близькість до аварійної межі. Коли значення лежить в діапазоні [НГ; ВГ] генерується повідомлення про нормальне перебігання процесу.

При невеликих коливаннях параметра поблизу однієї з меж буде створюватися велика кількість повідомлень при багаторазових переходах через межу. Коливання можуть бути на стільки малі, що переходом через межі можна знехтувати і повідомлення, що генеруються, будуть тільки заважати, тоді слід використовувати гістерезис, який дозволяє виключити подібне непотрібне генерування повідомлень. Перехід в сторону розвитку аварійної ситуації фіксується при заданих межах.

При зворотній зміні значення каналу межа коригується на величину встановленого гістерезису у відповідному напрямку.

Генерація повідомлень відбуватиметься з використанням скоригованих меж.

Словник повідомлень (рис. 4.18) дозволяє встановити відповідність між діапазонами, що задаються при налаштуванні

контролю кордонів, і повідомленнями, які генеруються при приналежності тому чи іншому діапазону, напрямком передачі даних про тривогу. У словнику повідомлень використовуються напрямки. Під напрямком слід розуміти параметр, що задають напрямком передачі повідомлення.

Направление	Категория	Текст
AR+G	<M> Сообщение	Все в порядке
AR+G	<W> Предупреждение	Выход за верхнюю предупредительную границу
AR+G	<W> Предупреждение	Выход за нижнюю предупредительную границу
AR+G	<A> Тревога	Выход за верхнюю аварийную границу
AR+G	<A> Тревога	Выход за нижнюю аварийную границу
AR+G	<E> Ошибка	Выход за верхний предел
AR+G	<E> Ошибка	Выход за нижний предел

Рисунок 4.18 – Словник повідомлень

Можна виділити такі напрямки передачі повідомлень:

1. AR + G + Prn;
2. AR + Prn;
3. AR;
4. AR + G + GSM;
5. AR + GSM;
6. AR + GSM + PRN;
7. AR + G + GSM + PRN;
8. G;
9. AR + Net;
10. AR + G + Net;
11. AR + GSM + Net;
12. Net;
13. Net + GSM;

14. AR + Play;
15. AR + G + Play;
16. AR + Net + Play;
17. Play;
18. AR + G + PlayStop;
19. AR + G + PlayLoopStop.

При завданні напрямку можуть використовуватися такі символи напрямків передачі повідомлень:

1. AR – в файл тривоги;
2. G – у виконавчі модулі, які здатні відображати звіт тривоги;
3. PRN – на принтер;
4. GSM – на стільникові телефони у вигляді SMS;
5. Net – в мережу;
6. Play, PlayStop, PlayLoopStop – відтворення файлу з ім'ям: {текст повідомлення}.wav.

Ступінь важливості повідомлення задається категоріями, які вказані вище. Середовище передбачає заповнення таблиці словника повідомлень за допомогою вікна, зображеного на рис. 4.19. Для заповнення даного вікна необхідно двічі клацнути лівою клавішею миші на рядку, який необхідно заповнити або скоригувати.

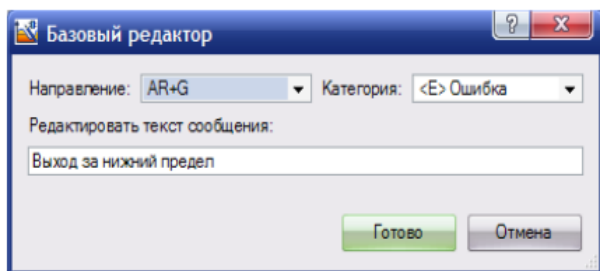


Рисунок 4.19 – Заповнення словника повідомлень

Для створення звіту тривоги необхідно зробити редагування вузла на закладці «Звіт тривоги/Дамп/Параметри»

(рис. 4.20). В полі «Максимум записів» вказується гранична кількість записів у файлі звіту тривог.

При переповненні файлу звіту тривог відбувається запис нових тривог з початку файлу звіту починаючи з другого рядка. Поле «Формат дати» задає формат запису дати і часу генерування тривоги в файлі звіту тривог. Поле «Стан» слід вибрати true.

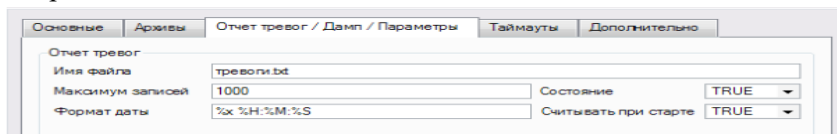


Рисунок 4.20 – Редагування вузла

При заданні формату часу та дати можна використовувати формати, наведені в таблиці 4.6.

Таблиця 4.6

Формати виведення дати та часу

Позначення	Формат
%a	скорочена назва дня тижня
%A	повна назва дня тижня
%b	скорочена назва місяця
%B	повна назва місяця
%c	дата та час відповідно до регіональних налаштувань ОС
%d	день місяця як ціле число
%H	години в форматі (00 – 23)
%I	години в форматі (01 – 12)
%j	день року як ціле число (001 до 366)
%m	місяць як ціле число (01 – 12)
%M	хвилини як ціле число (00 – 59)
%p	індикатор АМ/РМ для годинника в форматі 01 - 12
%S	секунди как ціле число
%U	тиждень як ціле число (00 – 53); перший день тижня – неділя
%w	день тижня як ціле число (0 – 6, 0 – неділя)
%W	тиждень як ціле число (00 - 53); перший день тижня - понеділок

%x	дата відповідно до регіональних налаштувань ОС
%у	рік без сторіччя як ціле число (00 - 99)
%Y	рік з сторіччям як ціле число (0000 - 9999)

Для встановлення зв'язку між створеним словником повідомлень і каналом слід зробити редагування каналу. На закладці «Архівація» слід поставити прапорець звіт тривоги, в полі «Індекс аварійного словника» вибрати потрібний словник повідомлень

4.7. Контрольні питання

1. Для чого потрібен навігатор проекту?
2. Як створити канал?
3. Як редагувати канал?
4. Наведіть типи каналів.
5. Які діапазони ви знаєте?
6. Як створюється прив'язка аргументів на каналі?
7. Основні вимоги до синтаксису в імені каналу?
8. Наведіть приклад створення об'єктів екрану?
9. Які об'єкти екрану у TRACE MODE ви знаєте?
10. Що таке тренд?
11. Наведіть приклад інструментів для створення статичного зображення.
12. Наведіть приклад інструментів «Об'ємні фігури» і для чого вони потрібні.
13. Наведіть приклад інструментів «Динамічне зображення» та поясніть їх призначення?
14. Які параметри використовують для динамічного переміщення?
15. Для чого використовують тривоги та події у TRACE MODE?
16. Які категорії повідомлення для користувача ви знаєте?
17. Які об'єкти дозволяють подати зміну певного параметра в часі?

18. Наведіть приклад тренда.
19. Що дозволяє побудувати залежність обраного параметра від часу на основі даних, які збережені в архіві даних?
20. Графічні мови програмування.
21. Які мови програмування використовує SCADA-система TRACE MODE?
22. Для чого у TRACE MODE реалізовані звіти?

5. ПРОГРАМУВАННЯ В TRACE MODE

5.1. Створення програм в TRACE MODE

SCADA – системи містять спеціальні мови програмування, що дозволяють розробнику описувати алгоритми обробки вимірювальної інформації, генерування сигналів, управління процесом. Система TRACE MODE підтримує модифіковані мови стандарту IEC6113-3:

- Techno ST (Structured Text);
- Techno SFC (Sequential Function Chart);
- Techno FBD (Function Block Diagram);
- Techno LD (Ladder Diagram);
- Techno IL (Instruction List).

Для забезпечення обміну даними між програмою і каналами вузла використовуються аргументи програми. Тип кожного аргументу визначає напрямок передачі даних. Так якщо аргумент служить для передачі значення в програму, слід вибрати тип IN, якщо для передачі з програми - OUT. У ряді випадків потрібен аргумент, який буде передавати дані як в програму, так і з неї, тоді слід вибрати тип IN/OUT. Для обміну даними програмою необхідно провести прив'язку кожного аргументу програми до того чи іншого каналу (аргументу каналу).

Крім аргументів можна створити локальні і глобальні змінні. Вікно редагування програми (рис. 5.1) можна відкрити подвійним клацанням лівою клавiшею миші по об'єкту «Програма» в навігаторі проекту або викликавши контекстне меню для нього і вибравши «Редагувати шаблон».

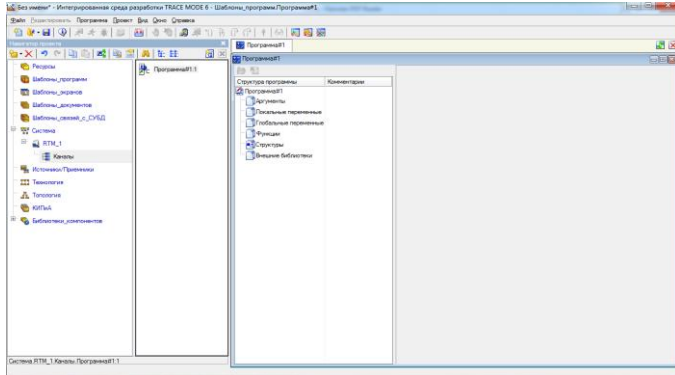


Рисунок 5.1 – Вікно редагування програми

Для створення аргументів необхідно виділити рядок «Аргументи» в структурі програми. Відкриється вікно, зображене на рис. 4.5. Створення аргументів описано вище (див. 4.1.3 Прив'язка аргументів). Для створення локальної або глобальної змінної необхідно виділити відповідний рядок в структурі програми. Відкриється вікно аналогічне створенню аргументів (аналогічно рис. 4.5). Змінні створюються аналогічно створенню аргументів. Відмінність створення змінних полягає в неможливості прив'язки змінної до значення (атрибуту) каналу.

Для написання програми необхідно вибрати рядок «Програма» в структурі програми. Буде запропонований вибір мови (рис. 5.2). Слід вибрати необхідну мову та приступити до написання програми.

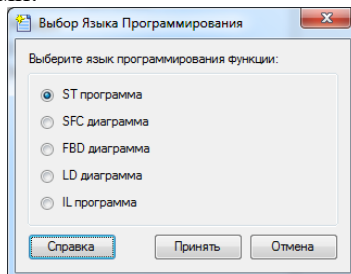


Рисунок 5.2 – Вікно вибору мови програмування

Розглянемо детальніше мови програмування Techno ST і Techno FBD, TechnoIL, TechnoSFC.

5.2. Синтаксис мови Techno ST

В алфавіт мови Techno ST входять:

- великі та малі літери латинського алфавіту (a, b, c, ..., z);
- цифри (1,2,3, ..., 0);
- спеціальні знаки (+ - * / <=>! : & | ^ ~% () [],; #).

Ідентифікатори можуть складатися з великих і малих літер латинського алфавіту, знака підкреслення «_», цифр. Ідентифікатори не чутливі до регістру.

Для даної мови характерні ключові слова: and, array, bool, break, by, byte, case, constant, continue, date, date_and_time, dint, do, dt, dword, else, elsif, end_case, end_for, end_function, end_function_block, end_if, end_program, end_repeat, end_struct, end_type, end_var, end_while, exit, false, for, function, function_block, goto, handle, if, int, lreal, mod, not, of, or, program, real, repeat, return, rol, ror, shl, shr, sint, string, struct, time, time_of_day, to, tod, true, type, udint, uint, until, usint, var, var_arg, var_global, var_inout, var_input, var_output, while, word, xor.

В якості роздільників використовуються лексеми: +, -, *, **, /, <, <=, <>, <<, >, >=, >>, !, !=, =, =, ::, :=, &, |, ^, ~, %, (), [], ,, ... , , , , ;.

Рядковий коментар починається з «//» і триває до кінця рядка. Блоковий коментар починається з «/*» і триває до «*/».

Вираз – послідовність операндів, роздільників і символічних операторів, що задають обчислення без присвоєння результату.

Речення – послідовність лексем, що визначає виконання логічно закінченої проміжної дії. Реченнями є привласнення змінній результату будь-яких обчислень, виклик функції, оператори. В кінці речення обов'язково повинен стояти символ «;». Винятком є визначення змінної.

Програму на мові Techno ST можна представити у вигляді структури:

```
PROGRAM  
{опис аргументу}  
{список речень}  
END_PROGRAM
```

Змінні можна задати, заповнюючи таблицю аргументів, локальних і глобальних змінних. Змінні визначаються в розділі опису аргументів автоматично відповідно до заповнених таблиць аргументів і змінних.

Мова TechnoST дозволяє створювати константи. Розглянемо їх подання. Можна створити числові і рядкові константи.

Цілочисельна десяткова константа починається з цифри, відмінної від нуля, після якої розташовуються будь-які цифри. Можна навести такі приклади цілочисельних десяткових констант: 123, 350, 498.

Двійкова цілочисельна константа починається з префікса «2#», після якого наводиться двійкове подання цілого числа. Приклади: 2#1011, 2#0111, 2#1001.

Вісімкові цілочисельні константи починаються з префікса «8#», після якого записується вісімкове подання числа. Приклади: 8#145, 8#0277, 8#756.

Шістнадцятиричні цілочисельні константи починаються з префікса «16#», після якого наводяться шістнадцятиричні подання чисел.

При записі шістнадцятиричного подання числа можна використовувати як рядкові символи а ... f, так і прописні А ... F. Приклади: 16#149, 16#A145E, 16#a145e.

Матеріальна константа складається з цілої і дробової частини. Допустима наявність тільки цілої або дробової частини. Приклади: .123, 0.456, 489. Можливо подання у форматі з плаваючою точкою (використовується префікс е або Е із зазначенням порядку). Приклади: 1.23E-6, 6.7504E4, 6.798e-5.

Окремим випадком числової константи є часовий інтервал, дата, час дня. Часовий інтервал записується у вигляді:

`t#<дні>d<годинник>h<хвилини>m<секунди>s<мілісекунди>ms`

Можливий також запис у вигляді:

`time#<дні>d<годинник>h<хвилини>m<секунди>s<мілісекунди>ms`

Будь-яка складова в наведених поданнях тимчасового інтервалу може бути опущена. Часовий інтервал рівний 2 годинам, 31 хвилині, 25 секундам і 10 мілісекундам може бути записаний як `t#2h31m25s10ms` або у вигляді `time#2h31m25s10ms`.

Дата записується у вигляді `d#<рік>-<місяць>-<день>`, можливий також запис у вигляді `date#<рік>-<місяць>-<день>`. 25 вересня 2016 року може бути записано у вигляді `d#2001-9-2016` або `date#2016-9-2001`.

Час дня можна записати в форматі

`tod#<година>:<хвилина>:<секунда>` або

`time_of_day#<година>:<хвилина>:<секунда>`.

Час 19 годин 15 хвилин 42 секунди може бути записано як `tod#19:15:42`, так і `time_of_day#19:15:42`.

Константа «дата і час» може бути записана як

`dt#<рік>-<місяць>-<день>-<година>:<хвилина>:<секунда>`, так і `date_and_time#<рік>-<місяць>-<день>-<година>:<хвилина>:<секунда>`.

Наприклад, 12 лютого 2017 року 13 годин 47 хвилин і 13 секунд можна записати у вигляді `dt#2017-2-12-13:47:13` або `date_and_time#2017-2-12-13:47:13`.

Крім числових констант часто використовуються рядкові константи, які представляють собою набір символів в одинарних або подвійних лапках. Приклад: «Перший рядок символів», `Другий рядок символів`. У рядках не можуть бути присутніми керуючі символи, лапки і символ \$. Для розміщення в рядку довільного символу, включаючи керуючі, використовується механізм ескейп-послідовностей.

Даний механізм дозволяє розмістити в рядку такі послідовності:

\$R – повернення каретки;

\$N – новий рядок;

\$T – табуляція;
 \$UXXXX – UNICODE символ, де XXXX – шістнадцятковий символ;

\$X – символ x («x» – будь-який символ).

Під символними операторами розуміють знаки операцій, виконуваних над операндами. Як операнди можуть виступати:

- імена констант;
- імена змінних;
- імена масивів із зазначенням індексу окремого елемента;
- виклик функцій для користувача;
- виклик бібліотечних функцій;
- вирази, укладені в дужки;
- імена елементів структур.

Арифметичні оператори наведені в таблиці 5.1.

Таблиця 5.1

Арифметичні оператори

Оператор	Дія
Унарний «-»	Зміна знака
Унарний «+»	Пуста операція
«+»	Складання чисел і конкатенція рядків
«-»	Віднімання
«*»	Множення
«/»	Ділення
«%»	Отримання залишку від ділення
«**»	Введення в ступінь

Побітові операції наведено в таблиці 5.2.

Таблиця 5.2

Побітові операції

Оператор	Дія
1	2
«&»	Побітове «І»
« »	Побітове «АБО»
«^» або хог	Побітове «виключне АБО»

Унарна «-»	Порозрядна інверсія
------------	---------------------

Продовження табл. 5.2

1	2
«<<» або shl	Зсув вліво на вказану кількість розрядів
«>>» або shr	Зсув вправо на вказану кількість розрядів
rol	Циклічний зсув вліво на вказану кількість розрядів
ror	Циклічний зсув вправо на вказану кількість розрядів

Оператори порівняння наведені в таблиці 5.3.

Таблиця 5.3

Оператори порівняння

Оператор	Дія
«= =»	Рівність
«!=» або «<>»	Нерівність
«<>»	Менше
«>>»	Більше
«<=»	Менше або дорівнює
«>=»	Більше або дорівнює

Логічні оператори наведені в таблиці 5.4.

Таблиця 5.4

Логічні операції

Оператор	Дія
«&&» або and	Логічне «І»
« » або or	Логічне «АБО»
«!» або not	Логічне заперечення

Оператор присвоювання дозволяє зробити присвоювання значення змінної. Є два синтаксису і оператора присвоювання:

$$\{операнд\} = \{вираз\} \text{ і } \{операнд\} := \{вираз\}$$

Як операнд може використовуватися ім'я змінної, масиву, уточнене ім'я змінної об'єкта.

У таблиці 5.5 наводиться пріоритет символічних операторів. Першими будуть виконуватися вирази, укладені в дужках, а потім в порядку зростання номера рядка.

Таблиця 5.5

Пріоритет символічних операторів

Пріоритет	Оператор
1	**
2	!,not,-,унарний -,унарний +
3	<<, >>, shl, shr, rol, ror
4	&, ,^,xor
5	*,/,%, mod
6	+,-
7	=,!=,<>,<,<=,>,>=
8	&&, and
9	, or
10	=, :=

У ST – програмі можна використовувати стандартні функції мови C: sin, cos, tan, asin, exp, log. Для роботи з портом і каналом є спеціальні функції в середовищі TRACE MODE, про які можна знайти інформацію в довідці системи.

Для розгалуження алгоритму використовується оператор if. Даний оператор завжди починається з ключового слова if і закінчується ключовим словом end_if. Існує три варіанти завдання даного оператора. Перший варіант:

```
if {вираз} then
{послідовність речень};
end_if;
```

В даному варіанті послідовність речень виконується тільки в тому випадку, якщо вираз правильний.

Другий варіант завдання оператора if має вигляд:

```
if {вираз} then
{послідовність речень1};
else {послідовність речень2};
end_if;
```

У другому варіанті завдання оператора if перевіряється вираз. Якщо воно правильне, то виконується послідовність речень 1, в іншому випадку-послідовність речень 2.

Третій варіант оператора if має вигляд:

```
if {вираз1} then
  {послідовність речень1};
elseif {вираз2} then
  {послідовність речень2};
...
elseif {виразN} then
  {послідовність речень N};
else
  {послідовність речень N+1};
end_if;
```

В останньому варіанті оператора if виконується і-а послідовність речень в тому випадку, якщо і-ий вираз правильний. Якщо всі вирази помилкові, то виконується послідовність речень, яка йде після ключового слова else.

Мова Techno ST містить оператор вибору case. Оператор починається з ключового слова case і закінчується ключовим словом end_case. Перший варіант оператора case можна представити таким чином:

```
case {вираз} of
  {список значень}: {послідовність речень};
...
{список значень}: {послідовність речень};
end_case;
```

В даному випадку обчислюється вираз, проводиться пошук результату обчислення в списках значень. Виконується та послідовність речень, в списках значень якої знайдено результат обчислень.

Другий варіант оператора case можна представити таким чином:

```
case {вираз} of
  {список значень}: {послідовність речень};
```

```
{список значень}: {послідовність речень};  
else {послідовність речень};  
end_case;
```

Відмінність другої форми запису оператора case від першої полягає в тому, що якщо результат обчислення виразу не знайдено в жодному зі списків значень, то виконується послідовність пропозицій після ключового слова else. У першому випадку при відсутності результату обчислень в списках значень, наведені в операторі послідовності речень не виконуються.

В обох поданнях оператора case список значень – набір цілих чисел або діапазонів цілих чисел, розділених комою. Діапазон вказується у вигляді:

```
{нижня межа діапазона} .. {верхня межа діапазона}
```

Мова Turbo ST дозволяє створювати цикли з використанням операторів while, repeat, for. Синтаксис оператора while має вигляд:

```
while {вираз} do  
{послідовність речень};  
end_while;
```

В даному операторі послідовність виразів виконується, доки вираз правильний. Щоразу перед виконанням послідовності речень проводиться перевірка виразу.

Оператор repeat можна подати у вигляді:

```
repeat {послідовність речень};  
until {вираз};  
end_repeat.
```

В операторі repeat послідовність виконується, перевіряється вираз, якщо воно правильне, то послідовність виразів повторюється знову, в іншому випадку відбувається вихід з циклу.

Цикл for можна представити у вигляді:

```
for {ім'я змінної} := {вираз1} to {вираз2} by  
{вираз3} do {послідовність речень};  
end_for;
```

Даний оператор спершу привласнює змінній циклу з вказаним ім'ям результат обчислення виразу 1, виконує послідовність речень, якщо обчислена змінна циклу не перевищить значення виразу 2. Потім до змінної циклу додається вираз 3, виконується послідовність речень, якщо обчислене значення не перевищує вираз 2. Виконання послідовності пропозицій і збільшення значення змінної на вираз 3 повторюється до тих пір, поки значення змінної циклу не перевищує вираз 2. Для цикла for характерно те, що він не дозволяє створювати цикл з негативним кроком.

Оператори break і exit дозволяють вийти з поточного циклу. Оператор continue служить для переходу в кінець циклу. При його виклику все наступне за ним до кінця циклу речення не виконуються.

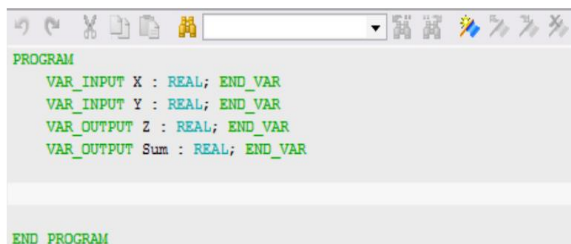
Розглянемо створення програми на мові Techno ST. Припустимо, створені аргументи, глобальна змінна дорівнюють зображеним на рис. 5.3, 5.4. Після вибору мови відкриється вікно, зображене на рис. 5.5. Всі створені аргументи, і локальні змінні будуть описані на початку програми. Глобальні змінні на відміну від локальних в самій програмі не описуються, але без проблем можуть використовуватися як операнди.

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
X	IN	REAL		X: Реальное значение (Система.RTM_1.Каналы)
Y	IN	REAL		Y: Реальное значение (Система.RTM_1.Каналы)
Z	OUT	REAL		Z: Входное значение (Система.RTM_1.Каналы)
Sum	OUT	REAL		Sum: Входное значение (Система.RTM_1.Каналы)

Рисунок 5.3 – Аргументы программы

Имя	Тип данных	[]	Начальное значение	Комментарий
S	REAL		0	

Рисунок 5.4 – Глобальні змінні програми





```
PROGRAM
VAR_INPUT X : REAL; END_VAR
VAR_INPUT Y : REAL; END_VAR
VAR_OUTPUT Z : REAL; END_VAR
VAR_OUTPUT Sum : REAL; END_VAR

END PROGRAM
```

Рисунок 5.5 – Діалогове вікно для створення програми

Зробимо розподіл аргументу X на аргумент Y, зробивши перевірку на рівність Y нулю. Результат присвоїмо аргументу Z. Аргументу Sum присвоюється сума всіх результатів ділення. Програма буде така:

```
PROGRAM
VAR_INPUT X : REAL; END_VAR
VAR_INPUT Y : REAL; END_VAR
VAR_OUTPUT Z : REAL; END_VAR
VAR_INOUT Sum : REAL; END_VAR
if Y == 0 then
Z = X / 1e-9;
else
Z = X / Y;
end_if;
Sum = S + Z;
S = Sum;
END_PROGRAM
```

Після того, як написали текст програми необхідно перевірити її. Для цього натиснемо на іконку «компіляція» . У вікні «повідомлення» буде виведений результат компіляції. Якщо вікно «повідомлення» закрито, то слід клацнути по іконці  або вибрати в меню «вид» вікно повідомлення. Якщо, наприклад, не вистачає коми в програмі, компілятор повідомить про помилку (рис. 5.6).

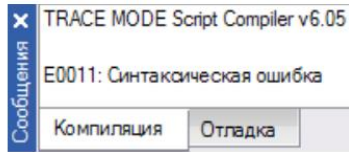


Рисунок 5.6 – Вікно повідомлення про помилку при компіляції

При відсутності помилок буде написано: «Программа#1.tms compiled successfully».

5.3. Мова Techno FBD

Розглянемо FBD програму. Дана програма є сукупністю функціональних блоків, які з'єднані між собою.

Функціональний блок – зображення виклику функції Techno ST. Як приклад розглянемо функціональний блок, що виробляє складання. Зображення його наведено на рис. 5.7.

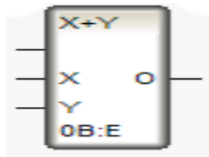


Рисунок 5.7 – Приклад функціонального блоку

У верхній частині блоку вказується позначення блоку. Внизу виводиться його номер. Номери блоком приписуються автоматично при їх розміщенні в робочому полі редактора програми. Після двокрапки вказується номер функціонального блоку, який буде виконуватися наступним. Горизонтальні лінії, розташовані зліва, виступають в якості входів, на які подається та чи інша описана локальна або глобальна змінна, аргумент програми, виходи з інших функціональних блоків. На вхід можна подати аргументи, тип яких In або In/Out. У кожного

входу вказуються його назва. У зазначеному прикладі назви: X, Y. Безіменний вхід, розташований зверху, керує виконанням блоку: блок виконує дію (в даному випадку складання) в тому випадку, якщо подається 0 або вхід не підключений, в іншому випадку функціональний блок не виконується.

Горизонтальна лінія праворуч позначає вихід, який містить результат виконання функціонального блоку. Вихід можна з'єднати з входом іншого функціонального блоку. Вихід функціонального блоку можна прив'язати до описаної глобальної або локальної змінної, аргументу, тип якого Out або In/Out. Для розміщення функціонального блоку слід відкрити вікно FBD блоки, для чого слід клацнути лівою клавшею миші по іконці або вибрати палітру FBD блоків в меню «Вид». З'явиться вікно FBD блоки (рис. 5.8).

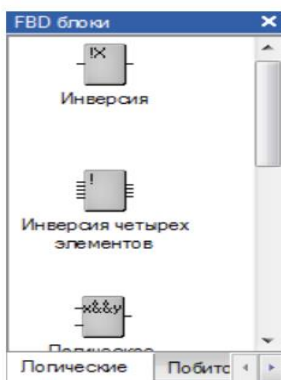


Рисунок 5.8 – Вікно FBD блоків

В даному вікні вибирається потрібна закладка («Логічні», «Побітові», «Арифметичні») відповідно до тематики потрібного блоку. У вікні відображається ряд блоків, що відносяться до даної тематики. Серед цих блоків відшукується потрібний блок і перетягується на поле редактора програми з використанням механізму drag-and-drop. Тобто, курсор наводиться на необхідний функціональний блок. Натискається ліва клавіша

миші. Блок «перетягується» на робоче поле в потрібне місце при натиснутій лівій клавіші миші. Коли курсор переведений в положення, де повинен розташовуватися функціональний блок ліва клавіша відпускається.

Для з'єднання виходу одного функціонального блоку з входом іншого слід навести курсор на вихід функціонального блоку і натиснути ліву кнопку миші. Блок стане синім, а ім'я виділеного виходу буде зеленим. Не відпускаючи клавіші миші, наведемо курсор на потрібний вхід функціонального блоку і відпустимо кнопку миші. Відповідний вхід і вихід будуть з'єднані лінією, якщо все було правильно зроблено.

Аналогічно можна натиснути ліву кнопку миші, навівши на потрібний вхід функціонального блоку, і відпустити, навівши курсор на необхідний вихід функціонального блоку.

Для прив'язки входу або виходу функціонального блоку слід виділити вхід (вихід) клацанням лівої кнопки миші. З'явиться контекстне меню, в якому слід вибрати «Прив'язати». З'явиться вікно зі списком аргументів і змінних, до яких можна прив'язати вхід (вихід) функціонального блоку. У цьому списку слід вибрати змінну або аргумент (рис. 5.9).



Рисунок 5.9 – Приклад прив'язки входу/виходу функціонального блоку

Створення константи розглянемо на прикладі створення константи рівної $1e-9$. Для цього слід виділити вхід, на який необхідно подати константу, викликаємо контекстне меню і вибираємо «Прив'язати». Буде виведено вікно, як зазначено на рис. 5.9. У вікні замість вибору аргументу або змінної вводиться значення константи. В даному випадку вводиться « $1e-9$ ».

Як приклад наведена реалізація прикладу програми, написаної на мові Techno ST, використовуючи мову Techno FBD (рис.5. 10).

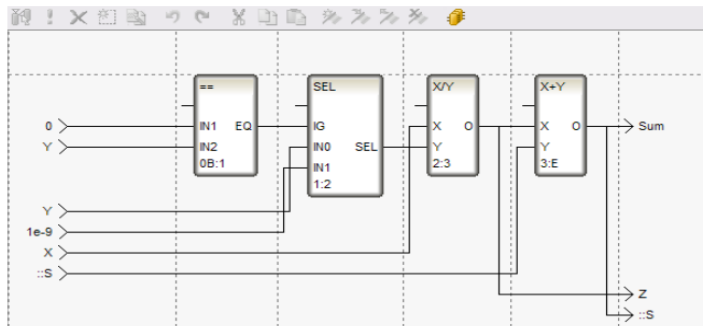


Рисунок 5.10 – Приклад реалізації FBD програми

5.4. Мова Techno SFC

SFC мову дозволяє створювати програму, оперуючи кроками і переходами. Під кроком слід розуміти підпрограму, написану на одній з мов, доступних в середовищі TRACE MODE, і виконує ту чи іншу дію. Перехід – умова, при виконанні якого виконується певний крок. На рис. 5.11, 5.12 зображені умовні графічні зображення кроку і відповідно переходу.

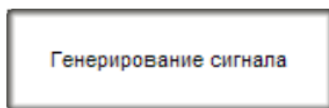


Рисунок 5.11 – Умовне графічне позначення кроку

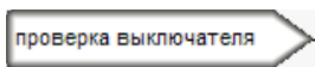


Рисунок 5.12 – Умовне графічне позначення переходу

Приклад SFC найпростішої програми наведено на рис. 5.13. Напрямок переходу від одного кроку до іншого вказується лінією зі стрілкою. Наступний етап виконується в разі, якщо виконується умова переходу. Лінія зі стрілкою, що з'єднує перехід з лінією, що відображає напрямок переходу, вказує на крок, який виконується, якщо умова переходу виконується.

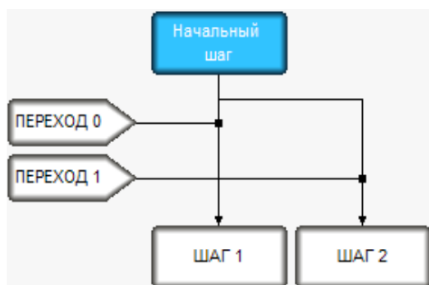


Рисунок 5.13 – Приклад найпростішої графічної програми SFC

Кожен крок і перехід можна виділити. Виділений крок (перехід) відображається синім кольором. При виділенні переходу виділяється додатково лінія переходу, на яку поширюється умова переходу. Можна виділити лінію переходу. Виділена лінія зображується червоним кольором. Лінія переходу виділяється разом з переходом, чинним на даній лінії переходу і розташованій вище інших переходів на виділеній лінії переходу. Крок (перехід) можна перейменувати, виділивши його і двічі клацнувши лівою кlawішею миші по ньому, після чого можна ввести нове ім'я кроку (переходу).

Об'єкт програма, аргументи і змінні програми створюються аналогічно іншим програмам. Вікно для створення SFC діаграми представлено на рис. 5.14.

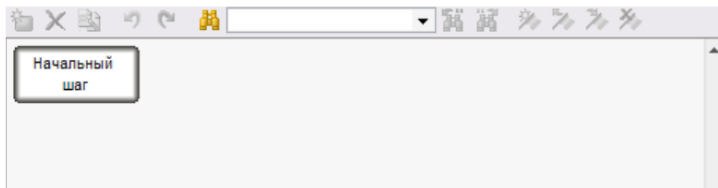

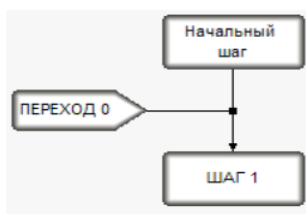


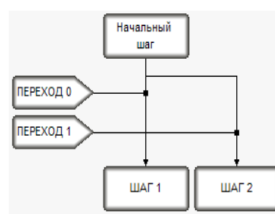
Рисунок 5.14 – Вікно створення SFC діаграми

З рис. 5.14 можна помітити, що в новій SFC діаграмі присутній один крок. Для створення нового кроку треба виділити крок, що передує йому. Можна або клацнути лівою клавішею миші на іконці  або викликати контекстне меню і вибрати створити крок/перехід. Буде створено новий крок, що виконується після виділеного раніше кроку, і перехід, який визначає умову переходу до новоствореного кроці.

Результат створення нового кроку наведено на рис. 5.15, а. Якщо існував крок, що виконується після виділеного блоку, то станеться розгалуження після створення нового кроку (рис. 5.15, б).



а)



б)

Рисунок 5.15 – Приклади створення кроків

Якщо все було правильно зроблено, вийде цикл (рис. 5.16). Умовою виконання циклу виступає перехід, створений разом з циклом.

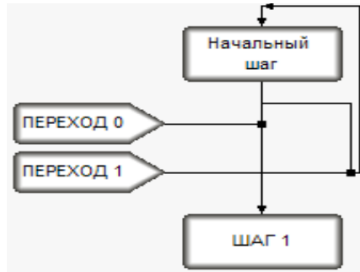



Рисунок 5.16 – Приклад циклу на SFC діаграмі

Розглянемо створення паралельного кроку. Для створення кроку, що виконується паралельно якомусь певному кроку (паралельно кроку 1, наприклад) слід виділити перехід або лінію, що виходить з нього, який визначає умову виконання паралельних кроків (в даному випадку перехід 0 – умова виконання кроку1), клацнути лівою клавішею миші на іконці  або викликати контекстне меню і вибрати створити крок/перехід. Буде створено новий крок (рис. 5.17), що виконується паралельно іншому кроку, перехід до якого був виділений при створенні нового кроку.

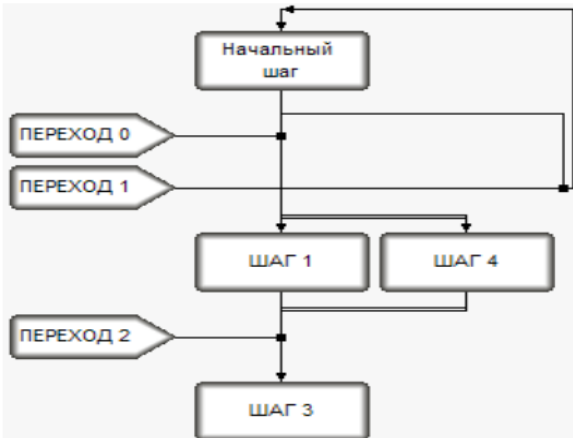

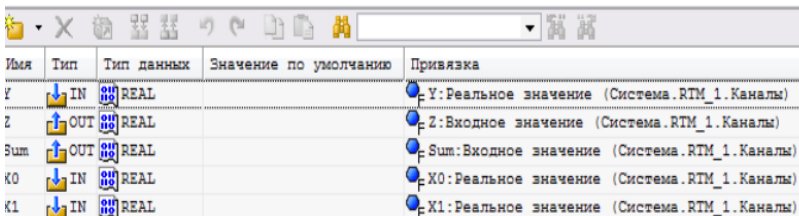


Рисунок 5.17 – Приклад створення паралельних кроків

Всі паралельні кроки повинні бути пов'язані з одним і тим же наступним переходом. Припустимо, створений крок 3, що виконується після кроку 1. Тоді необхідно натиснути ліву кнопку миші на нижньому краї кроку, паралельного кроку 1 (крок 2), перемістити курсор, не відпускаючи клавіші миші, на перехід, який визначає умову виклику кроку 3 або на лінію переходу, відпустити кнопку миші. Якщо все було правильно зроблено, кроки 1 і 2 будуть підключені паралельно.

Після створення діаграми необхідно описати кожен крок або перехід (зробити редагування кроку або переходу). Кожен крок і перехід - програма, написана на одній з мов середовища TRACE MODE. Для редагування кроку (переходу) необхідно виділити відповідний крок (перехід), клацнути лівою клавішею миші на іконці  або викликати контекстне меню і вибрати «Редагувати». Буде запропонований вибір програм, серед яких слід вибрати мову програмування, на якому буде описаний даний крок (умова переходу). Можна розкрити розділ SFC діаграма в навігаторі проекту. З'явиться розділ «Кроки та переходи», які містять всі кроки і переходи програми. Розкривши дані розділи і вибравши необхідний крок (перехід) можна зробити редагування кроку (переходу). Після вибору кроку або переходу в розділі «Кроки та переходи» буде запропонований вибір мови. Після вибору мови можна приступити до написання підпрограми.

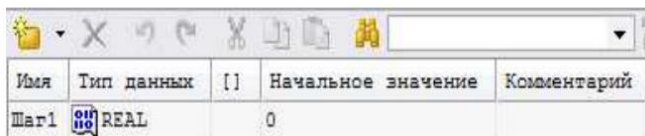
Як приклад розглянемо ділення суми аргументів X_0 і X_1 на аргумент Y з перевіркою на рівність нулю аргументу Y . У разі рівності аргументу Y нулю виконаємо ділення на константу $1e-9$, результат ділення дамо аргументу Z . Програми буде також обчислювати суму всіх результатів розподілу Sum. Перший крок буде полягати в знаходженні суми аргументів X_0 і X_1 . Другий крок - ділення на Z , за умови переходу: Z не дорівнює нулю. Третій крок - ділення на константу, $1e-9$ за умови: Z дорівнює нулю. Аргументи програми наведені на рис. 5.18.



Имя	Тип	Тип данных	Значение по умолчанию	Привязка
F	IN	REAL		Y: Реальное значение (Система.RTM_1.Каналы)
Z	OUT	REAL		Z: Входное значение (Система.RTM_1.Каналы)
Sum	OUT	REAL		Sum: Входное значение (Система.RTM_1.Каналы)
K0	IN	REAL		X0: Реальное значение (Система.RTM_1.Каналы)
K1	IN	REAL		X1: Реальное значение (Система.RTM_1.Каналы)

Рисунок 5.18 – Приклад аргумента програми

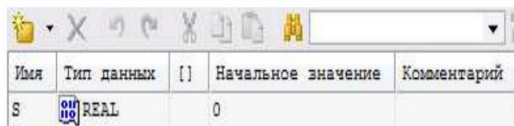
Оскільки необхідно передати результат складання з першого кроку, що виконує додавання X0 і X1, то створимо локальну змінну шаг 1 (рис. 5.19).



Имя	Тип данных	[]	Начальное значение	Комментарий
Шаг1	OUT	REAL	0	

Рисунок 5.19 – Приклад локальної змінної

Так як необхідно знайти суми всіх обчислених відношень, створимо глобальну змінну S, що зберігає попередню суму всіх знайдених відношень (рис. 5.20).



Имя	Тип данных	[]	Начальное значение	Комментарий
S	OUT	REAL	0	

Рисунок 5.20 – Приклад глобальної змінної

SFC діаграма зображена на рис. 5.21. Кожний крок і перехід може бути виконаний, використовуючи мову Techno ST.

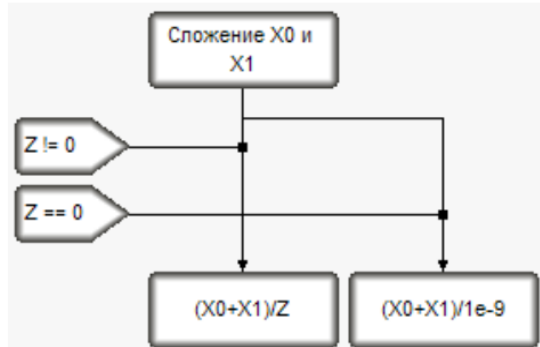


Рисунок 5.20 – SFC діаграма

Перший крок, що виконує додавання аргументів X_0 і X_1 матиме вигляд:

```

SFC_STEP «Сложение X0 и X1»
VAR_INPUT Y : REAL; END_VAR
VAR_OUTPUT Z : REAL; END_VAR
VAR_OUTPUT Sum : REAL; END_VAR
VAR_INPUT X0 : REAL; END_VAR
VAR_INPUT X1 : REAL; END_VAR
VAR Шаг1 : REAL := 0; END_VAR
Шаг1 = X0 + X1;
END_SFC_STEP
  
```

Перехід до кроку, що обчислює ділення на Z набуде вигляду:

```

SFC_TRANSITION «Z != 0» FROM( INITIAL_STEP ) TO( STEP_1 )
VAR_INPUT Y : REAL; END_VAR
VAR_OUTPUT Z : REAL; END_VAR
VAR_OUTPUT Sum : REAL; END_VAR
VAR_INPUT X0 : REAL; END_VAR
VAR_INPUT X1 : REAL; END_VAR
VAR Шаг1 : REAL := 0; END_VAR
Z != 0
END_SFC_TRANSITION
  
```

Сам крок, що виконує ділення на Z, набуде вигляду:

```
SFC_STEP «(X0+X1)/Z»  
VAR_INPUT Y: REAL; END_VAR  
VAR_OUTPUT Z: REAL; END_VAR  
VAR_OUTPUT Sum: REAL; END_VAR  
VAR_INPUT X0: REAL; END_VAR  
VAR_INPUT X1: REAL; END_VAR  
VAR Шаг1: REAL:= 0; END_VAR  
Sum = S + Шаг1 / Z;  
S = Sum;  
END_SFC_STEP
```

Перехід до кроку, що проводить ділення на константу 1e-9, набуде вигляду:

```
SFC_TRANSITION «Z == 0» FROM( INITIAL_STEP ) TO( STEP_2 )  
VAR_INPUT Y: REAL; END_VAR  
VAR_OUTPUT Z: REAL; END_VAR  
VAR_OUTPUT Sum: REAL; END_VAR  
VAR_INPUT X0: REAL; END_VAR  
VAR_INPUT X1: REAL; END_VAR  
VAR Шаг1: REAL:= 0; END_VAR  
Z == 0  
END_SFC_TRANSITION
```

Крок, що виконує ділення на константу 1e-9, набуде вигляду:

```
SFC_STEP «(X0+X1)/1e-9»  
VAR_INPUT Y: REAL; END_VAR  
VAR_OUTPUT Z: REAL; END_VAR  
VAR_OUTPUT Sum: REAL; END_VAR  
VAR_INPUT X0: REAL; END_VAR  
VAR_INPUT X1: REAL; END_VAR  
VAR Шаг1: REAL := 0; END_VAR  
Sum = S + Шаг1 / 1e-9;  
S = Sum;  
END_SFC_STEP
```

5.5. Мова Techno II

Програма на мові Techno II – послідовність інструкцій. Кожна інструкція починається з нового рядка, містить оператор з модифікатором, в разі деяких операторів наводиться один або більше операндів. Компілятор не чутливий до регістру. Інструкції ADD 10 15 і Add 10 15 рівнозначні.

У даній мові вводиться поняття акумулятора, під яким слід розуміти сховище поточного результату.

Як операнди можуть виступати:

- 1) змінні;
- 2) константи (див. мову Techno ST);
- 3) ім'я мітки;
- 4) ім'я функції.

Під модифікатором слід розуміти літери N, C, X, які приписуються праворуч до імені деяких операторів. Модифікатор N позначає логічне заперечення операнда, C позначає, що інструкція виконується, якщо результат попередньої операції порівняння правдивий, модифікатор X вказує на те, що інструкція виконується, якщо акумулятор містить значення true.

Приклад використання модифікатора:

OR b//акумулятор АБО b

ORN b// акумулятор АБО НЕ b

Оператори, які припускають наявність одного або двох операндів (арифметичні, логічні операції) допускають опустити перший операнд, тоді як перший операнда виступає значення, що зберігається в акумуляторі. Випадок, коли використовуються обидва операнди називається двоадресним режимом, випадок, коли перший операнд замінюється акумулятором – одноадресним режимом.

Приклад двоадресного режиму: ADD 10 15 // додавання 10 і 15.

Приклад того ж складання при одноадресному режимі:

LD 10 // привласнення акумулятору числа 10

ADD 15 // додавання 10 і 15

Розглянемо оператори мови Techno PL. Оператори для обміну з акумулятором наведені в таблиці 5.6.

Таблиця 5.6

Оператори для обміну з акумулятором

Позначення оператора	Синтаксис оператора	Припустимі модифікатори	Дія
LD	LD{операнд}	N	привласнює акумулятору значення операнда
ST	ST{операнд}	N	привласнює операнду значення акумулятора

Приклад обміну даними з акумулятором:

LD X // привласнення акумулятору операнда *X*

MULT K // множення *X* на *K*

ST X // привласнення операнду *X* значення, що зберігається в акумуляторі.

Логічні оператори наведені в таблиці 5.7.

Таблиця 5.7

Логічні оператори

Позначення оператора	Синтаксис оператора	Припустимі модифікатори	Дія
S	S{операнд}	-	привласнює операнду значення true
R	R{ операнд }	-	привласнює операнду значення false
AND	AND{ операнд 1 } { операнд 2 }	N	логічне «І»
OR	OR{ операнд 1 } { операнд 2 }	N	логічне «АБО»
XOR	XOR{ операнд 1 } { операнд 2 }	N	логічне «виключне АБО»

Приклад логічних операцій:

S X // привласнення операнду *X* значення true

R Y // привласнення операнду *Y* значення false

AND X Y // X «I» Y

Арифметичні оператори наведені в таблиці 5.8.

Таблиця 5.8

Арифметичні оператори

Позначення оператора	Синтаксис оператора	Припустимі модифікатори	Дія
ADD	ADD {операнд1} {операнд 2}	-	складання операндів
SUB	SUB {операнд1} {операнд 2}	-	віднімання операнда 2 з операнда 1
MUL	MUL {операнд1} {операнд 2}	-	множення операндів
DIV	DIV {операнд1} {операнд 2}	-	ділення операнда 1 на операнд 2

Приклад виконання арифметичних операцій:

MUL X K

ADD C

Оператори порівняння наведені в таблиці 5.9.

Таблиця 5.9

Оператори порівняння

Позначення оператора	Синтаксис оператора	Припустимі модифікатори	Дія
GT	GT {операнд1} {операнд 2}	-	повертає true, якщо операнд 1 більше операнда 2
GE	GE {операнд1} {операнд 2}	-	повертає true, якщо операнд 1 не менше операнда 2
EQ	EQ {операнд1} {операнд 2}	-	повертає true, якщо операнд 1 дорівнює операнду 2
NE	NE {операнд1} {операнд 2}	-	повертає true, якщо операнд 1 не дорівнює операнду 2
LE	LE {операнд1} {операнд 2}	-	повертає true, якщо операнд 1 не більше операнда 2
LT	LT {операнд1} {операнд 2}	-	повертає true, якщо операнд 1 менше операнда 2

Оператори переходу та виклику функцій наведено в таблиці 5.10.

Таблиця 5.10

Оператори переходу та виклику

Позначення оператора	Синтаксис оператора	Припустимі модифікатори	Дія
JMP	JMP {мітка}	C, X	перехід до рядка із вказаною міткою
CAL	CAL {ім'я функції (параметри)}	C, X	виклик функції
RET	RET	C, X	вихід з програми

Можливі такі наступні модифікації операторів переходу та виклику:

JMPC – умовний перехід, що виконується, якщо результат попередньої операції порівняння дійсний;

JMPX – умовний перехід, що виконується, якщо акумулятор містить значення true;

CALC – умовний виклик функції, що виконується, якщо результат попереднього порівняння дійсний;

CALX – умовний виклик функції, що виконується, якщо акумулятор містить значення true

RETC – умовний вихід з програми, що виконується, якщо результат попередньої операції порівняння дійсний;

RETX – умовний вихід з програми, що виконується, коли акумулятор містить значення true.

Приклад виконання порівняння та переходу:

EQ Z 0 // перевірка на рівність Z нулю

JMPC label // при рівності нулю перехід до рядка з міткою label

...

label: LD 1e-9 // привласнення аргументу Z значення 1e-9

ST Z

Приклад розподілу, наведеного на мові Techno SFC, з використанням мови Techno IL. Оскільки існує оператор S в

мові Techno PL, глобальну змінну перейменуємо з S на Sum0.
Тоді програма на мові Techno PL набуде вигляду:

```
PROGRAM
VAR_INPUT Y: REAL; END_VAR
VAR_OUTPUT Z: REAL; END_VAR
VAR_OUTPUT Sum: REAL; END_VAR
VAR_INPUT X0: REAL; END_VAR
VAR_INPUT X1: REAL; END_VAR
VAR Шаг1 : REAL:= 0; END_VAR
ADD X0 X1// Складання аргументів
ST Шаг1// Привласнення результату складання змінної
Шаг1
NE Y 0// Перевірка на рівність нулю
JMPC Деление_на_Y// Перехід до ділення на Z, якщо Z не
дорівнює //нулю ділення на 1e-9
DIV Шаг1 1e-9
ST Z// Привласнення аргументу Z обчисленого відношення
JMP Сложение// Перехід до складання
Деление_на_Y: DIV Шаг1 Y// Ділення на Z
ST Z// Привласнення аргументу Z обчисленого відношення
Сложение: ADD Z Sum0// Отримання суми всіх
результатів відношення
ST Sum0// Привласнення суми всіх результатів складання
аргументу Sum і глобальне змінній S
ST Sum
END_PROGRAM
```

5.6. Контрольні питання

1. Які мови програмування підтримує TRACE MODE?
2. Як визначається напрямок передавання даних?
3. Синтаксис мови Techno ST?;
4. Арифметичні оператори мови Techno ST?
5. Синтаксис мови Techno FBD?
6. Наведіть приклад функціонального блоку мови Techno FBD.
7. Які блоку мови Techno FBD ви знаєте?
8. Синтаксис мови Techno SFC?
9. Наведіть приклад умовних графічних позначенн мови Techno SFC.
10. Синтаксис мови Techno IL?
11. Як операнди можуть виступати у мові Techno IL?
12. Що таке модифікатор у мові Techno IL?
13. Оператори для обміну з акумулятором у мові Techno IL?

6. РОЗРОБКА АСК ТП НА ПРИКЛАДАХ

6.1. Підключення стандартних графічних елементів з бібліотеки

Для використання стандартних графічних елементів перед роботою в SCADA системі TRACE MODE користувач повинен підключити бібліотеку tmdevenv.tmul. Щоб скористатися даною бібліотекою необхідно скопіювати файл tmdevenv.tmul з піддиректорії %TRACE MODE% \ Lib в директорію %TRACE MODE%.

Після копіювання в нижній копії Навігатора (рис. 6.1) проекту в лівому вікні можна відкрити бібліотеку стандартних графічних компонентів: Библиотеки_компонентов-Регульована система-Library_1-Object_1-Resources. Нижче останнього пункту з'явиться набір назв різних технологічних елементів на англійській мові (Valves – Клапани, Tanks – Ємності і ін.).

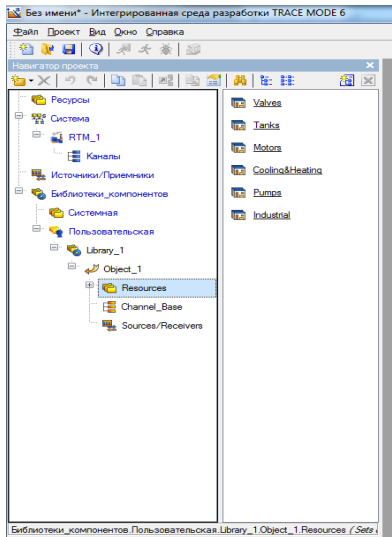


Рисунок 6.1 – Підключення бібліотеки стандартних елементів

Кожен з цих елементів потрібно перенести мишкою із нижнього вікна в верхнє на слово «Ресурси», щоб використовувати їх в поточному проекті. Після виконання даних дій в дереві побудови «Ресурсов» з'явиться побудова з назвою «Графические_элементы_1» (рис. 6.2).

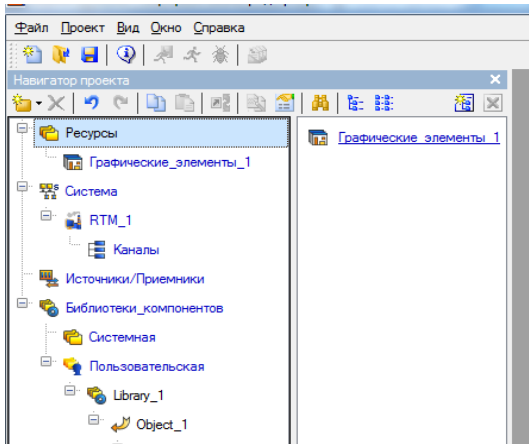




Рисунок 6.2 – Відображення підключеного ресурса

6.2. Створення проекту та налаштування каналів

В даному підрозділі навчимося створювати генератор сигналу (синусоїдального, пилообразного, трикутного), прив'язати генератор до каналу, і вивести на екран масштабовані значення за допомогою компонентів візуального відображення (стрілочного приладу, тренда, тексту).

На першому етапі створимо новий проект **Створення проекту Trace Mode**. Запустіть програму TRACE MODE 6 (файл tmdevenv.exe). Для створення нового проекту клацніть лівою клавішею миші на іконці  або на рядку **Новий** в меню **Файл**.

Створений проект набуде вигляду, зображеного на рис. 6.3. При відсутності навігатора проекту клацніть лівою

клавiшею мишi на рядку *Навiгатор проекту* меню *Вид*. Збережiть створений проект, клацнувши лiвою клавiшею мишi на iконцi  або на рядку *Сохранить* або *Сохранить как* в меню *Файл*.

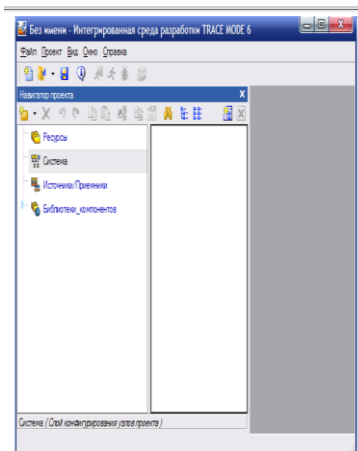


Рисунок 6.3 – Створений новий проект

На другому кроцi створення вузла. Для цього видiлiть рядок *Система* в навігаторі проекту (рис. 6.4). Викличте контекстне меню клацанням правої клавiшi мишi. У контекстному меню виберiть рядок *Создать узел*. Серед запропонованих типiв вузлiв виберiть *RTM*.

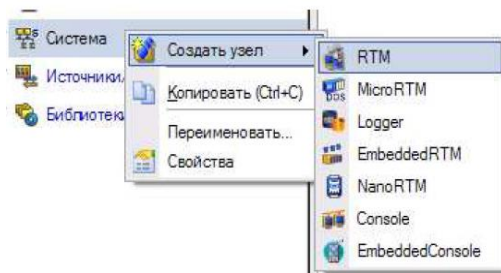


Рисунок 6.4 – Створення вузла RTM

Навігатор проекту в розділі *Система* відображає створений вузол. Виділіть створений вузол **RTM** і клацніть лівою клавішею миші по ньому. З'явиться можливість змінити ім'я вузла. Приклад створеного вузла **RTM** зображено на рис. 6.5.

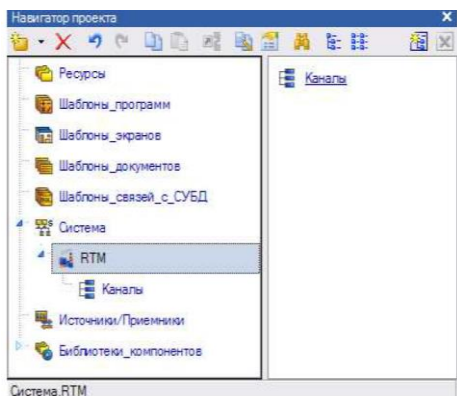


Рисунок 6.5 – Створений вузол

На наступному кроці створимо канал. Виділіть групу **Каналы** RTM вузла. Викличте контекстне меню. У контекстному меню виберіть рядок **Создать компонент**. Серед запропонованих компонентів виберіть **Канал_FLOAT**, приклад створення наведено на рис. 6.6.

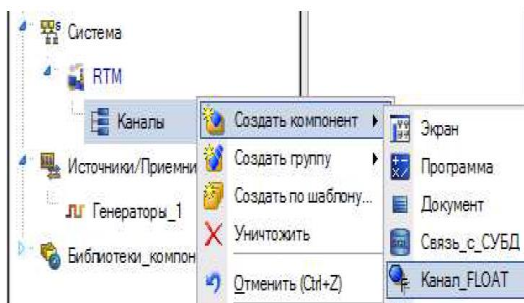


Рисунок 6.6 – Створення Канала FLOAT

Для ускладнення завдання розрахуйте множник K і зміщення Z . Перетворення сигналу буде здійснюватися за формулою:

$$Y = K \cdot X + Z,$$

де:

Y – результат перетворення;

X – вхідне значення каналу;

K – множник;

Z – зміщення.

Вхідне значення каналу змінюється в діапазоні $[0; 100]$. Знайдіть значення множника K і зміщення Z , які дозволять отримати необхідний діапазон зміни результату перетворення (діапазон сигналу).

Прикладом даної реалізації може послужити необхідність забезпечення сигналу $[10; 10]$, при множнику $K = 0,2$, зміщення $Z = -10$.

Для редагування створеного каналу необхідно двічі клацнути лівою кlawішею миші на імені створеного каналу (канал#1 на рис. 6.7).

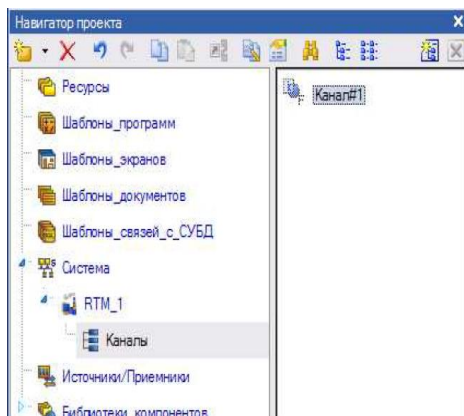


Рисунок 6.7 – Редагування каналу

Відкриється вікно для редагування каналу (рис. 6.8)

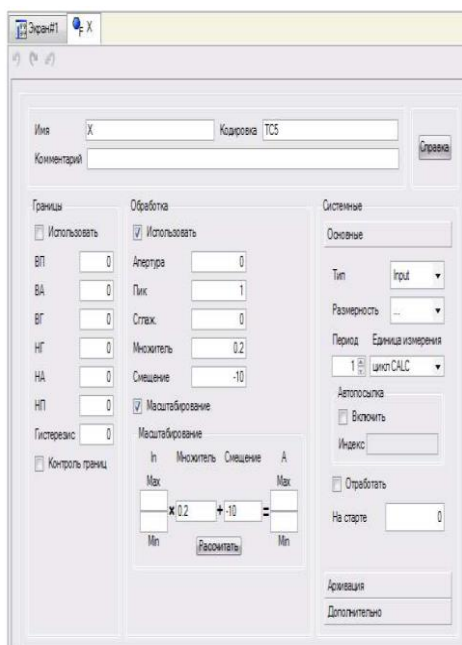


Рисунок 6.8 – Вікно редагування каналу

В поле **Имя** введіть нове ім'я каналу. Встановіть прапорець **Використовувати** на панелі **Обработка**. Встановіть **Апертуру** рівною 0, **Пік** рівним 1, згладжування рівним 0, обчислені значення множника K і зміщення Z . Переконайтеся, що тип каналу - **Input**.

На наступному етапі створимо генерацію сигналів. Для цього необхідно створити групу генераторів. Для цього виділіть рядок **Источники/Приемники** навігатора проекту. Виділіть контекстне меню. В меню оберіть рядок **Создать группу**. Серед запропонованих груп виберіть **Генераторы** (рис. 6.9).

Виділивши створену групу (генератори 1) і клацнувши лівою клавішею миші, змініть ім'я, наприклад, на «генератор».

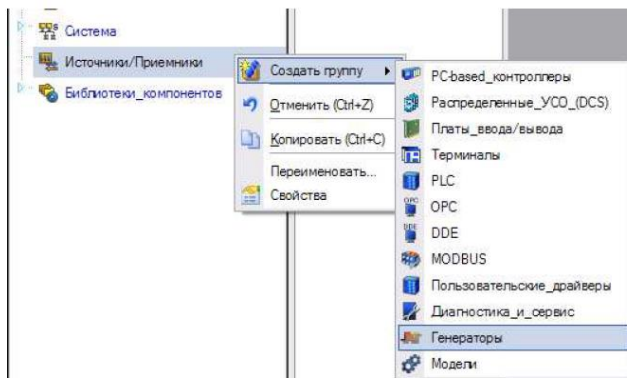


Рисунок 6.9 – Створення групи для генераторів

Виділіть створену групу. Викличте контекстне меню. У контекстному меню виберіть рядок створити компонент. Серед запропонованих генераторів виберіть тип генератора (рис. 6.10). Виділивши створений генератор і клацнувши лівою клавішею миші, змініть ім'я генератора, наприклад, на «Пила».

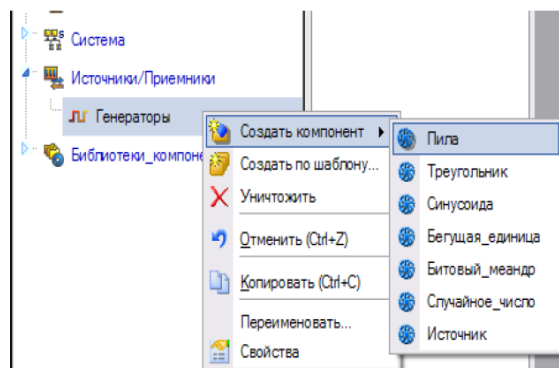




Рисунок 6.10 – Створення генератора «Пила»

Наступною дією необхідно прив'язати генератор до каналу. Для цього необхідно клацнути лівою клавшею миші по іконці . Відкриється ще одне вікно навігатора проекту. У лівій частині верхнього навігатора проекту виберіть групу **Канали** RTM вузла. В лівій частині нижнього навігатора проекту виберіть групу **Генераторы** групи **Источники/Приемники** (рис. 6.5). Натисніть ліву кнопку миші на створеному раніше генераторі. Не відпускаючи лівої кнопки миші, наведіть курсор миші на створений канал і відпустіть ліву кнопку миші. Іконка каналу повинна змінитися на  (рис. 6.11).

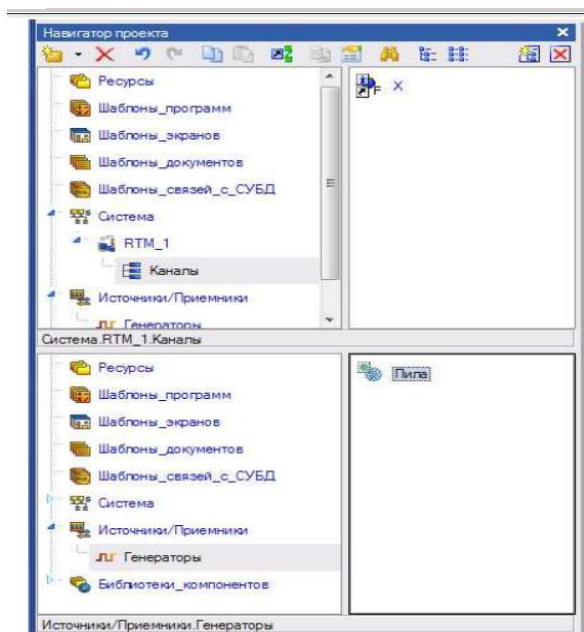



Рисунок 6.11 – Прив'язка генератора до каналу

Закрити одне з створених вікон навігатора проекту можна клацнувши лівою клавiшею миші по іконці .

На наступному етапі створимо екран. Для цього виділіть групу **Канали**, викличте контекстне меню. У контекстному меню виберіть рядок **Создать компонент**. Серед запропонованих компонентів слід вибрати **Экран**. Послідовність дій для вибору та створення екрана представлений на рис. 6.12.

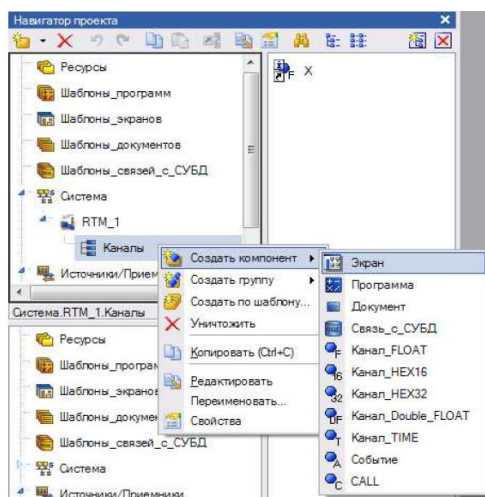





Рисунок 6.12 – Послідовність створення екрана

Виділіть створений екран і клацніть лівою клавiшею миші, змініть ім'я екрану. Наприклад, можна змінити ім'я на «екран». Двічі клацніть лівою клавiшею миші по створеному екрану. Відкриється вікно для редагування шаблону екрану (рис. 6.13).

Для візуалізації подання одержуваної інформації створимо об'єкт «стрілочний прилад». Для цього необхідно клацнути лівою клавiшею миші по іконці . Якщо на панелі інструментів замість іконки  присутня іконка , клацніть по ній правою клавiшею миші.

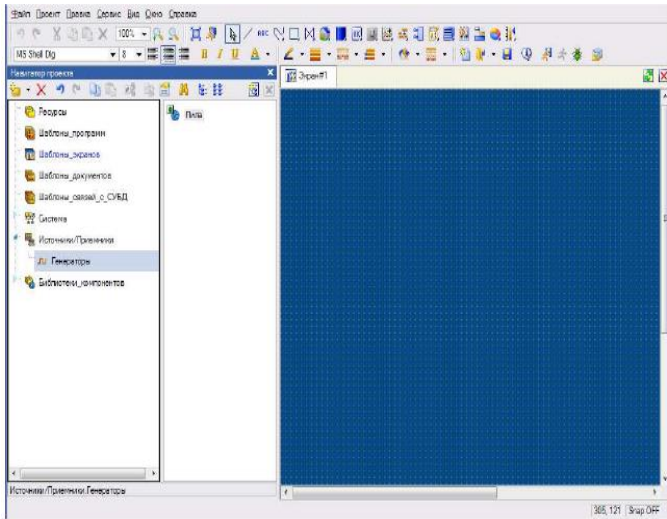



Рисунок 6.13 – Вікно редагування шаблону створеного екрана

Серед запропонованих об'єктів оберіть **Стрілочний прибор** . Після цього вибору клацніть лівою клавішею миші в тому місці екрана, де повинен розташовуватися один з кутів стрілочного приладу. Перемістіть курсор в положення, що відповідає положенню протилежного кута стрілочного приладу і клацніть лівою клавішею миші. Буде створено стрілочний прилад на екрані (рис. 6.14).

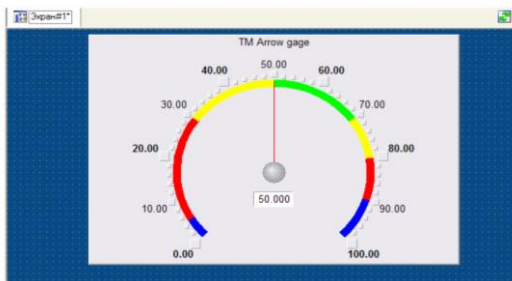


Рисунок 6.14 – Створення «стрілочного прилада» на шаблоні екрана

Для налаштування «стрілочного приладу» можна скористатися двома способами. Якщо вікно властивостей об'єкта відкрито, виділіть **Стрілочний прилад**, в іншому випадку двічі клацніть лівою клавішею миші по створеному стрілочному приладу. Відкриється вікно властивостей об'єкта (рис. 6.15).

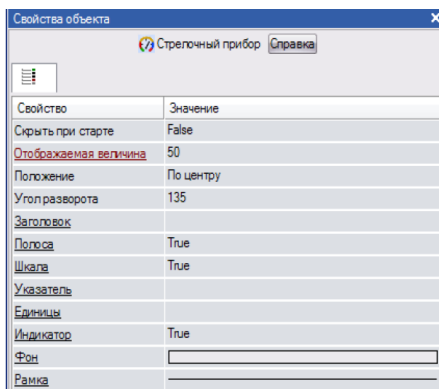


Рисунок 6.15 – Вікно налаштування властивостей об'єкта «Стрілочний прилад»

Розкрийте розділ **Заголовок** подвійним клацанням лівої кнопки миші по підкресленому рядку **Заголовок**. З'явиться поле **Текст**, введіть текст заголовка, який буде виведений на стрілочному приладі. Розкрийте розділ **Смуга** подвійним клацанням лівої кнопки миші по підкресленому рядку **Смуга**. В поле **Верхний предел шкалы** введіть верхню межу діапазону, в поле **Нижний предел шкалы** – нижню межу діапазону. Заповніть поля HL, HA, HW, LL, LA, LW будь-якими значеннями, що задовольняють умові:

Нижня межа шкали <LL <LA <LW <HW <HA <HL
<Верхня межа шкали.

Приклад заповнення полів властивостей стрілочного приладу наведено на рис. 6.16.

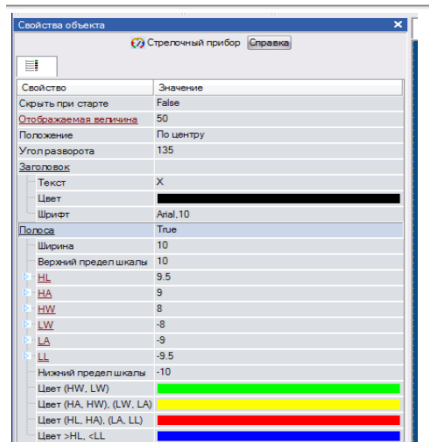


Рисунок 6.16. – Пример заполнения полей объекта Стрелочный прибор

Зробимо прив'язку об'єкта до каналу. Для цього необхідно виконати такі дії. Розкрийте розділ **Отображаемая величина**. Для цього двічі клацніть лівою клав'яшею миші на підкресленому рядку **Отображаемая величина**. Клацніть лівою клав'яшею миші в полі **Привязка**. Відкриється вікно властивостей прив'язки з порожньою таблицею. Встановіть тип IN у створеному аргументу ARG_000. Двічі клацніть лівою клав'яшею миші в стовпці **Привязка** таблиці. Відкриється вікно прив'язки. У лівій частині відкритого вікна виділіть **Канал** RTM вузла, створений раніше. У правій частині вікна виберіть аргумент **Реальное значение** (рис. 6.17).

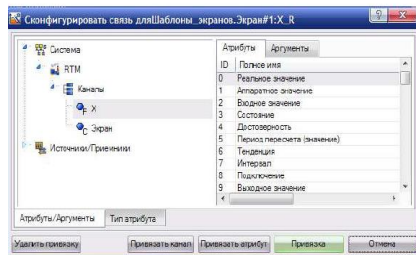


Рисунок 6.17 – Створення прив'язки

Клацніть лівою клавiшею миші по кнопці **Привязка**. Вікно властивостей прив'язки набуде вигляду, зображеного на рис. 6.18. В поле **Имя** можна ввести нове ім'я каналу аргументу. Клацніть лівою клавiшею миші по кнопці **Готово**. В полі **Привязка** буде виведено ім'я створеного аргументу.

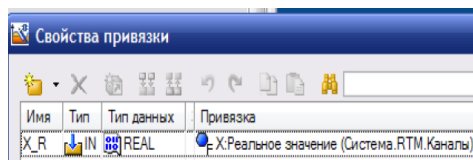








Рисунок 6.18 – Результати прив'язки

Тепер необхідно створити тренд. Для створення тренда клацніть лівою клавiшею миші по іконці . Якщо замість неї на панелі інструментів іконка архівний тренд , тренд XY  або архівна гістограма , то необхідно клацнути правою клавiшею миші на відповідній іконці та обрати тренд . Клацніть лівою клавiшею миші по іконці .

Для розміщення тренда на екрані клацніть лівою клавiшею миші там, де повинен розташовуватися один з кутів тренда. Переведіть курсор в положення, де повинен розташовуватися протилежний кут тренда і клацніть лівою клавiшею миші (рис. 6.19).

Для налаштування тренда необхідно виконати такі дії. Виділіть створений тренд. Відкриється вікно властивостей тренда. Спочатку буде відкрита закладка **Основные свойства**. В поле **Заголовок** введіть текст, який ви хочете вивести в якості заголовка. Перейдіть на закладку **Кривые**. Виділіть рядок **Кривые** і викличте контекстне меню. У контекстному меню виберіть **Кривая** (рис. 6.20).

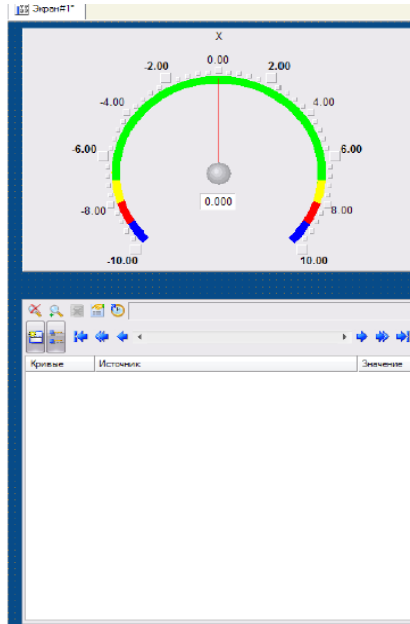


Рисунок 6.19 – Створення тренда

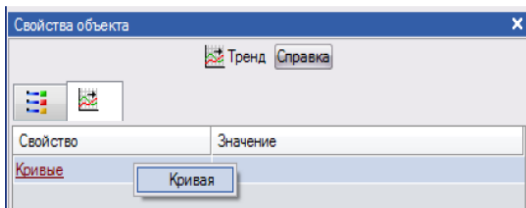


Рисунок 6.20 – Створення кривої

У полі **Имя** введіть ім'я створеної кривої. В поле **Мак. значение** введіть значення верхньої межі діапазону, в поле **Мин. значение** – нижньої межі діапазону. Для прив'язки кривої клацніть лівою клавішею миші в полі **Привязка**. Відкриється вікно, зображене на рис. 6.18. Виберіть створений раніше

аргумент і клацніть лівою клавiшею миші по кнопці **Готово**. Тренд прийме зовнішній вигляд, зображений на рис. 6.21.

Для збільшення розмірів графіка наведіть курсор на межу легенди та графіка. Курсор прийме форму двох паралельних ліній зі стрілками. Натисніть ліву кнопку миші. Не відпускаючи лівої кнопки миші, опустіть курсор вниз і відпустіть ліву кнопку миші. Тренд набуде вигляду, зображеного на рис. 6.22.

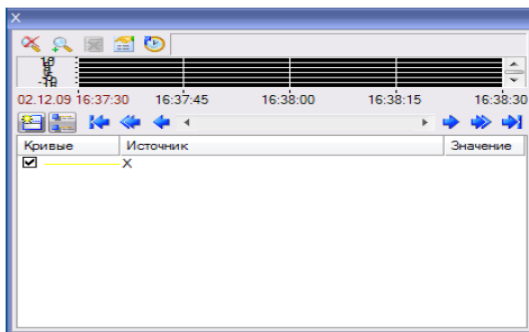


Рисунок 6.21 – Створений тренд

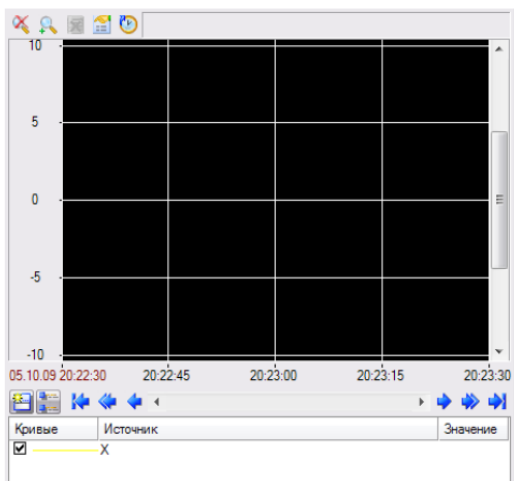


Рисунок 6.22 – Результат створення тренда

Створимо опис, для цього використаємо **Объект текст**. Клацніть лівою клавішею миші на іконці **ABC**. Клацніть лівою клавішею миші там, де повинен розташовуватися один з кутів об'єкта текст. Переведіть курсор миші в положення, де повинен розташовуватися інший край об'єкта текст і клацніть лівою клавішею миші. Поруч аналогічно розмістіть ще один об'єкт текст (рис. 6.23).

Клацніть лівою клавішею миші по іконці .

Виконаємо налаштування об'єкта текст. Для цього виділіть об'єкт текст, розташований зліва. В поле **Текст** введіть ім'я каналу RTM вузла. Виділіть об'єкт текст, розташований праворуч. Відкрийте розділ **Текст** подвійним клацанням по підкресленому рядку **Текст**. В поле **Вид динамізації** виберіть значення (рис. 6.24).

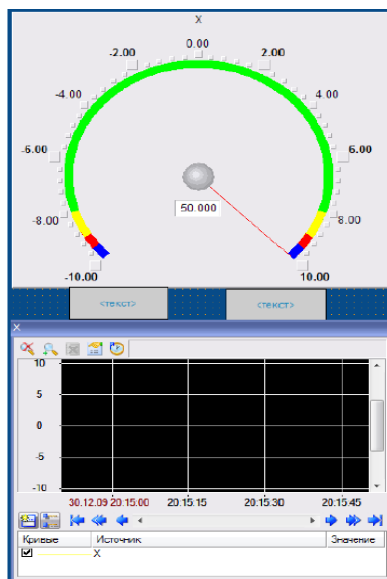


Рисунок 6.23 – Приклад розміщення **Объект текст**

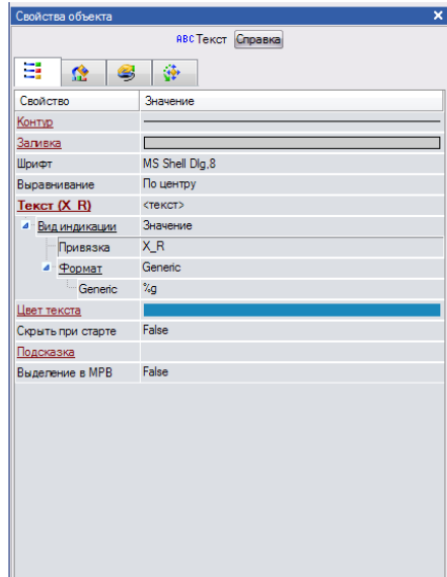






Рисунок 6.24 – Редагування Объект текст

Клацніть лівою клавішею миші в полі **Привязка**, що з'явилося. Відкриється вікно, зображене на рис. 6.18. Виберіть створений раніше аргумент і клацніть лівою клавішею миші по кнопці **Готово**. Об'єкти текст набудуть вигляду, зображеного на рис. 6.25.



Рисунок 6.25 – Об'єкт текст

Проведемо перевірку створеного проекту. Для його запуску клацніть лівою клавішею миші по іконці  або по

рядку **Сохранить для MPB** в меню **Файл**. В навігаторі проекту виділіть створений RTM узел. Клацанням лівої кнопки миші по іконці  відкрийте профайлер. Якщо профайлер не запуситься, запустіть файл rts.exe, клацніть лівою клавiшею миші по іконці  або рядку **Открыть** в меню **Файл**. Для запуску проекту клацніть лівою клавiшею миші по іконці . В результаті запуску профайлера стрілка стрілочного приладу повинна зміститися, тренд буде графік, а об'єкт текст виводить поточне значення створеного каналу (рис. 6.26).

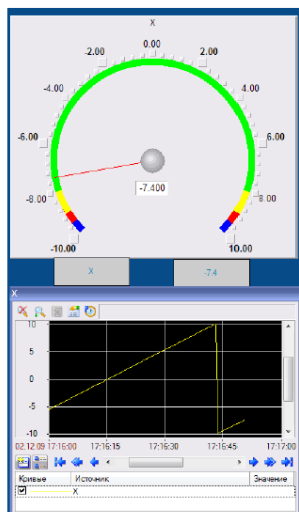


Рисунок 6.26 – Результат запуску проекта

6.3. Створення статичних і динамічних об'єктів

В даному підрозділі познайомимося зі стандартними об'єктами, призначеними для створення статичних і динамічних зображень. Для спрощення створення розіб'ємо завдання на такі підзадачі:

1. Створити новий проект. В даному проекті створити генератор, прив'язаний до каналу, значення якого визначається рівнем продукту в ємності.
2. Створити екран, розташувати на ньому тренд, який буде залежність рівня продукту в часі.
3. Створити статичне зображення ємності в розрізі, насоса, трьох труб, по одній трубі продукт надходить в ємність, за іншою впливає з неї. Друга труба з'єднана з третьою через насос.
4. Створити динамічний об'єкт, що імітує заповнення ємності, використовуючи графічний файл.

Для вирішення поставлених підзадач виконаємо такі кроки:

1. Створіть канал, який буде пропорційний рівню продукту в ємності. Назвемо цей канал рівнем, для кращого сприйняття.
2. Створіть генератор синусоїдального сигналу.
3. Проведіть прив'язку створеного генератора до створеного каналу.
4. Створіть екран.
5. Створіть тренд, налаштуйте криву і зробіть прив'язку до створеного каналу.

Приклад екрану, створеного при виконанні пунктів 1-5 наведено на рис. 6.27.

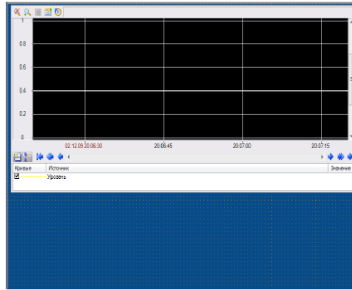







Рисунок 6.27 – Результат виконання кроків 1-5

Приступимо до створення статистичного зображення. На першому етапі створимо рамку. Для створення рамки клацніть лівою клавшею миші по іконці . Якщо немає зазначеної іконки на панелі інструментів, то клацніть правою клавшею миші по одній з іконок: , , або . Серед запропонованих об'єктів виберіть рамку . (рис. 6.28).

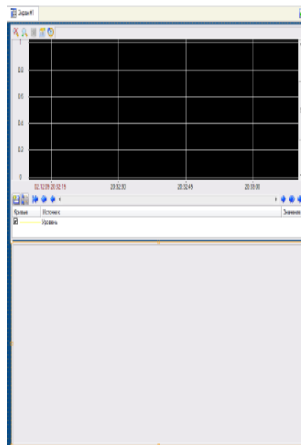
















Рисунок 6.28 – Результат створення рамки

Клацанням лівої клавiші миші задайте протилежні кути рамки. Для переходу в режим редагування клацніть лівою клавiшею миші по іконці .

Тепер приступимо до створення ємності. Клацніть лівою клавiшею миші по іконці . Якщо даної іконки немає на панелі інструментів, то клацніть правою кнопкою миші по одній з іконок: , , , , , , , ,  або . Серед запропонованих об'єктів виберіть ємність . Після вибору інструменту помістіть ємність на екрані. Задайте протилежні кути ємності клацанням лівої кнопки миші, клацніть лівою клавiшею миші по іконці . Ємність набуде вигляду, зображеного на рис. 6.29.

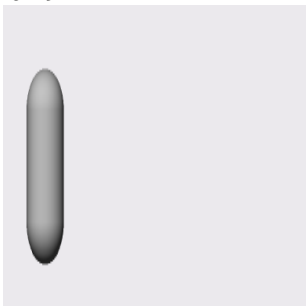


Рисунок 6.29 – Ємність

Виділіть створену ємність або двічі клацніть по ній лівою клавiшею миші. Відкриється вікно властивостей об'єкта. В поле **Товщина стенок** задайте товщину більше 0. Ємність буде зображена в розрізі (рис. 6.30).

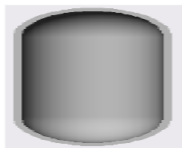




Рисунок 6.30 – Вид ємності при товщині стінки 0

В полі **Верхний и нижний край** виберіть необхідний вид краю. Наприклад, верхній край набуде вигляду , а нижній – . Для задання матеріалу ємності розкрийте розділ **Материал**, двічі клацнувши лівою клавішею миші по підкресленому рядку **Материал**, якщо цей розділ не розкритий. В полі **Выбрать из списка** виберіть значення true. В полі **Материал** виберіть необхідний матеріал, наприклад, хром. В полі **Стандартная текстура** виберіть необхідну текстуру, наприклад, гравіювання. При необхідності можна додати інші об'ємні фігури.






Для розміщення конуса клацніть правою клавішею миші по іконці  і серед запропонованих інструментів виберіть . Розмістіть на екрані конус, задавши протилежні кути прямокутника, в який буде вписаний конус, клацанням лівої кнопки миші. Виділіть створений конус. Якщо не відкриється вікно властивостей конуса, двічі клацніть лівою клавішею миші по ньому. В полі **Толщина стенок** задайте ту ж товщину, що і у ємності. Розкрийте розділ **Материал** подвійним клацанням миші по підкресленому рядку **Материал**, якщо він ще не розкритий. Виберіть значення **true** в полі **Выбрать из списка**. Виберіть необхідний матеріал в полі **Материал**, наприклад, олово. В полі **Стандартная гравировка** задайте гравіювання, наприклад, шліфування. Ємність набуде вигляду, зображеного на рис. 6.31



Рисунок 6.31 – Приклад отриманої ємності

На наступному етапі приступимо до створення насоса. Клацніть правою клавiшею миші по іконці . Серед запропонованих інструментів виберіть . Для розміщення насоса на екрані задайте протилежні кути прямокутника, в який буде вписаний насос клацанням лівої кнопки миші. Клацніть лівою клавiшею миші по іконці . Виділіть насос. Якщо не відкриється вікно властивостей насоса, двічі клацніть лівою клавiшею миші по створеному насосу. Розкрийте розділ **Матеріал**, двічі клацнувши по підкресленому тексту **Матеріал**, якщо він не розкритий. В полі **вибрати з списку** встановить значення **true**. В полі **Матеріал** виберіть необхідний матеріал, наприклад, пластик чорний. В поле форма насоса виберіть потрібну форму насоса. Статичне зображення набуде вигляду, зображеного на рис. 6.32.

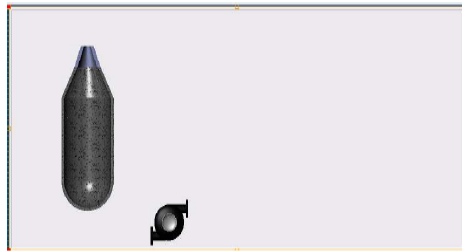





Рисунок 6.32 – Розміщення насоса

Для створення візуального зв'язку створимо труби, для цього клацніть правою клавiшею миші по іконці . Серед запропонованих інструментів виберіть . Створіть трубу, через яку товар надходить в ємність і по якій з ємності тече в насос. Для цього клацанням лівої кнопки миші відзначте місце розташування початку труби. Переведіть курсор миші в положення вигину труби і знову клацніть лівою клавiшею миші. Таким чином, відзначаються всі точки вигину труби. Коли курсор переведений в положення, де розміщується кінець труби,

клацніть правою кнопкою миші, завершуючи створення поточної труби. Створіть аналогічно трубу, через яку товар надходить в насос і впливає з нього. Для редагування властивостей кожної труби виділіть трубу. Якщо не відкриється вікно властивостей, двічі клацніть лівою клавшею миші по трубі. В полі **Толщина** підберіть товщину кожної труби, яка краще буде підходити для рисунка. В полі **Базовий цвет** виберіть необхідний колір, наприклад, . Статичне зображення набуде вигляду, вказаного на рис. 6.33

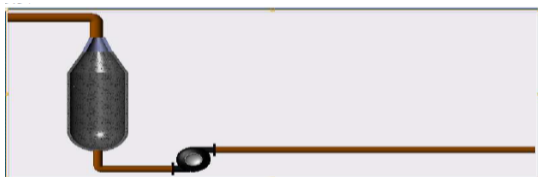


Рисунок 6.33 – Статичне зображення труби

Розглянемо послідовність створення динамічного зображення, для цього необхідно імпортувати зображення. Виділіть рядок **Ресурси** навігатора проекту. Викличте контекстне меню. Оберіть рядок **Создать группу**. Серед запропонованих груп оберіть **Картинки** (рис. 6.34).

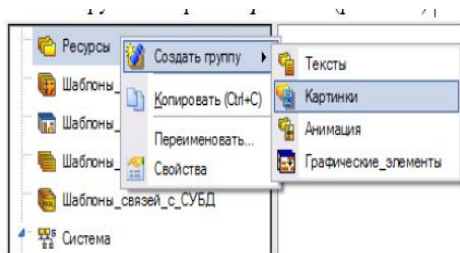


Рисунок 6.34 – Створення групи Картинки

Виділіть створену групу **Картинки**, викличте контекстне меню, оберіть **Создать компонент**. Серед запропонованих компонентів оберіть **Библиотека_Изображений** (рис. 6.35).

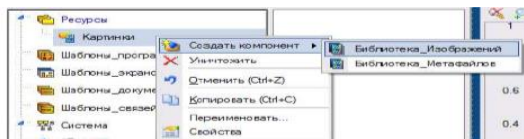


Рисунок 6.35 – Створення Библиотеки_Изображений

Для імпортування зображення двічі клацніть лівою клавішею миші по створеній бібліотеці зображень. Відкриється порожнє вікно. В даному порожньому вікні викличте контекстне меню і виберіть **Импортировать**. Відкриється діалогове вікно для відкриття графічного файлу. Відкрийте графічний файл зображення, яке буде використовуватися в подальшому.








Приступимо до створення динамічної заливки. Створіть багатокутник. Клацніть лівою клавішею миші по іконці . Якщо на панелі інструментів немає іконки , клацніть правою клавішею миші по одній з іконок: , , , . Серед запропонованих інструментів виберіть . Для розміщення багатокутника клацніть лівою клавішею миші там, де повинен розташовуватися один з кутів багатокутника. Задайте точки зламу клацанням лівою клавішею миші там, де будуть розташовуватися інші кути багатокутника. Останній кут багатокутника слід відзначити клацанням правої клавіші мишки. Результат створення багатокутника наведено на рис. 6.36.



Рисунок 6.36 – Приклад створення багатокутника для заливки

Виділіть створений багатокутник. Якщо вікно властивостей не відкриться, двічі клацніть лівою клавiшею миші по створеному багатокутнику. Розкрийте розділ *Заливка* подвійним клацанням лівої кнопки миші, якщо він не розкритий. У полі *Стиль* виберіть *без заливки*. В результаті зображення прийме вигляд, зазначений на рис 6.33.


Перейдіть на закладку *Динамическая заливка* . Поставте прапорець *Разрешено*. Розкрийте розділ *Слой*. Для цього двічі клацніть лівою клавiшею миші на тексті *Слой*. У полі *Имя* введіть ім'я для шару. Проведіть прив'язку шару до аргументу, за допомогою якого проведена прив'язка тренда до каналу, що зберігає значення рівня продукту в ємності. В полі *Тип заливки* виберіть *Изображение*. Клацніть в полі зображення. Відкриється вікно, яке пропонує вибір зображень, що зберігаються в бібліотеці зображень, створеній раніше (рис. 6.37).



Рисунок 6.37 – Вибір зображення для заливки

Виберіть потрібне зображення і клацніть лівою клавiшею миші по кнопці *Готово*. В полі *Макс.* встановіть значення верхньої межі діапазону значень, що зберігаються в каналі, створеному раніше, а в поле *Мин.* – нижньої межі.

Проведіть запуск проекту, аналогічно першому прикладу. Приклад результату виконання створеного проекту наведено на рис. 6.38.

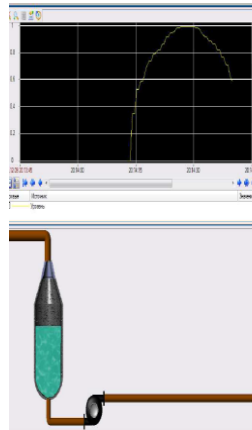


Рисунок 6.38 – Результат виконання проекту з динамічною заливкою

Приступимо до наступного кроку вивчення SCADA системи TRACE MODE 6.0. Розглянемо приклади застосування мов Techno ST і Techno FBD і реалізуємо систему АСК ТП з використання програмної обробки. Для цього поставимо такі завдання:

1. Взяти за основу проект, створений при виконанні другого прикладу (статичне та динамічне зображення), додати канали, які передають значення вартості продукту, витрати, сумарної витрати, сумарної вартості продукту, періоду генерації сигналу, видалити генератор, що використовувався. У всіх каналів слід зняти прапор **Использовать** на панелі обробка при редагуванні каналу.
2. Додати, кнопки, при натисканні на які вводиться вартість продукту, період генерації рівня продукту в ємності, витрата, об'єкти текст для відображення сумарної витрати продукту, вартості витраченого продукту, вартості продукту, періоду генерації рівня продукту в ємності, витрати і вартості.

3. Додати FBD діаграму, яка генерує рівень продукту в ємності. Якщо період генерації менше 10, то програма повинна встановити період рівний 10. Програма повинна повертати масштабований сигнал, тобто значення повинні змінюватися в заданому діапазоні.
4. Додати програму ST, яка перевіряє значення вартості та витрати і робить розрахунки. Якщо значення вартості або витрати менше 1, то встановлює відповідний параметр рівним 1. Розраховує сумарну витрату продукту та сумарну вартість витраченого продукту.

Розглянемо необхідні вимоги для вирішення поставлених

завдань:

1. У всіх каналів і у програм повинен бути встановлений при редагуванні однаковий період 1-3 секунди.
2. Обчислювати сумарну витрату слід за формулою:

$$\text{Сумарний_витрати} = \text{Попередній_витрати} + \text{Витрати} \cdot \text{Період_обробки}$$

де:

Сумарний_витрати – аргумент програми, повертає значення сумарної витрати вхідному значенню каналу, який передає значення сумарної витрати;




Попередній_витрати – глобальна змінна, що містить значення сумарної витрати, обчислене при попередньому виклику програми;

Період_обробки – аргумент, прив'язаний до аргументу період перерахунку каналу, який передає значення витрати.

3. Після обчислення сумарної витрати необхідно глобальній змінній *Попередній_витрати* привласнити обчислене значення *Сумарний_витрати*, сумарну вартість продукту обчислити як добуток обчисленого значення сумарної витрати на вартість продукту.

Для реалізації даного проекту перезберегти проект виконаний у другому прикладі під новим ім'ям. Виділіть об'єкт

Екран і викличте контекстне меню. Виберіть **Редактировать**. У вікні редагування каналу виберіть **Период** рівним 1, одиницю виміру – сек. Аналогічно змініть період перерахунку каналу, який зберігає рівень продукту в ємності. Додайте канали, що зберігають значення вартості продукту, витрати продукту, періоду генерації. Для кращого сприйняття назвемо канал, який зберігає вартість продукту як вартість, який зберігає витрату продукту – витрата, період генерації – період генерації. Встановіть тип кожного каналу – Input (див. Приклад 1). Додайте канали, що зберігають результати обчислень: сумарну витрату продукту, сумарну вартість продукту. Для зручності сприйняття назвемо канал, який зберігає сумарну вартість продукту, як сумарна вартість, а сумарну витрату продукту - сумарна витрата. Встановіть тип зазначених каналів – **Input**. Для всіх каналів встановимо однаковий період перерахунку значення. Для цього викличте вікно редагування для кожного каналу. В полі **Период** встановіть необхідний період перерахунку 1 і одиницю виміру – секунду (сек).

Теперь створимо **Кнопки для задания необходимых параметров**. Клацніть лівою кнопкою миші по іконці . Якщо даної іконки нема на панелі інструментів, то клацніть правою клавішею миші по одній з іконок:  або . Розмістіть кнопки на екрані. Для цього задайте протилежні кути кнопок клацанням лівої кнопки миші (рис. 6.39).

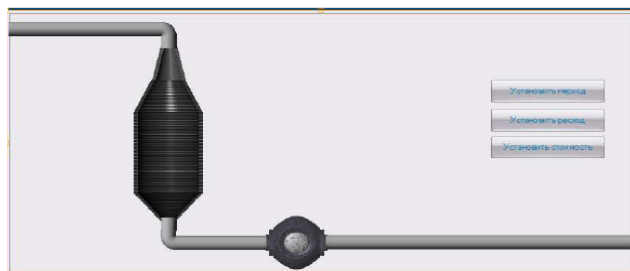



Рисунок 6.39 – Приклад розміщення кнопок

Для кожної кнопки відкрийте вікно властивостей. В полі **Текст** введіть призначення кнопки. Так, для кнопки, яка встановлює період генерації можна ввести «Встановити період», яка встановлює витрату продукту – «Встановити витрату», яка встановлює вартість продукту – «Встановити вартість». Перейдіть на закладку **Действия** . Виділіть рядок **mousePressed** і викличте контекстне меню (рис. 6.40).

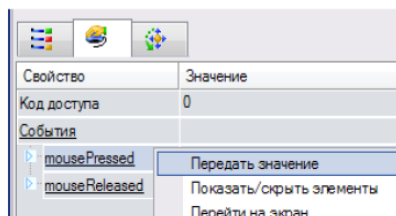


Рисунок 6.40 – Створення події

Виберіть в контекстному меню рядок **Передать значение**. У полі **Тип передачи** встановіть **Вести и передать**. В полі **Результат** зробіть прив'язку до вхідного значення каналу, якому передаватиметься значення, що вводиться (приклад налаштування кнопки – рис. 6.41).

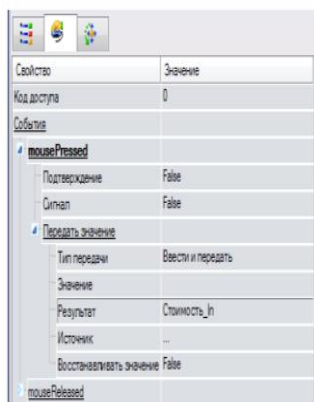


Рисунок 6.41 – Вікно налаштування події

В результаті прив'язки будуть створені чотири аргументи в таблиці вікна властивостей прив'язки (рис. 6.42).

Имя	Тип	Тип данных	Привязка	Фикс
Уровень_R	IN	REAL	Уровень:Реальное значение (Система.RTM.Каналы)	
Период_генерации_лн	OUT	REAL	Период генерации:Входное значение (Система.RTM.Каналы)	
Расход_лн	OUT	REAL	Расход:Входное значение (Система.RTM.Каналы)	
Стоимость_лн	OUT	REAL	Стоимость:Входное значение (Система.RTM.Каналы)	

Рисунок 6.42 – Атрибути для введення даних

Для виведення значень, що зберігаються в ряді каналів, створить об'єкти **Текст**. Приклад створення об'єктів **Текст** зображено на рис. 6.43.

Период генерации, с	<текст>	Установить период
Расход, л/с	<текст>	Установить расход
Стоимость руб/л	<текст>	Установить стоимость
Суммарный расход, л	<текст>	
Суммарная стоимость, руб	<текст>	

Рисунок 6.43 – Створення об'єктів **Текст**

Об'єкти **Текст**, розташовані зліва, виводять підказку (де який параметр виведений), а праворуч – значення, що зберігають канали. Прив'язку об'єктів **Текст**, розташованих в правій колонці, виконуєте до реальних значень відповідних каналів. Для прив'язки, при необхідності, створить додаткові аргументи, як показано на рис. 6.44.

Имя	Тип	Тип данных	Привязка
Уровень_R	IN	REAL	Уровень:Реальное значение (Система.RTM.Каналы)
Период_генерации_лн	OUT	REAL	Период генерации:Входное значение (Система.RTM.Каналы)
Расход_лн	OUT	REAL	Расход:Входное значение (Система.RTM.Каналы)
Стоимость_лн	OUT	REAL	Стоимость:Входное значение (Система.RTM.Каналы)
Период_генерации_р	IN	REAL	Период генерации:Реальное значение (Система.RTM.Каналы)
Расход_р	IN	REAL	Расход:Реальное значение (Система.RTM.Каналы)
Стоимость_р	IN	REAL	Стоимость:Реальное значение (Система.RTM.Каналы)
Суммарный_расход_р	IN	REAL	Суммарный расход:Реальное значение (Система.RTM.Каналы)
Суммарная_стоимость_р	IN	REAL	Суммарная стоимость:Реальное значение (Система.RTM.Каналы)

Рисунок 6.44 – Атрибути екрана

Виконаємо розрахунок множника K та зміщення C . Масштабування сигналу в FBD програмі буде виконуватися за формулою:

$$Y = K \cdot X + C,$$

де

- Y – результат перетворення;
- X – вхідне значення каналу;
- K – множник;
- C – зміщення.

FBD блоки, що генерують сигнал повертають сигнал, що змінюється в діапазоні $[-1;+1]$. Знайдіть значення множника і зміщення, які дозволять отримати необхідний діапазон зміни результату перетворення (діапазон сигналу). Як приклад: необхідно забезпечити діапазон сигналу $[0;+1]$. Множник K складе $0,5$, зміщення – $Z - 0,5$.

Створимо FBD діаграми, для цього виділіть групу **Канали** RTM вузла і викличте контекстне меню. В меню оберіть **Создать компонент**. Серед запропонованих варіантів виберіть **Программа** (рис. 4.49).

Виділивши створену програму і, клацнувши лівою клавішею миші, можете змінити ім'я програми, наприклад, на «синусоїда». Виділіть створений об'єкт-програму і викличте контекстне меню. В меню оберіть **Редактировать**. Відкриється вікно, аналогічне редагування каналу **float**. В полі **Период** виберіть **1**, виберіть одиницю вимірювання – секунда (сек). Після налаштування періоду перерахунку вікно редагування каналу можна закрити.

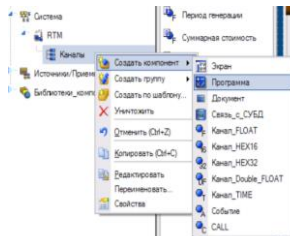


Рисунок 6.45 – Створення об'єкта програми

Подвійним клацанням лівої кнопки миші по створеній програмі або викликавши контекстне меню і вибравши рядок **Редактировать шаблон**, викличте вікно для створення (або редагування) програми. Відкриється вікно, представлене на рис. 6.46.

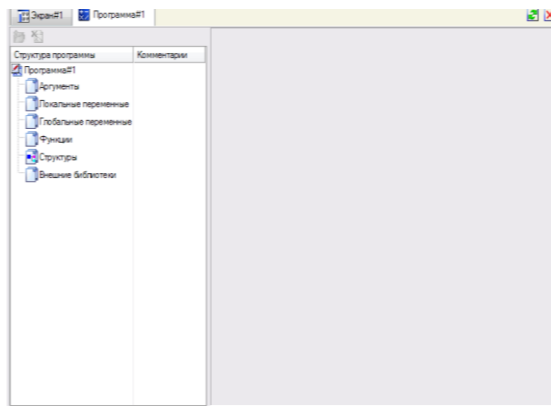


Рисунок 6.46 – Вікно редагування шаблону програми

В структурі програми виділіть рядок **Аргументы**. Заповніть інформацію таблиці, що з'явилася. Кожний рядок – аргумент програми. Встановіть тип In для аргументів, які будуть передавати дані в програму, Out – які будуть передавати дані з програми, In / Out – які будуть передавати в програму і з неї. Приклад створення аргументів програми для генерування синусоїдального сигналу наведено на рис. 6.47.

Имя	Тип	Тип данных	Привязка
Период_генерации_R	↑ IN/OUT	REAL	Период генерации:Реальное значение (Система.RTM.Каналы)
Уровень_In	↑ OUT	REAL	Уровень:Входное значение (Система.RTM.Каналы)

Рисунок 6.47 – Атрибути програми Techno FBD

Виділіть рядок **Программа#** в структурі програми (рис.6.46). Серед запропонованих мов програмування оберіть FBD діаграму (рис. 6.47).

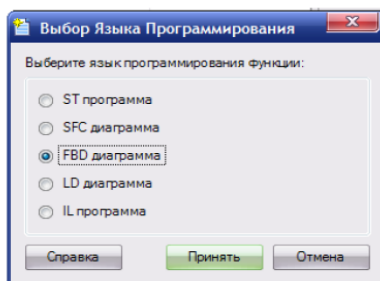



Рисунок 6.47 – Вибір мови

Клацніть лівою клавiшею миші по іконці  або оберіть рядок **Палитра FBD блоків** в меню **Вид**, якщо палітра FBD блоків (рис. 6.48) не відкрита.

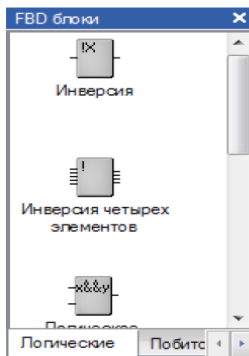


Рисунок 6.48 – Вікно вибору FBD блоків

Перейдіть на закладку **Сравнение**. Натисніть ліву клавiшу миші на блоці **Больше или равно**. Перемістіть його на робоче поле й отпустіть клавiшу миші. Аналогічно розмістіть на робочому полі FBD блоки (рис. 6.49): **Выбор из двух** (зкладка

сообщения. При відсутності помилок буде написано: «Програма # 1.tms compiled successfully». При наявності помилки буде написано: «E0011 Синтаксична помилка».

Приступимо до розгляду створення ST програми. Створіть об'єкт **Програма** в групі **Канали** RTM вузла. Назву програми можна змінити на «обробка». Встановіть період перерахунку рівним 1 секунд. Відкрийте програму подвійним клацанням лівої кнопки миші по створеному об'єкту програми. У структурі програми виберіть аргументи. Створіть аргументи, які будуть передавати в програму витрату продукту, вартість продукту, період перерахунку каналів, а також аргументи, які будуть повертати з програми сумарну витрату продукту і сумарну вартість всього витраченого продукту (рис. 6.50).

Имя	Тип	Тип данных	Привязка
Расход	IN/OUT	REAL	Расход: Реальное значение (Система RTM. Каналы)
Стоимость	IN/OUT	REAL	Стоимость: Реальное значение (Система RTM. Каналы)
T	TN	REAL	Уровень: Период пересчета (значение) (Система RTM. Каналы)
Суммарный_расход	OUT	REAL	Суммарный расход: Входное значение (Система RTM. Каналы)
Суммарная_стоимость	OUT	REAL	Суммарная стоимость: Входное значение (Система RTM. Каналы)

Рисунок 6.50 – Аргументи програми на мові Techno ST

Оберіть рядок **Глобальные переменные**. Аналогічно створенню аргументів створіть глобальну змінну, яка буде зберігати значення сумарної витрати, обчислене при попередньому виклику програми, вказавши початкове значення рівним нулю (рис. 6.51).

Имя	Тип данных
Предыдущий_суммарный_расход	REAL

Рисунок 6.51 – Створення глобальної змінної

Виділіть рядок **Программа#**. У вікні, що відкрилося, оберіть ST програму. При вказаних вище іменах аргументів і глобальної змінної текст програми набуде вигляду, вказаного на рис. 6.52.

```


Документ_1_in
PROGRAM
VAR_INOUT Расход : REAL; END_VAR
VAR_INOUT Стоимость : REAL; END_VAR
VAR_INPUT T : REAL; END_VAR
VAR_OUTPUT Суммарный_расход : REAL; END_VAR
VAR_OUTPUT Суммарная_стоимость : REAL; END_VAR

If Расход < 1 Then
    Расход = 1;
End_if;
If Стоимость < 1 Then
    Стоимость = 1;
End_if;
Суммарный_расход = Предыдущий_суммарный_расход + Расход * T;
Предыдущий_суммарный_расход = Суммарный_расход;
Суммарная_стоимость = Суммарный_расход * Стоимость;

END_PROGRAM

```

Рисунок 6.52 – Програма на мові Techno ST

Перевірте програму, клацнувши лівою клавiшею миші по іконці . Запустіть проект, аналогічно попереднім прикладам. Результат запуску наведено на рис. 6.53.

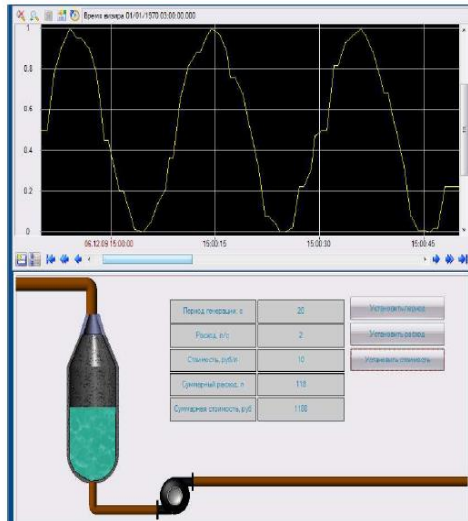


Рисунок 6.53 – Результат запуску проекту

6.4. Програмування на мовах Techno IL і Techno SFC

Мета даного підрозділу вивчити мови програмування Techno IL і Techno SFC на базі попередніх прикладів і створити АСК ТП.

Для вивчення візьмемо АСК ТП, створену в попередньому підрозділі, видалимо обидві програми написаних на мові Techno ST і Techno FBD, додамо вимикач на екрані, створимо канал, який визначає необхідність визначення сумарної витрати і сумарної вартості продукту та до якого виконується прив'язка вимикача.

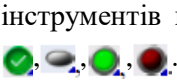
Розглянемо основні кроки, які необхідно виконати для реалізації поставленої мети:

- перший крок: виконання перевірок значення періоду дискретизації та значення витрати, якщо період дискретизації виявиться менше 10, то встановить значення за замовчуванням (10), якщо витрата виявляється менше 1, то встановить відповідний параметр рівним 1, змінити значення, програма повинна повертати відповідним каналам;
- другий крок: генерування рівня в ємності на мові Techno FBD (він перевіряється на першому кроці); в якості умови переходу слід записати константу true;
- третій крок: виконується, коли вимикач повертає 1, реалізований, з використанням мови Techno IL: обчислення сумарної витрати, сумарної вартості з використанням глобальної змінної для зберігання попереднього значення сумарної витрати, обчислені значення сумарної витрати, сумарної вартості програма повинна повертати відповідним каналам;
- четвертий крок: виконується, коли вимикач повертає 0, реалізований за допомогою мови Techno IL: обчислюється сумарна витрата продукту, результат обчислення не повертається програмою відповідного каналу, а тільки

присвоюється глобальної змінної, призначеної для зберігання попереднього значення сумарного витрата.

Як обмеження поставимо значення періоду перерахунку у всіх каналів програми повинен один і той же в межах до 3 секунд.

Приступимо до реалізації вирішення поставленого завдання в заданих обмеженнях. На першому кроці збережіть проект, створений в ході виконання попереднього прикладу під новим ім'ям і видаліть обидві програми.

Створіть ще один канал. Встановіть період перерахунку рівним 1 секундi. Для зручності сприйняття назвемо новий канал «вимикач». Для створення вимикача необхідно клацнути лівою клавiшею миші по іконці вимикача на робочому столі. На панелі інструментів може бути одна з наступних іконок: . Всі наведені перемикачі мають однакові можливості. Тому не важливо, який перемикач буде розташований на екрані. Досить вибрати той, який розміщений на панелі інструментів. Після вибору інструменту для створення вимикача клацніть лівою клавiшею миші там, де хочете помістити вимикач. Приклад розміщення вимикача наведено на рис. 6.54.

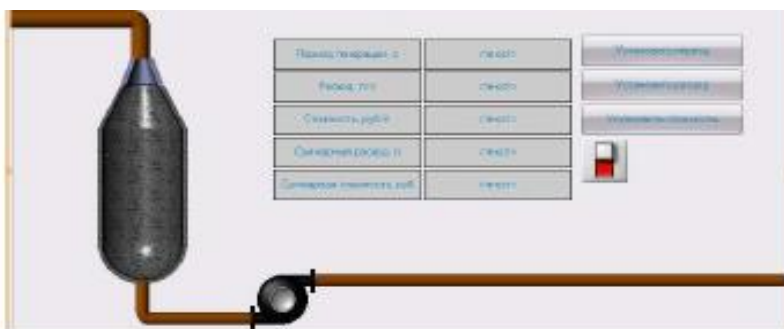


Рисунок 6.54 – Приклад розміщення вимикача

Відкрийте вікно властивостей об'єкта «вимикач». Проведіть налаштування, як показано на рис. 6.55.

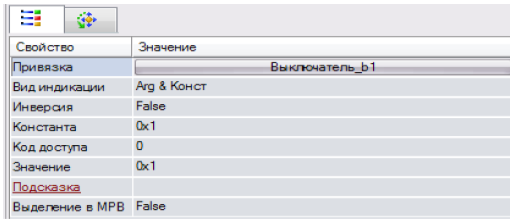


Рисунок 6.55 – Вікно налаштування вимикача

Вимикач_b1 – аргумент для прив'язки до першого біта каналу вимикача (рис. 6.56).

Имя	Тип	Тип данных	Привязка
Уровень_R	IN	REAL	Уровень:Реальное значение (Система.RTM.Каналы)
Период_генерации_In	OUT	REAL	Период генерации:Входное значение (Система.RTM.Каналы)
Расход_In	OUT	REAL	Расход:Входное значение (Система.RTM.Каналы)
Стоимость_In	OUT	REAL	Стоимость:Входное значение (Система.RTM.Каналы)
Период_генерации_R	IN	REAL	Период генерации:Реальное значение (Система.RTM.Каналы)
Расход_R	IN	REAL	Расход:Реальное значение (Система.RTM.Каналы)
Стоимость_H	IN	REAL	Стоимость:Реальное значение (Система.HIM.Каналы)
Суммарный_расход_R	IN	REAL	Суммарный расход:Реальное значение (Система.RTM.Каналы)
Суммарная_стоимость_R	IN	REAL	Суммарная стоимость:Реальное значение (Система.RTM.Каналы)
Выключатель_b1	IN/OUT	BOOL	Выключатель:Бит 1 (Система.RTM.Каналы)

Рисунок 6.56 – Аргумента екрана

Приступимо до створення SFC діаграми. Створіть об'єкт **Программа**. Створіть атрибути, аналогічно до попереднього прикладу. Крім атрибутів, що використовувалися в попередньому прикладі, знадобиться атрибут, що передає в програму реальне значення каналу, який зберігає стан вимикача. Приклад створення атрибутів наведено на рис. 6.57.


Имя	Тип	Тип данных	Привязка
Уровень	OUT	REAL	Уровень:Входное значение (Система. RTM. Каналы)
Период_генерации	IN/OUT	REAL	Период генерации:Реальное значение (Система. RTM. Каналы)
Расход	IN/OUT	REAL	Расход:Реальное значение (Система. RTM. Каналы)
Стоимость	IN/OUT	REAL	Стоимость:Реальное значение (Система. RTM. Каналы)
Период_пересчета	IN	REAL	Уровень:Период пересчета (значение) (Система. RTM. Каналы)
Суммарный_расход	OUT	REAL	Суммарный расход:Входное значение (Система. RTM. Каналы)
Суммарная_стоимость	OUT	REAL	Суммарная стоимость:Входное значение (Система. RTM. Каналы)
Выключатель	IN	BOOL	Выключатель:Бит 1 (Система. RTM. Каналы)

Рисунок 6.57 – Аргументы программы

Створіть глобальну змінну, для зберігання результату обчислення сумарної витрати при попередньому виклику програми (рис. 6.54). Виділіть рядок **Программа#** в структурі програми. Серед запропонованих мов програмування виберіть SFC діаграму. На робочому полі відображається тільки один крок: **Начальный шаг** (рис. 6.58).



Рисунок 6.58 – Початковий крок програми на мові Techno SFC

Двічі клацніть по даному кроці лівою клавiшею миші. Введіть нове ім'я кроку, наприклад, «перевірка вихідних даних». Для створення нового кроку виділіть єдиний існуючий крок. Клацніть лівою клавiшею миші по іконці  або викличте контекстне меню і виберіть створити **Создать шаг/переход**. Буде створено новий крок і перехід. Подвійним клацанням по створеному кроці і переходу змініть їх ім'я. Крок можна назвати як «генерування сигналу», наприклад, перехід – «true». Виділіть крок «генерування сигналу» і створіть новий крок та перехід. Створений крок і перехід можна назвати як «повний розрахунок» та «реальне значення вимикача». Виділіть крок «генерування сигналу». Створіть новий крок і перехід. Новий крок і перехід можна назвати як «розрахунок тільки за

попереднє значення сумарного значення» і «реальне значення вимикача з запереченням». Зовнішній вигляд створеної SFC діаграми наведено на рис. 6.59.



Рисунок 6.59 – Програма на мові SFC

Приступимо до програмування і реалізуємо це в наступні кроки:

а) Створення кроку «**Проверка исходных данных**». У структурі програми розкрийте розділ *SFC діаграма*. Розкрийте розділ *Шаги* и *переходы*. Виділіть рядок *Проверка исходных данных*. Серед запропонованих мов виберіть TechnoIL. При зазначених вище іменах атрибутів і глобальній змінній текст програми, що виконує перевірку вихідних даних набуде вигляду:

```

SFC_STEP «Проверка исходных данных»
VAR_OUTPUT Уровень: REAL; END_VAR
VAR_INOUT Период_генерации: REAL; END_VAR
VAR_INOUT Расход: REAL; END_VAR
VAR_INOUT Стоимость: REAL; END_VAR
VAR_INPUT Период_пересчета: REAL; END_VAR
VAR_OUTPUT Суммарный_расход: REAL; END_VAR
VAR_OUTPUT Суммарная_стоимость: REAL; END_VAR
VAR_INPUT Выключатель: REAL; END_VAR
GE Расход I//проверка расхода
JMPC Проверка_стоимости//переход к проверке
стоимости при расходе > I
  
```

LD 1
 ST Расход//Расход по умолчанию
 Проверка_стоимости: GE Стоимость 1
 JMPC Проверка_периода//переход к проверке периода
 генерации, когда стоимость > 1

LD 1
 ST Стоимость//Стоимость по умолчанию
 Проверка_периода: GE Период_генерации 10
 JMPC конец//выход при периоде > 10
 LD 10
 ST Период_генерации//Период генерации по умолчанию
 конец:

END_SFC_STEP.

б) Створення кроку «Генерирование сигнала». Виділіть рядок **Генерирование сигнала**. Серед запропонованих мов оберіть **FBD**. Створіть FBD програму, як показано на рис. 6.60

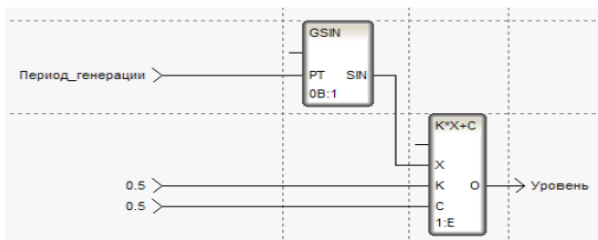


Рисунок 6.60 –Крок «Генерирование сигнала»

Для створення переходу до створеного кроку виділіть рядок **true** в розділі **Переходы**. Серед запропонованих мов оберіть **ST**. Текст програми набуде вигляду:

```
SFC_TRANSITION «true» FROM(INITIAL_STEP) TO(STEP_1)
VAR_OUTPUT Уровень: REAL; END_VAR
VAR_INOUT Период_генерации: REAL; END_VAR
VAR_INOUT Расход: REAL; END_VAR
VAR_INOUT Стоимость: REAL; END_VAR
VAR_INPUT Период_пересчета: REAL; END_VAR
VAR_OUTPUT Суммарный_расход: REAL; END_VAR
```

VAR_OUTPUT Суммарная_стоимость: REAL; END_VAR
 VAR_INPUT Выключатель: REAL; END_VAR
 true
 END_SFC_TRANSITION

е) Створення кроку **«Полный расчет»**. Оберіть рядок **Полный расчет**. Серед запропонованих мов оберіть **IL**. Текст програми набуде вигляду:

SFC_STEP «Полный расчет»
 VAR_OUTPUT Уровень: REAL; END_VAR
 VAR_INOUT Период_генерации: REAL; END_VAR
 VAR_INOUT Расход: REAL; END_VAR
 VAR_INOUT Стоимость: REAL; END_VAR
 VAR_INPUT Период_пересчета: REAL; END_VAR
 VAR_OUTPUT Суммарный_расход: REAL; END_VAR
 VAR_OUTPUT Суммарная_стоимость: REAL; END_VAR
 VAR_INPUT Выключатель: REAL; END_VAR
 MUL Период_пересчета Расход//вычисление расхода за период пересчета
 ADD Предыдущий_суммарный_расход//Нахождение суммарного расхода
 ST Предыдущий_суммарный_расход//присвоение вычисленного значения суммарного расчета
 ST Суммарный_расход
 MUL Стоимость//вычисление суммарной стоимости
 ST Суммарная_стоимость//присвоение аргументу результата вычисления суммарной стоимости
 END_SFC_STEP

Для створення переходу до створеного кроку виділіть рядок **Реальное значение выключателя**. Серед запропонованих мов оберіть **ST**. Текст програми набуде вигляду:

SFC_TRANSITION «Реальное значение выключателя»
 FROM(STEP_1) TO(STEP_2)
 VAR_OUTPUT Уровень: REAL; END_VAR
 VAR_INOUT Период_генерации: REAL; END_VAR
 VAR_INOUT Расход: REAL; END_VAR

VAR_INOUT Стоимость: REAL; END_VAR
 VAR_INPUT Период_пересчета: REAL; END_VAR
 VAR_OUTPUT Суммарный_расход: REAL; END_VAR
 VAR_OUTPUT Суммарная_стоимость: REAL; END_VAR
 VAR_INPUT Выключатель: BOOL; END_VAR
 Выключатель
 END_SFC_TRANSITION

д) Створення кроку «**Расчет только предыдущего значения суммарного расчета**». Виділіть рядок **Расчет только предыдущего значения суммарного расчета**. Серед запропонованих мов оберіть **IL**. Текст програми набуде вигляду:

SFC_STEP «**Расчет только предыдущего значения суммарного расхода**»

VAR_OUTPUT Уровень: REAL; END_VAR
 VAR_INOUT Период_генерации: REAL; END_VAR
 VAR_INOUT Расход: REAL; END_VAR
 VAR_INOUT Стоимость: REAL; END_VAR
 VAR_INPUT Период_пересчета: REAL; END_VAR
 VAR_OUTPUT Суммарный_расход: REAL; END_VAR
 VAR_OUTPUT Суммарная_стоимость: REAL; END_VAR
 VAR_INPUT Выключатель: REAL; END_VAR

MUL Период_пересчета Расход//вычисление расхода за период пересчета

ADD Предыдущий_суммарный_расход // Нахождение суммарного расхода

END_SFC_STEP

Для створення переходу до створеного кроку виділіть рядок **Реальное значение выключателя с отрицанием**. Серед запропонованих мов оберіть **ST**. Текст програми набуде вигляду:

SFC_TRANSITION «**Реальное значение выключателя с отрицанием**» *FROM(STEP_1) TO(STEP_3)*

VAR_OUTPUT Уровень: REAL; END_VAR
 VAR_INOUT Период_генерации: REAL; END_VAR
 VAR_INOUT Расход: REAL; END_VAR

VAR_INOUT Стоимость: REAL; END_VAR
VAR_INPUT Период_пересчета: REAL; END_VAR
VAR_OUTPUT Суммарный_расход: REAL; END_VAR
VAR_OUTPUT Суммарная_стоимость: REAL; END_VAR
VAR_INPUT Выключатель: BOOL; END_VAR
!Выключатель
END_SFC_TRANSITION

Проведем запуск проекту. При обох положеннях перемикача має відбуватися заповнення ємності. При одному з положень перемикача повинен проводитися виведення нових отриманих значень сумарної витрати та вартості, при іншому – нові значення сумарної витрати та вартості не виводяться. Приклад результату роботи представлений на рис. 6.61.

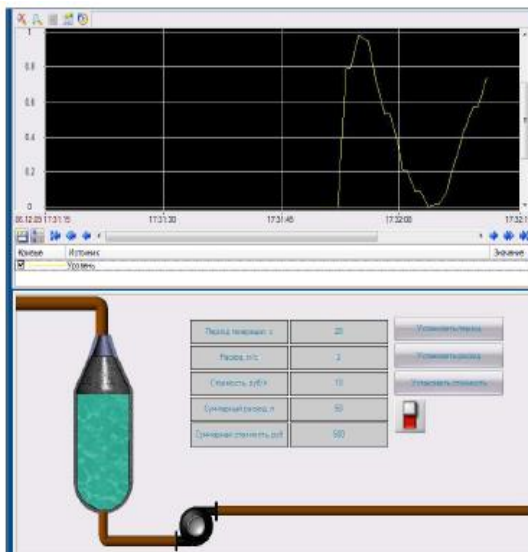


Рисунок 6.61 – Приклад використання програми

6.5. Створення звіту тривоги та СПАД архіву

Наступним кроком розглянемо і вивчимо створення звітів тривоги і СПАД архіву. Для реалізації цього поставимо такі завдання:

- а) взяти за основу останній приклад (програмування в середі TRACE MODE на мовах Techno IL і Techno SFC);
- б) замінити тренд на архівний тренд;
- в) налаштувати аналіз меж каналу, що передає рівень продукту в ємності;
- г) створити словник повідомлень для каналу, що передає рівень продукту в ємності, налаштувати вузол для створення звіту тривоги;
- д) змінити динамічну заливку ємності на заливку кольором і налаштувати вибір кольору відповідно до приналежності значення рівня тому чи іншому діапазону;
- е) налаштувати СПАД архівування каналу, що зберігає рівень продукту.

На першому кроці скористаємося попереднім прикладом. Збережіть проект, під новим ім'ям. Видалимо тренд. Додайте на екрані архівний тренд, який налаштуйте на виведення рівня продукту в ємності. Бажано змінити період коливань за замовчуванням з 10 на 40. Крок, що виконує перевірку вихідних даних набуде вигляду:

```
SFC_STEP «Проверка исходных данных»  
VAR_OUTPUT Уровень: REAL; END_VAR  
VAR_INOUT Период_генерации: REAL; END_VAR  
VAR_INOUT Расход: REAL; END_VAR  
VAR_INOUT Стоимость: REAL; END_VAR  
VAR_INPUT Период_пересчета: REAL; END_VAR  
VAR_OUTPUT Суммарный_расход: REAL; END_VAR  
VAR_OUTPUT Суммарная_стоимость: REAL; END_VAR  
VAR_INPUT Выключатель: BOOL; END_VAR  
GE Расход 1  
JMPC Проверка_стоимости
```

LD 1
ST Расход
Проверка_стоимости: GE Стоимость 1
JMPC Проверка_периода
LD 1
ST Стоимость
Проверка_периода: GE Период_генерации 10
JMPC Конец
LD 40
ST Период_генерации
Конец:
END_SFC_STEP

Виконаємо **Настройку контроля границ**. Виділіть **Уровень** в групі **Каналы** RTM вузла та викличте контекстне меню. Оберіть рядок **Редактировать** в меню, що з'явилося. На панелі **Границы** поставте прапор **Использовать**, встановіть межі діапазонів, як наведено на рис. 6.62. Встановіть прапор контроль меж.



Рисунок 6.62 – Встановлення меж каналу

Створимо **Словарь сообщений**. Для цього необхідно виділити вузол **RTM** і викличете контекстне меню (рис. 6.63).

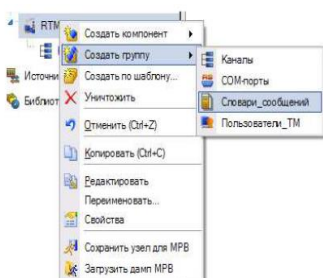


Рисунок 6.63 – Створення групи словників повідомлень

Оберіть рядок **Создать группу**. Серед запропонованих груп оберіть рядок **Словари_сообщений**. Виділіть створену групу **Словари_сообщений** і викличете контекстне меню. Серед запропонованих словників оберіть **Словарь для_FLOAT** (рис. 6.64).

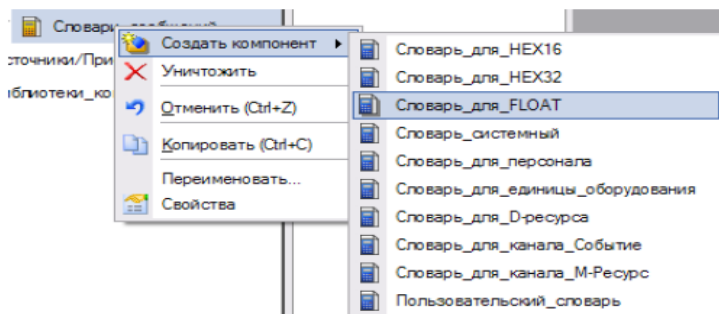


Рисунок 6.64 – Створення словників повідомлень

Зробимо редагування словника. Подвійним клацанням по створеному словнику повідомлень або виділивши його, викликавши контекстне меню і вибравши **Редактировать** відкриєте вікно для редагування словника. В полі **Имя** введіть

ім'я словника. Подвійним клацанням лівої кнопки миші на кожному рядку відкрийте вікно для редагування повідомлення (рис. 6.65).

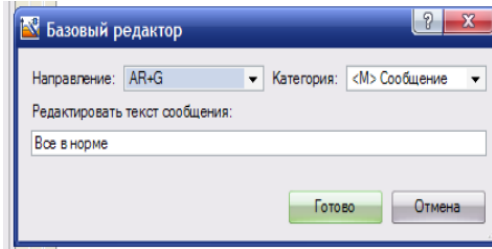


Рисунок 6.65 – Вікно для введення повідомлення

Редагуючи всі повідомлення, налаштуйте словник повідомлень, як показано на рис. 6.66

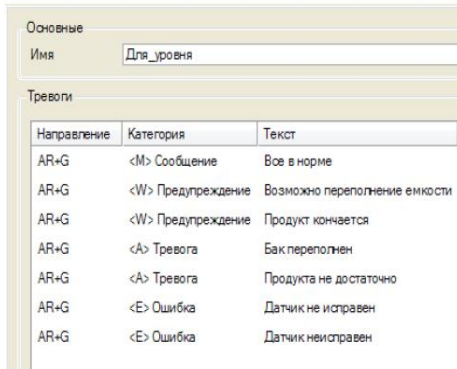


Рисунок 6.66 – Вікно редактора повідомлень

Редагування вузла. Виділіть вузол **RTM**. Викличте контекстне меню та оберіть **Редактировать**. Перейдіть на закладку **Отчет тревог/дамп/параметры**. Виконайте заповнення полів, як показано на рис. 6.67.

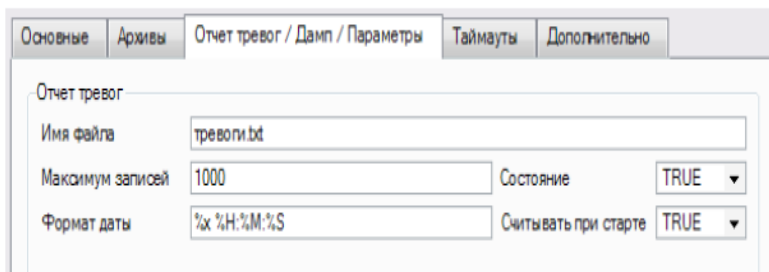


Рисунок 6.67 – Вікно редагування вузла

Перейдіть на закладку **Архивы**. Виконайте заповнення полів, як показано на рис. 6.68.

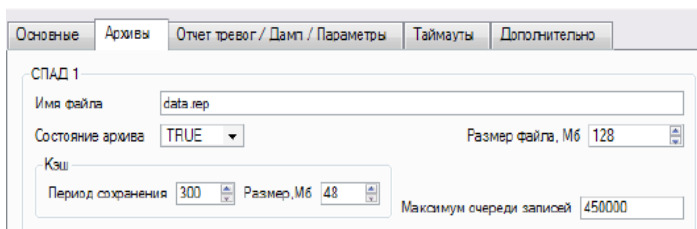


Рисунок 6.68 – Вікно заповнення полів у СПАД архіві

Наступним кроком проведемо редагування каналу, для цього в вікні **Редактирование канала уровень** задайте межі, на закладці **Архивация** налаштуйте архівацію і створення звіту тривог, як показано на рис. 6.69.

Налаштування заливки ємності. Відкрийте вікно властивостей об'єкта багатокутник (елемент зображення, що забезпечує динамічну заливку). Для отримання залежності кольору заливки від стану контрольованого процесу відрегулюйте як показано на рис. 6.70.

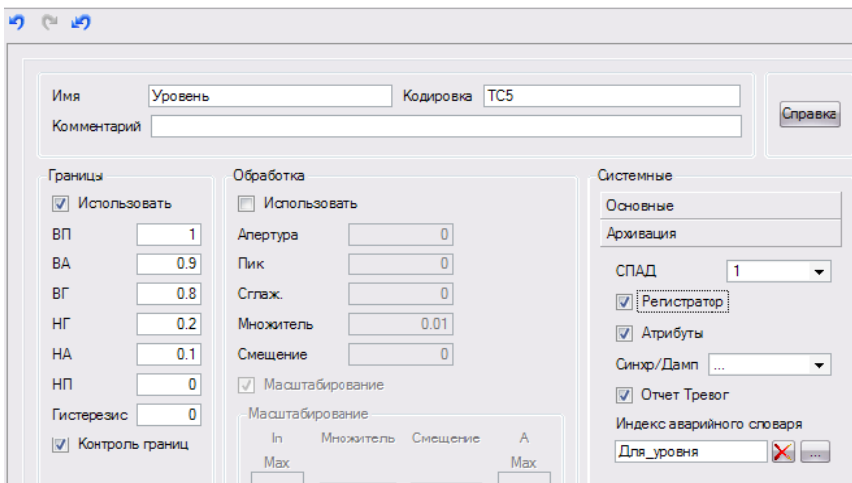


Рисунок 6.69 – Вікно редагування каналу

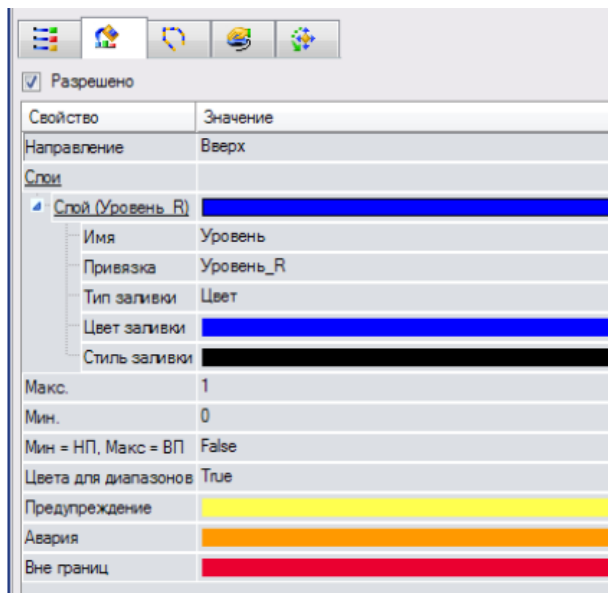


Рисунок 6.70 – Вікно налаштування зміни кольору динамічної заливки

Результат запуску проекту на виконання буде відрізнятися тим, що буде виводитися результат попередніх запусків проекту на виконання, періодично буде відбуватися оновлення тренда, який буде залежність на основі даних архіву. Колір заливки ємності буде визначатися рівнем продукту в ємності (рис. 6.71). Подивіться створений звіт тривоги, який розміщується в директорії з ім'ям вузла, розміщеної в директорії проекту

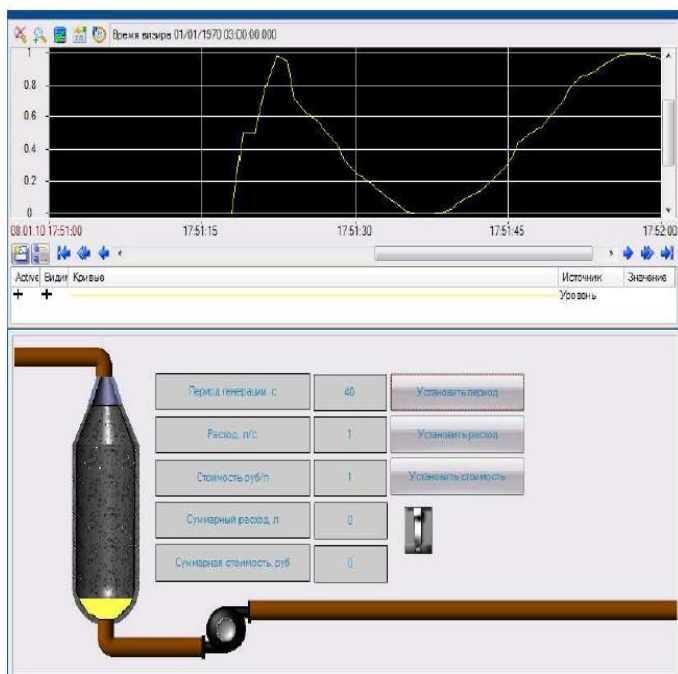


Рисунок 6.71 – Результат запуску програми

6.6. Контрольні питання

1. Для чого використовують бібліотеки системи TRACE MODE?
2. Як створити новий проект системи TRACE MODE?
3. Які генератори існують у системі TRACE MODE?
4. Як створити архівний тренд?
5. Послідовність створення статичних і динамічних об'єктів?

6.7. Контрольні завдання

Задача 1. Розробити комбіновану програму двохпозиційного регулювання температури води, поєднану з програмою емуляції роботи ємнісного термоелектричного нагрівача, використовуючи мову Function Block Diagram. Розробити програму сигналізації на мові Structured Text.

Об'єкт: ємність з водою з встановленим всередині термоелектричним нагрівачем, керованим за двохпозиційним законом (ввімкнено / вимкнено). Кількість води в баку вважати незмінною. Задана температура нагріву змінюється в межах [А]°С. Температуру води, що заливається в ємність, вважати рівною [Б]°С. Припустиме відхилення (гістерезис перемикання) становить ±[В]°С. Постійна часу нагрівача – [Г] хвилини. При відхиленні температури більш, ніж на [Д]°С від заданої, виконується індикація «НЕДОГРІВ» і «ПЕРЕГРІВ» відповідно.

Вимоги до інтерфейсу користувача: Графічний інтерфейс повинен містити засоби відображення поточної температури води в нагрівачі (цифровий індикатор), стану нагрівача (зміна кольору об'єкта) і засоби встановлення та зміни необхідної температури нагріву (повзунок). Також динаміка зміни температури води в баку повинна бути подана у вигляді тренда. Індикація «НЕДОГРІВ» і «ПЕРЕГРІВ» виконуються анімацією

типу «сигнальна лампа» жовтого та червоного кольорів відповідно.

№	Задана температура А, °С	Початкова температура Б, °С	Гістерезис В, °С	Постійна часу ОК Г, хв.	Допустиме відхилення Д, °С
1	30...98	12	2	2	4
2	20...90	10	2,5	1,5	5
3	15...95	10	3	1	5
4	15...90	14	1,5	0,5	3
5	20...92	11	2	0,8	4
6	30...90	14	1,5	1	3
7	25...90	8	2	1,2	5
8	40...90	20	3	2	5

Задача 2. Розробити комбіновану програму емуляції наливу води в циліндричну вертикальну ємність, поєднану з програмою трипозиційного регулювання рівня води, використовуючи мову Function Block Diagram. Розробити програму сигналізації на мові Structured Text.

Об'єкт: циліндрична ємність з водою висотою [А]м. Наповнення ємності виробляється насосом зі швидкістю [Б]см/с. Відбір води з ємності проводиться через трубопровід з встановленим на ньому соленоїдним двохпозиційним запірним клапаном з рівною швидкістю [В]см/с (гідростатичним тиском знехтувати). Оператором задається необхідне значення рівня води, а управління виконавчими пристроями виконується програмним забезпеченням автоматично, виходячи із заданих умов. Допустиме відхилення (зона нечутливості) становить [Г]см. При відхиленні рівня більш, ніж на [Д]см від заданого, виконується індикація «НЕДОЛИВ» і «ПЕРЕЛИВ» відповідно.

Вимоги до інтерфейсу користувача: Графічний інтерфейс повинен містити статичну мнемосхему (об'ємні або плоскі фігури) ємності, трубопроводів, клапана і насоса; засоби відображення поточного рівня води в ємності (цифровий

індикатор і динамічна заливка типу «оглядове вікно»), активного і неактивного стану насоса і клапана (зміна кольору об'єкта на мнемосхемі), засоби завдання і зміни необхідного значення рівня води (повзунок). Також динаміка зміни рівня води в ємності повинна бути представлена у вигляді тренда. Індикація «НЕДОЛИВ» і «ПЕРЕЛИВ» виконуються анімацією типу «сигнальна лампа» жовтого кольору.

№	Висота А, м	Швидкість наливу В, см/с	Швидкість зливу В, см/с	Зона нечутливості і Г, см	Допустиме відхилення Д, см
1	4,5	5	2,5	25	20
2	5	10	4	45	30
3	2,5	8	6	40	30
4	3	12	8	55	35
5	3,5	6	2	30	25
6	4	4	2	20	20
7	1,5	5	3	25	20

Задача 3. Розробити комбіновану програму ПД-регулювання температури нафти, поєднану з програмою емуляції нагріву нафти паром в кожухотрубчастому теплообміннику, використовуючи мову Function Block Diagram. Розробити програму сигналізації на мові Structured Text.

Об'єкт: в кожухотрубчастий теплообмінник надходить нафта з початковою температурою [А]°С. Для здійснення нагріву в теплообмінник також подається пар, витрата якого регулюється клапаном, і в номінальному режимі при витраті [Б]кг/год забезпечується нагрів нафти до [В]°С. Максимальна витрата пари складає [Г]кг/год. Постійну часу теплообмінника прийняти рівню [Д] хвилинам. При відхиленні температури більш, ніж на [Е]°С від заданої, виконується індикація «ВИСОКА ТЕМПЕРАТУРА» і «НИЗЬКА ТЕМПЕРАТУРА» відповідно.

Вимоги до інтерфейсу користувача: Графічний інтерфейс повинен містити статичну мнемосхему (об'ємні або плоскі фігури) теплообмінника, трубопроводів, клапана, а також засоби відображення поточної витрати пари (цифровий індикатор) і температури нафти на виході з теплообмінника (цифровий індикатор), засоби завдання і зміни необхідного значення температури нафти (повзунок), а також завдання припустимого порогу відхилення від завдання (цифровий індикатор з введенням значення). Динаміка зміни температури нафти в теплообміннику повинна бути представлена у вигляді тренда. Індикація «ВИСОКА ТЕМПЕРАТУРА» і «НИЗЬКА ТЕМПЕРАТУРА» виконуються анімацією типу «сигнальна лампа» червоного і жовтого кольорів відповідно.

№	Початкова температура аА, °С	Номинал. випрата париБ, кг/год	Номинал. температур аВ, °С	Максимальна випрата париГ, кг/год	Постійна часуОК Д,хв	Припустиме відхилення Е, °С
1	22	860	70	1040	2	5
2	26	900	74	1100	2,5	3
3	30	1020	68	1240	1,5	4
4	24	920	66	1100	1	5
5	28	980	58	1180	1,8	3
6	32	1040	62	1250	1,6	4
7	30	820	60	990	2	4

1. Провести серію віртуальних експериментів, змінюючи уставку технологічного параметра і настройки регулятора технологічного параметра.
2. Опрацювати результати експерименту: заповнити таблицю результатів експерименту, привести тренди (графіки зміни технологічного параметра).
3. Зробити висновки.

ПЕРЕЛІК ПОСИЛАНЬ

1. Современные технологии промышленной автоматизации: учебник /О. В. Шишов. Саранск: Изд-во Мордов. ун-та, 2007. – 273 с. ISBN 5-7103-1123-5.
2. A beginners guide to Ethernet 802.3. Engineer-to-Engineer Note EE-269. – Analog Devices, In. – 2005, 26 p.
3. Abe N., Yamanaka K. Smith predictor control and internal model control – a tutorial SICE 2003 Annual Conference. – vol. 2, 4-6 Aug. 2003, p.1383 – 1387.
4. Aguero J.C., Goodwin G.C. Virtual closed loop identification: a subspace approach. – 43rd IEEE Conference on Decision and Control, 2004, CDC, vol. 1, 14-17 Dec. 2004, p. 364 – 369 Vol.1.
5. Ahn S.H., Sim B.S., Chi D.Y., Park K.N., Lee C.Y., Kim Y.J. Smith predictor control for water pressure control system with time delay. – 30th Annual Conference of IEEE Industrial Electronics Society, IECON 2004, vol. 1, 2-6 Nov. 2004, p. 663 – 666.
6. Akhyar S., Omatu S. Self-tuning PID control by neural-networks. Proceedings of 1993 International Joint Conference on Neural Networks, 1993. IJCNN '93-Nagoya, vol. 3, 25-29 Oct. 1993, p. 2749 – 2752.
7. Akyildiz I.F., Wang X. A survey on wireless mesh networks. – IEEE Communications Magazine, Vol. 43, Issue 9, Sept. 2005, p. S23 – S30.
8. Alamouti S. M. A simple transmit diversity technique for wireless communications. – IEEE J. Select. Areas Commun., vol. 16, Oct. 1998. – p. 1451–1458.
9. Aleksy M., Korthaus A., Schader M. Implementing distributed systems with Java and CORBA. – Springer, 2005, 343 p.
10. Andy Swales. Open Modbus/TCP specification. – Release 1.0, 29, Schneider Electric, March 1999.

11. Ang K. H., Chong G., Li Y. PID control system analysis, design, and technology. *IEEE Trans. on Control Syst. Tech.*, vol.13, No. 4, July 2005, p. 559-576.
12. Angrisani L.; Schiano Lo Moriello, R.; D'Apuzzo, M. New proposal for uncertainty evaluation in indirect measurements. – *IEEE Transactions on Instrumentation and Measurement*, Vol. 55, No. 4, 2006. – p. 1059 – 1064.
13. ANSI/IEEE Std 802.11, 1999 Edition. Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. – IEEE-SA Standards Board, 12 June 2003. – 528 p.
14. Antonini , G.; Scogna, A.C.; Orlandi, A. Grounding, unbalancing and length effects on termination voltages of a twinax cable during bulk current injection. – *IEEE Transactions on Electromagnetic Compatibility*, vol. 46, Issue 2, May 2004, p. 302 – 308.
15. Apkarian P., Noll D. Nonsmooth H_{∞} synthesis. – *IEEE Transactions on Automatic Control*, vol. 51, Issue 1, Jan. 2006, p. 71 – 86.
16. Aratani T. PID control system fed back along with squares of control error and its derivative-optimum design of nonlinear PID controller. – *Proceedings of the 41st SICE Annual Conference SICE 2002*, vol. 4, 5-7 Aug. 2002, p. 2628 – 2631.
17. Asano M., Yamamoto T., Oki T., Kaneda M. A design of neural-net based predictive PID controllers. – *1999 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 4, 1999, p. 1113 – 1118.
18. Astrom K.J., Hagglund T.. *Advanced PID control*. – ISA – The Instrumentation, Systems, and Automation Society, 2006, 460 p.
19. Astrom K.J., Lee T.H., Tan K.K., Johansson K.H. Recent advances in relay feedback methods – A survey. – *IEEE International Conference on Systems, Man and Cybernetics*, 1995.

- 'Intelligent Systems for the 21st Century', vol. 3, 22-25 Oct 1995, p. 2616 – 2621.
20. Astrom K.J., Hang C.C., Lim B.C. A new Smith predictor for controlling a process with an integrator and long dead-time. – IEEE Transactions on Automatic Control, vol. 39, Issue 2, Feb 1994, p. 343 – 345.
 21. Astrom K.J.; Hang C.C.; Lim B.C. A new Smith predictor for controlling a process with an integrator and long dead-time. – IEEE Transactions on Automatic Control, vol. 39, Issue 2, 1994, p. 343 – 345.
 22. Azevedo G.P.; Oliveira A.L. Control centers with open architectures.- IEEE Computer Applications in Power, vol. 14, No. 4, Oct 2001, p. 27 – 32.
 23. Baba Y., Shigemasa T., Yukitomo M., Kojima F., Takahashi M., Sasamura E. Model-driven PID control system in single-loop controller. – SICE 2003 Annual Conference. vol. 1, 4-6 Aug. 2003, p.187 – 190.
 24. Baer W.C.; Lally, R.W. An open-standard smart sensor architecture and system for industrial automation. – IEEE Aerospace Conference Proceedings, 2000, vol. 6, 2000, p. 123 – 131.
 25. Bahill A. A simple adaptive Smith-predictor for controlling time-delay systems: A tutorial. – IEEE Control Systems Magazine, vol. 3, Issue 2, May 1983, p.16 – 22.
 26. Bahill A. A simple adaptive Smith-predictor for controlling time-delay systems: A tutorial. – Control Systems Magazine, IEEE, vol. 3, Issue 2, May 1983, p.16 – 22.
 27. Bailo C., Alderson G., Yen J. Requirements for Open, Modular Architecture Controllers for Applications in the Automotive Industry. – www.isa.org.
 28. Basso M.; Bucher, R.; Romagnoli, M.; Vassalli, M. Real-Time Control with Linux: A Web Services Approach. – 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 12-15 Dec. 2005, p. 2733 – 2738.

29. Benefits problems, and issues in open systems architectures. – IEEE Transactions on Power Systems, vol. 9, No. 1, Feb 1994, p. 191 – 197.
30. Bertocco M., Cappellazzo S., Flammini A., Parvis M. A multi-layer architecture for distributed data acquisition. Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference, 2002. IMTC/2002, vol. 2, 2002, p. 1261 – 1264.
31. Boiko I. Autotune identification via the locus of a perturbed relay system approach. – Proceedings of the American Control Conference, 2003. vol. 5, 4-6 June 2003, p. 4385 – 4390.
32. Bonastre A.; Capella J.V.; Ors R. A new generic architecture for the implementation of intelligent and distributed control systems. – IECON 02 – IEEE 2002 28th Annual Conference of the Industrial Electronics Society, vol. 3, 5-8 Nov. 2002, p. 1790 – 1795.
33. Bonivento A., Carloni L.P., Sangiovanni-Vincentelli A. Platform-Based Design of Wireless Sensor Networks for Industrial Applications. – Design, Automation and Test in Europe, DATE '06. Vol. 1, 6-10 March 2006, p. 1-6.
34. Branch M., Bradley B. Real-time web-based system monitoring. – IEEE Industry Applications Magazine, Vol. 13, Issue 2, March-April 2007, p.12 – 16.
35. Braun M.W., Rivera D.E., Stenman A., Foslien W., Hrenya C. Multi-level pseudo-random signal design and “model-on-demand” estimation applied to nonlinear identification of a RTP wafer reactor. – Proceedings of the 1999 American Control Conference, vol. 3, 1999, p. 1573 – 1577.
36. Brooks T. Wireless technology for industrial sensor and control networks. – Proceedings of the First ISA/IEEE Conference Sensor for Industry, 2001, p. 73 – 77.
37. Burleson J. Wiring and grounding to prevent power quality problems with industrial equipment. Textile, Fiber and Film Industry Technical Conference, 8-9 May, 1991, p. 5/1 – 5/6.

38. Burluson J. Wiring and grounding to prevent power quality problems with industrial equipment. Textile, Fiber and Film Industry Technical Conference, 8-9 May, 1991, p. 5/1 – 5/6.
39. Business Justification of open architecture control. White paper version 1.0. – Users Group, www.omac.org. – 23 p.
40. Cabling and connector pin assignment. – CiA DR-303-1, ver.1.0, 1999. – 26 p.
41. Cai W.-J., Fang Y., Wang Y.-G. Robust closed-loop process identification from step test. – Control, Automation, Robotics and Vision Conference, 2004. ICARCV, vol. 2, 6-9 Dec. 2004, p.907 – 912.
42. Calcev G., Gorez R. Iterative technique for PID controller tuning. – Proceedings of the 34th IEEE Conference on Decision and Control, 1995, vol. 3, 13-15 Dec 1995, p. 3209 – 3210.
43. CAN physical layer for industrial applications. – 1994, CiA, DS-102. – 13 p.
44. CAN application layer for industrial application. – CiA, DS201-207, ver. 1.1, 1996. – 182 p.
45. CAN specification 2.0, part A . <=«« a=«« style=«text-decoration: none; color: teal;»>-www.can-cia.org .
46. CAN specification 2.0, part B . – CAN in Automation, www.can-cia.org.
47. CAN open electronic data sheet specification for CANopen. – CiA draft standard 306, v. 1.3, Jan. 01 2005. – 30 p.
48. CANopen application layer and communication profile. – CiA draft standard 301, v. 4.02, Feb. 13, 2002. – 135 p.
49. CANopen interface and device profile for IEC 61131-3 programmable devices. – CiA draft standard 405, v. 2.0, 2002. – 41 p.
50. Cao J.-Y., Liang J., Cao B.-G. Optimization of fractional order PID controllers based on genetic algorithms. – Proceedings of 2005 International Conference on Machine Learning and Cybernetics, vol. 9, 18-21 Aug. 2005, p.5686 – 5689.
51. Caruso M., Analog grounding considerations. – Honeywell, AN-103.

52. Chen J.-X., Li W.. Application of fuzzy control PID algorithm in temperature controlling systems. – 2003 International Conference on Machine Learning and Cybernetics, vol. 4, 2-5 Nov. 2003, p. 2601 – 2604.
53. Chien K. L., Hrones J. A., Reswick J. B. On automatic control of generalized passive systems. – Trans. ASME, 74, p. 175 – 185.
54. Chiu S. Using fuzzy logic in control applications: beyond fuzzy PID control. IEEE Control Systems Magazine, vol. 18, Issue 5, Oct 1998, p. 100 – 104.
55. СО -153-34.21.122—2003. Инструкция по устройству молниезащиты зданий, сооружений и промышленных коммуникаций. М.: Изд-во МЭИ, 2004. – 57 с.
56. Cristaldi L., Ferrero A., Muscas C., Salicone S., Tinarelli, R. The impact of Internet transmission on the uncertainty in the electric power quality estimation by means of a distributed measurement system. – IEEE Transactions on Instrumentation and Measurement, vol. 52, № 4, Aug. 2003, p. 1073 – 1078.
57. Cucej Z., Gleich D., Kaiser M., Planinsic P. Industrial networks. – Electronics in Marine, 2004. Proceedings 46th International Symposium Elmar 2004, 16-18 June 2004, p. 59 – 66.
58. Current Loop. Application Note № CLAN1495. – B&B Electronics, 1995. – 13 p.
59. Dashevsky Z., Rabinovich D. Ultrafast response and high sensitivity semiconductor thermocouple. – 15th IEEE Conf. on Thermoelectrics. – 1996, p. 321 – 325.
60. Data Access Automation Interface Standard. Version 2.02, Feb. 4, 1999. – OPC Foundation, 1994. – 100 p.
61. Determining Clock Accuracy Requirements for UART Communications. – Maxim/Dallas Application Note №2141. – 7 p.
62. Denisenko V. Nonparametric Model for PID Controller Auto-tuning// 18th IEEE International Conference on Control Applications. Part of 2009 IEEE Multi-conference on Systems and Control. Saint Petersburg, Russia, July 8-10, 2009, p. 43-47.

63. Denisenko V.V. The dynamic error of a multichannel measurement system// Measurement Techniques: vol. 52, Issue 1, 2009, p.1-4.
64. Denisenko V.V. Modifications of PID Regulators // Automation and Remote Control, 2010. V. 72. No. 6. P.345-355.
65. Dichen L.; Suqiao D. The study of lightning-protection grounding for power dispatching and communication equipment. – Environmental Electromagnetics, 2000. CEEM 2000. Proceedings. Asia-Pacific Conference on Date: 2000, Pages: 351 – 354
66. Ding S.-F., Shi Z.-Z., Wang V.-C., Li S.-S.. A novel feature extraction algorithm. – Proceedings of 2005 International Conference on Machine Learning and Cybernetics, 2005, vol. 3, 18-21 Aug. 2005, p.1762 – 1767.
67. Dierckx P. Curve and surface fitting with splines. – Claredon Press. Oxford.,1996. – 285 p.
68. Durham M.O., Arnold, R.G. Data quality and grounding considerations for a medical facility. 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on Circuits and Systems, Vol. 2, 4-7 Aug. 2002 p. II-184 – II-187.
69. Escamilla -Ambrosio P.J., Mort N. A novel design and tuning procedure for PID type fuzzy logic controllers. – 2002 First International IEEE Symposium on Intelligent Systems, 2002, vol. 1, 2002, p. 36 – 41.
70. Fault -tolerant Systems S7-400H. System Manual. – Siemens, Sept. 2007. – p. 376.
71. Feldmann K., Wolf, W., Weber, M. Development of an open, event-based and platform independent architecture for distributed and intelligent control systems. 2nd IEEE International Conference on Industrial Informatics, 2004. INDIN '04. 2004 24-26 June 2004, p. 560 – 566.
72. Feng D.-M., Pan F., Han R.-C. Improved self-adaptive Smith predictive control scheme for time-delay system. – Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on., vol. 1, 2002, p.463 – 466

73. Feng D.-M., Pan F., Han R.-C. Improved self-adaptive Smith predictive control scheme for time-delay system. – 2002 International Conference on Machine Learning and Cybernetics, vol. 1, 2002, p. 463 – 466.
74. Fernandez R.O., Peres, R. Remote monitoring and control of electronic nutrunners with embedded Internet. – IEEE 2002 28th Annual Conference of the IECON 02, vol. 3, Nov. 5-8, 2002 p. 2371 – 2375.
75. Fink A., Nelles O. Nonlinear internal model control based on local linear neural networks. – 2001 IEEE International Conference on Systems, Man, and Cybernetics, vol. 1, 2001, p. 117 – 122.
76. Fleming P. J., Purshouse R. C. Genetic algorithms in control systems engineering. – IFAC Professional Brief. – <http://www.ifac-control.org>.
77. Floating oscilloscope measurements ... and operator protection – Tektronix Technical Brief, 1998.
78. Fourteen Industrial Ethernet solutions under the spotlight. – The Industrial Ethernet Book, Issue 35 (ethernet.industrial-networking.com).
79. Fowler K. Grounding and shielding. II. Grounding and return. – IEEE Instrumentation & Measurement Magazine, vol. 3, Issue 2, Jun 2000, p. 45 – 48.
80. Fowler K. Grounding and shielding. I. Noise. – IEEE Instrumentation & Measurement Magazine, vol. 3, Issue 2, Jun 2000, p. 41 – 44.
81. Furr S. What is Real Time and Why Do I Need It? – www.qnx.com.
82. Galvin P., Silberschatz A. Operation Systems Concepts, 5-th Edition. – Addison-Wesley, 1998. – 890 p.
83. Gang Y., He J., Bicheng M., Zou J., Chen S., Gao Y., Zeng R. Electromagnetic interference in control cables of substation caused by short circuit fault. Power System Technology. – 2002. Proceedings. International Conference on PowerCon 2002. Vol. 2, 2002, p. 980 – 984.

84. Ganjefar S., Momeni H., Sharifi J. F., Beheshti H. M. T. Behavior of Smith predictor in teleoperation systems with modeling and delay time errors. – Proceedings of 2003 IEEE Conference on Control Applications, 2003. CCA 2003, vol. 2, 23-25 June 2003, p. 1176 – 1180.
85. Gao W., Li Y., Liu G., Zhang T.. An adaptive fuzzy Smith control of time-varying processes with dominant and variable delay. – Proceedings of the 2003 American Control Conference, vol. 1, 4-6 June 2003, p. 220 – 224.
86. Garcia J., Palomo F.R., Luque A., Aracil C., Quero J.M., Carrión D., Gamiz F., Revilla P., Perez-Tinao J., Moreno M., Robles P., Franquelo L.G. Reconfigurable distributed network control system for industrial plant automation. – IEEE Transactions on Industrial Electronics, Vol. 51, Issue 6, Dec. 2004, p. 1168 – 1180.
87. Ghiani E., Locci N., Muscas C. Auto-Evaluation of the Uncertainty in Virtual Instruments. – IEEE Trans. on Instrumentation and Measurement, Vol. 53, No. 3, 2004. – p. 250 – 672-677.
88. Glickman S., Kulesky R., Nudelman G. Identification-Based PID Control Tuning for Power Station Processes. – IEEE Trans. on Control System Technology, v. 12, № 1, 2004. p. 123 – 132.
89. Grassi E.; Tsakalis K.S. Integrated system identification and PID controller tuning by frequency loop-shaping. – Proceedings of the 38th IEEE Conference on Decision and Control, vol. 2, 1999, p. 1517 – 1522.
90. Grega W., Kolek K. Simulation and realtime control: from Simulink to industrial application. – 2002 IEEE Int. Symp. on Computer Aided Control Syst. Design Proc., Sept. 18-20, Glasgow, 2002. – p. 104 – 109.
91. Grimm G., Messina M.J., Tuna S.E., Teel A.R. Model predictive control: for want of a local control Lyapunov function, all is not lost. – IEEE Transactions on Automatic Control, vol. 50, Issue 5, May 2005, p. 546 – 558.

92. Groza V., Cretu V., Bogoevici M., Petriu E.M. Distributed virtual instrumentation architecture. – Proceedings of the First ISA/IEEE Conference Sensor for Industry, 2001, 2001, p. 177 – 180.
93. Gu J.-J., Shen L., Zhang L.-Y.. Application of Internal Model and Self-Adaptive PSD Controller in the Main Steam Temperature System. – Proceedings of 2005 International Conference on Machine Learning and Cybernetics, 2005, vol. 1, 18-21 Aug. 2005, p. 570 – 573.
94. Guo Q., Wang D., Han P., Yu P.. Virtual Cascade PID controller with strong robustness. – IEEE International Conference on Systems, Man and Cybernetics, 2003, vol. 1, 5-8 Oct. 2003, p. 654 – 659.
95. Guo J.; Boroyevich, D.; Edwards, S.H., Distributed, modular, open control architecture for power conversion systems. – IEEE 35th Annual Power Electronics Specialists Conference, vol. 3, 20-25 June 2004, p. 2258 – 2263.
96. Gutierrez , J.A.; Durocher, D.B.; Bin Lu; Habetler, T.G. Applying Wireless Sensor Networks in Industrial Plant Energy Evaluation and Planning Systems. – Pulp and Paper Industry Technical Conference, 2006, 18-23 June 2006, p. 1 – 7.
97. H. Fischer, etc. The COMPASS Data Acquisition System. – IEEE Trans. on Nuclear Science, vol. 49, No. 2, April 2002, p. 443 – 447.
98. Hac A. Wireless Sensor Network Designs. – John Wiley & Sons, Ltd, 2003. – 391 p.
99. Hagglund T. A predictive PI controller for processes with long dead times. – IEEE Control Systems Magazine, vol. 12, Issue 1, Feb 1992, p. 57 – 60.
100. Hang C.C., Wang Q.-G., Yang X.-P. A modified smith predictor for a process with an integrator and long dead-time. – The 2002 International Conference on Control and Automation, 2002. ICCA. Final Program and Book of Abstracts. June 16-19, 2002, p.134 – 134.

101. HART field communication protocol. Application guide HCF LIT 34. – HART Communication Foundation, 1999. – 80 p., www.hartcomm.org.
102. Hayashi H., Takabayashi Y., Tsuji H., Oka M. Rapidly increasing application of Intranet technologies for SCADA (supervisory control and data acquisition system). – IEEE/PES Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. Vol. 1, 6-10 Oct. 2002, p. 22 – 25.
103. Hekkala A.; Tarvainen, T. Screening attenuation of differential cable-connector assemblies. – 8th IEEE Workshop on Signal Propagation on Interconnects Proceedings, 2004, 9-12 May 2004, p. 175 – 178.
104. Helei Wu; Jing Ming; Yirong Yang; Shanan Zhu. Integrating embedded-Web technology and real-time Ethernet for modern distributed control. – Fifth World Congress on Intelligent Control and Automation, 2004. WCICA 2004. vol. 2, 15-19 June 2004, p. 1323 – 1325.
105. Hemerly E.E. PC-based packages for identification, optimization, and adaptive control. IEEE Control Systems Magazine, vol. 11, Issue 2, Feb. 1991, p. 37 – 43.
106. Hirai J., Kim T.-W., Kawamura A. Practical study on wireless transmission of power and information for autonomous decentralized manufacturing system. – IEEE Trans. Ind. Electron., vol. 46, № 2, 1999, p. 349–359.
107. Ho W.K., Lee T.H., Han H.P., Hong Y. Self-tuning IMC-PID control with interval gain and phase margins assignment. – IEEE Transactions on Control Systems Technology, vol. 9, Issue 3, May 2001, p.535 – 541.
108. Ho W. K., Lee T. H., Han H. P., and Hong Y. Self-Tuning IMC-PID Control with Interval Gain and Phase Margins Assignment. – IEEE Trans on Control Syst. Techn., vol. 9, №. 3, May 2001, p.p. 535-541.
109. Ho M.-T. and Lin C.-Y. PID Controller Design for Robust Performance. – IEEE Trans. on Automatic Control, v. 48, № 8, Aug. 2003, p.1404-1409.

110. Ho H.F., Wong Y.K., Rad A.B. Adaptive PID controller for nonlinear system with tracking performance. – Physics and Control, 2003. Proceedings. 2003 International Conference. Vol. 4, 20-22 Aug. 2003, p. 1315 – 1319.
111. Ho S.-J., Shu L.-S., Ho S.-Y.. Optimizing fuzzy neural networks for tuning PID controllers using an orthogonal simulated annealing algorithm OSA. – IEEE Transactions on Fuzzy Systems, vol. 14, Issue 3, June 2006, p. 421 – 434.
112. Ho S.-H., Ho S.-Y., Shu L.-S. OSA: orthogonal simulated annealing algorithm and its application to designing mixed H_2/H_∞ optimal controllers. – IEEE Transactions on Systems, Man and Cybernetics, Part A, vol. 34, Issue 5, Sept. 2004, p. 588 – 600.
113. Hodel A. S., Hall C. E. Variable-Structure PID Control to Prevent Integrator Windup. – IEEE Trans. on Industrial Electronics, v. 48, № 2, 2001, p. 442-451.
114. Hogenauer E. An economical class of digital filters for decimation and interpolation. – Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing, Vol. 29, No. 2, 1981. – p. 155 – 162.
115. Hong-Yan Li. Web-based remote monitoring and control for process plants. International Conference on Machine Learning and Cybernetics, 2005. Proceedings of 2005, vol. 2, 18-21 Aug. 2005, p. 936 – 941.
116. Hsuan-Ming Feng. A self-tuning fuzzy control system design. – IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th. Vol. 1, 25-28 July 2001, p. 209 – 214 vol.1.
117. Hu H., Hu Q., Lu Z., Xu D.. Optimal PID controller design in PMSM servo system via particle swarm optimization. – Industrial Electronics Society, 2005. IECON 2005. 32nd Annual Conference of IEEE. 6-10 Nov. 2005, p.5.
118. Hu B., Mann G.K.I., Gosine R.G. New methodology for analytical and optimal design of fuzzy PID controllers. – IEEE

- Transactions on Fuzzy Systems, vol. 7, Issue 5, Oct 1999, p. 521 – 539.
119. Hu B.-G., Mann G.K.I., Gosine R.G. A systematic study of fuzzy PID controllers-function-based evaluation approach. – IEEE Transactions on Fuzzy Systems, vol. 9, Issue 5, Oct 2001, p. 699 – 712.
 120. Huang J.-J., DeBra D.B. Automatic Smith-predictor tuning using optimal parameter mismatch. – IEEE Transactions on Control Systems Technology, vol. 10, Issue 3, May 2002, p. 447 – 459.
 121. Huang Y., Yasunobu S. A general practical design method for fuzzy PID control from conventional PID control. – The Ninth IEEE International Conference on Fuzzy Systems, FUZZ IEEE 2000, vol. 2, 2000, p. 969 – 972.
 122. Huang H.-M.; Michaloski J., Tarnoff N., Nashman M. An open architecture based framework for automation and intelligent system control. – International IEEE/IAS Conference on Industrial Automation and Control: Emerging Technologies, 1995., 22-27 May 1995, p. 282 – 288.
 123. Hummer O., Sunder C., Zoitl A., Strasser T., Rooker M.N., Ebenhofer G. Towards Zero-downtime Evolution of Distributed Control Applications via Evolution Control based on IEC 61499. – IEEE Conference on Emerging Technologies and Factory Automation, 2006. ETFA '06, 20-22 Sept. 2006, p. 1285 – 1292.
 124. Hwang H.-S., Choi J.-N., Lee W.-H., Kim J.-K.. A tuning algorithm for the PID controller utilizing fuzzy theory. – International Joint Conference on Neural Networks, IJCNN '99, vol. 4, 1999, p. 2210 – 2215.
 125. IEC 61131-7(2000) Programmable controllers. Part 7. Fuzzy control programming. 2000.
 126. IEC 61131-3(2003). Programmable controllers – Part 3: Programming languages. Опубликовано 31.01.2003.
 127. IEC 61511-1(2003)/Cor.1(2004). Functional safety – Safety instrumented systems for the process industry sector – Part 1:

- Framework, definitions, system, hardware and software requirements.
128. IEC 61511-2(2003) Functional safety – Safety instrumented systems for the process industry sector – Part 2: Guidelines for the application of IEC 61511-1.
 129. IEC 61511-3(2003) Functional safety – Safety instrumented systems for the process industry sector – Part 3: Guidance for the determination of the required safety integrity levels.
 130. IEC 61511-1(2003)/Cor.1(2004). Functional safety – Safety instrumented systems for the process industry sector – Part 1: Framework, definitions, system, hardware and software requirements. Corrigendum 1.
 131. IEC 61511-2(2003) Functional safety – Safety instrumented systems for the process industry sector – Part 2: Guidelines for the application of IEC 61511-1.
 132. IEC 61511-3(2003) Functional safety – Safety instrumented systems for the process industry sector – Part 3: Guidance for the determination of the required safety integrity levels.
 133. IEC 61499-1(2005). Function blocks – Part 1: Architecture.
 134. IEC 61499-2(2005). Function blocks – Part 2: Software tool requirements.
 135. IEC 61499-4(2005). Function blocks – Part 4: Rules for compliance profiles.
 136. IEC 61804-2(2006). Function blocks (FB) for process control – Part 2: Specification of FB concept.
 137. IEC 61804-3(2006). Function blocks (FB) for process control – Part 3: Electronic Device Description Language (EDDL).
 138. IEC/TR 61499-3(2004). Function blocks for industrial-process measurement and control systems – Part 3: Tutorial information.
 139. IEC /TR 61131-8(2000) Programmable controllers – Part 8: Guidelines for the application and implementation of programming languages. Опубликовано 01.01.2000.

140. IEC /TR 61499-3(2004). Function blocks for industrial-process measurement and control systems – Part 3: Tutorial information.
141. IEC /TS 61804-1(2003). Function blocks (FB) for process control – Part 1: Overview of system aspects.
142. IEC /TR 61804-4(2006). Function blocks (FB) for process control – Part 4: EDD interoperability guideline.
143. IEEE Std. 802.3. IEEE standard for information technology – Telecommunications and information exchange between systems – Local and metropolitan area network – Specific requirements. Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. – IEEE Computer Society. – 2005, section 1, 2, 3, 4, 5. – 2628 p.
144. IEEE Std 802.15.4™-2003. IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE Computer Society. IEEE-SA Standards Board, 12 May 2003. – 679 p.
145. IEEE Std 802.1D™- 2004. IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges. (Revision of IEEE Std 802.1D -1998 incorporating IEEE Std 802.1t -2001 IEEE Std 802.1w -2001, replaced by 802.1D-2004). – IEEE, 2004. – 281 p.
146. IEEE Std 802.1Q -2005. IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks. (Incorporates IEEE Std 802.1Q-1998, IEEE Std 802.1u™-2001, IEEE Std 802.1v™-2001, and IEEE Std 802.1s™-2002). – IEEE, 2006. – 303 p.
147. IEEE Std. 802.3. IEEE standard for information technology – Telecommunications and information exchange between systems – Local and metropolitan area network – Specific

- requirements. Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. – IEEE Computer Society. – 2005, section 1, 2, 3, 4, 5. – 2628 p.
148. Insaurralde C.C., Seminario M.A., Jimenez J.F., Giron-Sierra J.M. IEC 61499 Model for Avionics Distributed Fuel Systems with Networked Embedded Holonic Controllers. – IEEE Conference on Emerging Technologies and Factory Automation, 2006. ETFA '06, 20-22 Sept. 2006, p. 388 – 396.
 149. International standard IEC 61499-1. First Edition, 2005-01. – International Electrotechnical Commission, 3 PO Box 131, CH-1211 Geneva 20, Switzerland, web: www.iec.ch.
 150. Introduction to the Controller Area Network (CAN). Application Report SLOA101. – Texas Instruments, Aug. 2002. – 17 p.
 151. Ishihara Y., Shiota Y., Sekiguchi K., Sato S., Sawai K. A recent trends in the implementation of Intranet based measurement and monitoring. – Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES, vol. 3, 6-10 Oct. 2002, p. 2261 – 2266.
 152. ISO 11898-1:2003. Part 1: Data link layer and physical signaling. – 45 p.
 153. ISO 11898-2:2003 Part 2: High-speed medium access unit. – 21 p.
 154. ISO 11898-3:2006 Part 3: Low-speed, fault-tolerant, medium-dependent interface. – 25 p.
 155. ISO 11898-4:2004 Part 4: Time-triggered communication. – 32 p.
 156. ISO 15765-1:2004 Diagnostics on Controller Area Networks (CAN). Part 1: General information – 4 p.
 157. ISO 16484 -3:2005. Building automation and control systems (BACS) – Part 3: Functions.
 158. ISO /DIS 11898-5 Part 5: High-speed medium access unit with low-power mode (в стадии разработки)

159. Ivanova E., Hadjiski M. Rules based adaptation of Smith predictor. – Intelligent Systems, 2002. Proceedings. 2002 First International IEEE Symposium, vol. 3, 2002, p. 26 – 30.
160. Iwanitz F., Lange J. OPC: fundamentals, implementation, and application. – Softing, 2-nd ed., Heidelberg: Huting, 2002. – 225 p.
161. Iwanitz F., Lange J. OPC fundamentals, Implementation, and Application. – 2nd rev. ed., Heidelberg: Hthig, 2002, 225 p.
162. Iwasa T., Morizumi N., Omatu S. Temperature control in a batch process by neural networks. – The 1998 IEEE International Joint Conference on Neural Networks Proceedings, vol. 2, 4-9 May 1998, p. 992 – 995.
163. Jammes F., Smit H., Arandyelovitch C., Depeisses F. Intelligent device networking in industrial automation. – 2nd IEEE International Conference on Industrial Informatics, INDIN '04, 24-26 June 2004, p. 449 – 456.
164. Jiang P., Ren H.; Zhang L., Wang Z.; Xue A. Reliable Application of Wireless Sensor Networks in Industrial Process Control. – The Sixth World Congress on Intelligent Control and Automation, 2006. WCICA 2006. Vol. 1, 21-23 June 2006, p. 99 – 103.
165. Jiao H.; Jianguo L.; Min G. Embedded temperature Web controller based on Internet. – IEEE International Symposium on Industrial Electronics, vol. 1, 4-7 May 2004, p. 237 – 240.
166. Kamrani E., Momeni H.R., Sharafat A.R. Novel adaptive control system for stable teleoperation via internet. – Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005. 28-31 Aug. 2005, p. 1164 – 1169.
167. Kapsalis V., Charatsis K., Kalogeras A.P., Papadopoulos G. Web gateway: a platform for industry services over Internet. – Proceedings of the 2002 IEEE International Symposium on Industrial Electronics, 2002. ISIE 2002, vol. 1, 8-11 July 2002, p. 73 – 77.

168. Kapsalis V., Charatsis K., Georgoudakis M., Papadopoulos G.
169. Karakawa K., Abe N., Ichihara H. Joint design method of closed-loop identification and IMC structure for temperature control system with time delay. – SICE 2002. Proceedings of the 41st SICE Annual Conference, vol. 3, 5-7 Aug. 2002, p.1592 – 1595.
170. Karimi , A.; Garcia, D.; Longchamp, R. PID controller tuning using Bode’s integrals. – Control Systems Technology, IEEE Transactions on, vol. 11, Issue 6, Nov. 2003, p.: 812 – 821.
171. Kato , M.; Yamamoto, T.; Fujisawa, S. A Skill-Based PID Controller Using Artificial Neural Networks. – Computational Intelligence for Modeling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, vol. 1, 28-30 Nov. 2005, p. 702 – 707.
172. Kawafuku , R.; Sasaki, M.; Kato, S. Self-tuning PID control of a flexible micro-actuator using neural networks. – Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on. Vol. 3, 11-14 Oct 1998, p. 3067 – 3072 vol.3.
173. Ke H., Lee W.-J., Chen M.-S., Liu J.-P., Yang J. S. Grounding techniques and induced surge voltage on the control signal cables. – IEEE Trans. on Industry Applications, Vol. 34, No. 4, 1998, p. 663- 668.
174. Keel L. H., Rego J. I., and Bhattacharyya S. P. A New Approach to Digital PID Controller Design. – IEEE Trans. on Automatic Control, v. 48, NO. 4, APRIL 2003. p. 687 – 692.
175. Kim D.H.. Comparison of PID controller tuning of power plant using immune and genetic algorithms. – 2003 IEEE International Symposium on Computational Intelligence for Measurement Systems and Applications, 2003. CIMSAs ‘03, 29-31 July 2003, p.169 – 174.
176. Kim D.H., Cho J.H., Robust tuning for disturbance rejection of PID controller using evolutionary algorithm. – IEEE Annual Meeting on Fuzzy Information, 2004. Processing NAFIPS ‘04, vol. 1, 27-30 June 2004, p. 248 – 253.

177. Kim K. Real-Time Software Framework for Distributed Control Systems. – Proceedings on the 15th CISL Winter Workshop, Kushu, Japan, Feb. 2002.
178. Kim W., Ji K. Ambike A. Real-time operating environment for networked control systems. – IEEE Transactions on Automation Science and Engineering, Vol. 3, Issue 3, July 2006, p. 287 – 296.
179. Klimchynski I. Extensible embedded web server architecture for Internet-based data acquisition and control. – IEEE Sensors Journal, vol. 6, No. 3, June 2006, p. 804 – 811.
180. Kolaric D.; Skala K.; Dubravic A. Remote control and measurement of temperature over the web. – 47th International Symposium ELMAR, 2005. 8-10 June 2005, p. 117 – 119.
181. Kolla S.; Michaloski, J.; Rippey, W. Evaluation of component-based reconfigurable machine controllers. – Proceedings of the 5th Biannual World Automation Congress, vol. 14, 2002, p. 625 – 630.
182. Kondakci S. An approach to noise reduction in human skin admittance measurements. – 2001 Proc/ of the 23rd Annual EMBS Int. Conf., Oct. 25-28, Stambul, 2001. – p. 2028 – 2031.
183. Kosc D., Hamer P. S. Grounding practices – A system-wide systematic approach. – IEEE Trans. on Industry Applications., Vol. 39, No. 5, 2003, p. 1475 – 1485.
184. Kristiansson B., Lennartson B. Robust PI and PID controllers including Smith predictor structure. – Proceedings of the 2001 American Control Conference, vol. 3, 2001, p. 2197 – 2202.
185. Labrosse J. J. MicroC OS II: The Real Time Kernel. – CMP Books, 2002. – 648 p.
186. Laddomada M.; Lo Presti, L.; Mondin, M.; Ricchiuto, C. An efficient decimation sinc-filter design for software radio applications. – 2001 IEEE Third Workshop on Signal Processing. Advances in Wireless Communications, 2001, p. 337 – 339.
187. Lange J.. Free choice of platform. The next OPC generation. – Softing AG, (www.softing.com), Nov. 2006. – 4 p.

188. Layer E. Theoretical principles for establishing a hierarchy of Dynamic Accuracy with the integral-Square-Error as an Example. – IEEE Transactions on Instrumentation and Measurement, Vol. 46, No. 5, 1997. – p. 1178 – 1182.
189. Lee W.S., Shi J. Modified IMC-PID controllers and generalized PID controllers for first-order plus dead-time processes. – 7th International Conference on Control, Automation, Robotics and Vision, 2002. ICARCV 2002, vol. 2, 2-5 Dec. 2002, p. 898 – 903.
190. Lee W.S., Shi J. Modified IMC-PID controllers and generalized PID controllers for first-order plus dead-time processes. – Seventh Int. Conf. on Control, Automation, Robots and Vision (ICARCV'02), Dec. 2002, Singapore, p.p. 898-903.
191. Lee C.C. Fuzzy logic in control systems: fuzzy logic controller. – IEEE Transactions on Systems, Man and Cybernetics, vol. 20, Issue 2, Mar/Apr 1990, p. 404 – 418.
192. Lee B.-K., Chen H.-W., Chen B.-S.. Power control of cellular radio systems via robust Smith prediction filter. – IEEE Transactions on Wireless Communications, vol. 3, Issue 5, Sept. 2004, p.1822 – 1831.
193. Leva A., Cox C., Ruano A. Hands-on PID autotuning: a guide to better utilization. IFAC Professional Brief. – <http://www.ifac-control.org>.
194. Leva A. and Colombo A.M. IMC-based Synthesis of the Feedback Block of ISA-PID Regulators. Proc. ECC 2001, Porto (P).
195. Leva A., Piroddi L. Model-based PID autotuning enhanced by neural structural identification. – Proceedings of the 2004 American Control Conference, 2004, vol. 3, 30 June-2 July 2004, p. 2427 – 2432.
196. Leva A. Autotuning process controller with enhanced load disturbance rejection. – Proceedings of the 2004 American Control Conference, 2004, vol. 2, 30 June-2 July 2004, p.1400 – 1405.
197. Lewis R.W. Programming industrial control systems using IEC 113-3 Revised edition. – The Institution of Electrical Engineers, London, UK, 1998. – 329 p.

198. Li Y., Ang K.H, Chong G.C.Y. Patents, Software, and Hardware for PID control. An overview and analysis of the current art. *IEEE Control Systems Magazine*, Feb. 2006, p. 41-54.
199. Li J., Xie J.-Y.; Wu Z.-M.. Design of Disturbance Rejection PID Controllers for Time Delay System Based on Genetic Algorithms. – *Neural Networks and Brain*, 2005. ICNN&B '05. International Conference on. vol. 2, 13-15 Oct. 2005, p. 876–880.
200. Li Q., Cheng Z., Qian J. Self-learning fuzzy PID controller based on neural networks. – *Proceedings of the American Control Conference*, vol. 3, 21-26 Jun 1998, p. 1860 – 1861.
201. Li H.-X., Gatland H.B. Conventional fuzzy control and its enhancement. – *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 26, Issue 5, Oct 1996, p. 791 – 797.
202. Li Y., Ang K.H., Chong G.C.Y. Patents, software, and hardware for PID control: an overview and analysis of the current art. – *IEEE Control Systems Magazine*, vol. 26, Issue 1, Feb. 2006, p. 42 – 54.
203. Li H.-X., Zhang L., Cai K.-Y., Chen G. An improved robust fuzzy-PID controller with optimal fuzzy reasoning. – *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 35, Issue 6, Dec. 2005, p.1283 – 1294.
204. Li D., Serizawa Y., Kiuchi M. Concept design for a Web-based supervisory control and data-acquisition (SCADA) system. – *Asia Pacific*
205. Lian F. -L., Yook J.K., Tilbury D.M., Moyne J. Network architecture and communication modules for guaranteeing acceptable control and communication performance for networked multi-agent systems. – *IEEE Transactions on Industrial Informatics*, vol. 2, No. 1, Feb. 2006, p. 12 – 24.
206. Liang C., Wei Z. EMI in hydropower plant and EMC design for its computer monitoring and control system. – *China Electromagnetic Compatibility*, 2002, 3rd International Symposium, 21-24 May 2002 p. 378 – 381.

207. Lin P. I.-H., Broberg, H.L. Internet-based monitoring and controls for HVAC applications. – IEEE Industry Applications Magazine, vol. 8, No. 1, Jan/Feb 2002, p. 49 – 54.
208. Liu G., Tang T., Liu L.. Study on the Web based remote monitoring and controlling system for heat exchanger. – IEEE International Symposium on Communications and Information Technology, 2005. ISCIT 2005. vol. 2, 12-14 Oct. 2005, p. 1301 – 1304.
209. Liu P.X., Farias W., Gibson S., Ross, D. Remote control of a robotic boat via the Internet. – 2005 IEEE International Conference on Information Acquisition, 27 June – 3 July 2005, p. 548-553.
210. Ljung L. and T. Glad. Modeling of Dynamic Systems. – Prentice Hall, Englewood Cliffs, N.J. 1994, 255 p.
211. Ljung L. System Identification – Theory for the User. – Prentice Hall, Upper Saddle River, N.J. 2nd edition, 1999, 375 p.
212. Long Transmission Lines and Data Signal Quality. National Semiconductor Application Notes 808, March 1992. – 23 p.
213. Low K.S., Win W.N.N., Er M.J. Wireless Sensor Networks for Industrial Environments. – International Conference on Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, Vol. 2, 28-30 Nov. 2005, p. 271 – 276.
214. Low level measurements. Keithley, 5-th edition.
215. Malek-Zadeh A.; Dietsch, H. Web-Based Information Technology for Remote Handling of Electrical Drives and Control Devices: An Integrated Approach. International Conference on Systems and International Conference on Mobile Communications and Learning Technologies Networking, 2006. ICN/ICONS/MCL 2006, 23-29 April 2006, p. 176 – 176.
216. Malki H.A., Misir D. Determination of the control gains of a fuzzy PID controller using neural networks. – Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, vol. 2, 8-11 Sep 1996, p. 1303 – 1307.

217. Mamdani E. H. Application of fuzzy algorithms for control of simple dynamic plant. – Proc. Inst. Elect. Eng. Contr. Sci., vol. 121, 1974, p. 1585–1588.
218. Mamdani E. H., Assilian S. An experiment in linguistic synthesis with a fuzzy logic controller. – Int. J. Man-Mach. Stud., vol. 7, 1975, p. 1–13.
219. Mamdani E.H. Application of fuzzy algorithm for simple dynamic plant/Proc. IEEE 1974, №12, p.1585-1588.
220. Mann G.K.I., Bao-Gang Hu., Gosine R.G. Analysis of direct action fuzzy PID controller structures. – IEEE Transactions on Systems, Man and Cybernetics, Part B, vol. 29, Issue 3, Jun 1999, p. 371 – 388.
221. Mann G.K.I.; Hu B.-G., Gosine R.G. Fuzzy PID controller structures. – IEEE 1997 Canadian Conference on Electrical and Computer Engineering, vol. 2, 25-28 May 1997, p. 788 – 791.
222. Mann G.K.I., Hu B.-G., Gosine R.G. Two-level tuning of fuzzy PID controllers. – IEEE Transactions on Systems, Man and Cybernetics, Part B, vol. 31, Issue 2, Apr 2001, p. 263 – 269.
223. Matausek M.R., Micic A.D. A modified Smith predictor for controlling a process with an integrator and long dead-time. – IEEE Transactions on Automatic Control, vol. 41, Issue 8, Aug 1996, p. 1199 – 1203.
224. Mauris G.Expression of Measurement Uncertainty in a Very Limited Knowledge Context: A Possibility Theory-Based Approach. – IEEE Trans. on Instr. and Measueremnt, vol. 56, No. 3, 2007. – p. 731-735.
225. Milanovi V., Gaitan M., Zaghloul M. E. Micromachined Thermocouple Microwave Detector by Commercial CMOS Fabrication. – IEEE Trans. on Microwave Theory and Techniques, vol. 46, No. 5, May 1998, p. 550 – 553.
226. Miller R.M., Kwok K.E., Shah S.L., Wood R.K. Development of a stochastic predictive PID controller. – Proceedings of the American Control Conference, vol. 6, 21-23 Jun 1995, p. 4204 – 4208.

227. Mitsubishi Safety Programmable Controller. MELSEC QS Series. CC-Link Safety System. Remote I/O Module. User's Manual. Mitsubishi Electric Corp. – p.114
228. Miyazaki K., Takamiya T., Tsukamoto H., Fabrication of micro-thin film thermocouples. – 22nd Int. Conf. on Thermoelectrics. – 2003, p. 673 – 676.
229. Modbus over serial line specification and implementation guide, v1.02. – Modbus.org, Dec 20, 2006. – 44 p.
230. Modbus application protocol specification v1.1a. – www.Modbus-IDA.org, June 4, 2004. – 51 p.
231. Modbus messaging on TCP/IP implementation guide, V1.0a. – www.Modbus-IDA.org, June 4, 2004, 46 p.
232. Modicon Modbus Protocol Reference Guide. PI-MBUS-300 Rev. J. – MODICON, Inc., Industrial Automation Systems, June 1996. – 121 p.
233. Mohamed F.A., Koivo H.N. Diesel Engine Systems with Genetic Algorithm Self Tuning PID Controller. – 2005 International Conference on Future Power Systems, 16-18 Nov. 2005, p.1 – 5.
234. Mohan B.M., Sinha A. A simplest fuzzy PID controller: analytical structure and stability analysis. – India Annual Conference, 2004. Proceedings of the IEEE INDICON 2004. First. 20-22 Dec. 2004, p. 288 – 293.
235. Moon Y.-H., Ryu H.-S., Choi B.-K., Cho B.-H.. Modified PID load-frequency control with the consideration of valve position limits. – IEEE Power Engineering Society 1999 Winter Meeting, vol. 1, 31 Jan-4 Feb 1999, p. 701 – 706.
236. Moradi M. H. New techniques for PID Controller Design. – CCA 2003. Proceedings of 2003 IEEE Conference on Control Applications. Vol. 2, 23-25 June 2003 p. 903 – 908.
237. Moxa Wite Paper. Redundancy in automation. – Moxa Networking Co., Ltd. – 16 p., www.moxanet.com.
238. Mu J., Liu G.P. Rees D. Design of robust networked predictive control systems. – Proceedings of the 2005 American Control Conference, 8-10 June 2005, p. 638 – 643.

239. Nabielec J. An outlook on the dynamic error «blind» correction for the time-varying measurement channel. – Proceedings of the 21st IEEE Instrumentation and Measurement Technology Conference, 2004. IMTC 04, 18-20 May, Vol. 1, 2004. – p. 152 – 155.
240. Nakano K., Yamamoto T., Hinamoto T. A design of robust self-tuning GPC-based PID controllers – The 29th Annual Conference of the IEEE Industrial Electronics Society, IECON '03, vol. 1, 2-6 Nov. 2003, p. 285 – 290.
241. Natori K., Tsuji T., Ohnishi K., Hace A., Jezernik K. Robust bilateral control with Internet communication. – 30th Annual Conference of IEEE Industrial Electronics Society, IECON 2004, vol. 3, 2-6 Nov. 2004, p. 2321 – 2326.
242. Natori K., Tsuji T., Ohnishi K., Hace A., Jezernik K. Robust bilateral control with Internet communication. – 30th Annual Conference of IEEE Industrial Electronics Society, IECON 2004, vol. 3, Nov. 2-6, 2004, p. 2321 – 2326.
243. Navarrina , F.; Colominas, I.; Casteleiro, M. Why do computer methods for grounding analysis produce anomalous results? – IEEE Transactions on Power Delivery, vol. 18, Issue 4, Oct. 2003, p. 1192 – 1202.
244. Neag I.A.; Tyler, D.F. An open architecture for automatic test systems with system-level diagnosis capabilities. The 19th Digital Avionics Systems Conferences, 2000. Proceedings. DASC. vol. 2, 2000, p. 6B4/1 – 6B4/8.
245. Nelson J. P. High-Resistance Grounding of Low-Voltage Systems: A Standard for the Petroleum and Chemical Industry. IEEE Transactions on Industry ASpplications, Vol. 35, № 4, 1999 pp. 941-948.
246. Nikolakopoulos G.; Koundourakis M.; Tzes A. An integrated system based on Web and/or WAP framework for remote monitoring and control of industrial processes. IEEE International Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2003. VECIMS '03. 27-29 July 2003, p. 201 – 206.

247. Noise Control in Strain Gage Measurements. – Vishey Micro-Measurements, Tech. Note TN-501-2, 2007, 8 p.
248. Numerical recipes in C. The art of scientific computing. Second Edition. Cambridge University Press. <http://www.nr.com>.
249. Obika M., Yamamoto T. An evolutionary design of robust PID controllers. – 2005 IEEE International Conference on Mechatronics and Automation, vol. 1, 29 July-1 Aug. 2005, p.101 – 106.
250. Object Messaging Specification for the MODBUS/TCP Protocol. V.1, – Modbus-IDA, Nov. 08, 2004. – 27 p.
251. O'Brien , K.; Scheible, G.; Gueldner, H. Analysis of wireless power supplies for industrial automation systems. – Industrial Electronics Society, 2003. IECON '03. The 29th Annual Conference of the IEEE, vol. 1, 2-6 Nov. 2003, p. 367 – 372.
252. O'Dwyer A. PID compensation of time delayed processes 1998-2002: a survey. – Proceedings of the American Control Conference, Denver, Colorado, June 4-6. ZW3 p. 1494- 1499.
253. O'Dwyer , A. PID compensation of time delayed processes 1998-2002: a survey. – Proceedings of the 2003 American Control Conference, vol. 2, June 4-6, 2003, p.1494 – 1499.
254. OLE for Process Control. OPC Overview. Version 1.0. – Oct. 27, 1998. – 16 p.
255. OLE for Process Control. Data Access Standard. Version 1.0A. Sept. 11, 1997. – 229 c.
256. Omatu S., Iwasa T., Yoshioka M. Skill-based PID control by using neural networks. – 1998 IEEE International Conference on Systems, Man, and Cybernetics, vol. 2, 11-14 Oct 1998, p. 1972 – 1977.
257. OPC Foundation. OPC historical data access specification. Version 1.20. Released Dec. 10, 2003. – 160 p.
258. OPC Foundation. OPC unified architecture. Release candidate specification. Part 1: Concepts. Version 1.20. June 1, 2006. – 25 p.
259. Optically coupled 20 mA current loop transmitter. HCPL-4100. Technical Data. – Agilent Technology, 1999. – 12 p.

260. Overstreet J.W., Tzes A. An Internet-based real-time control engineering laboratory. – IEEE Control Systems Magazine, vol. 19, No. 5, Oct 1999, p. 19 – 34.
261. Oviedo J.J.E., Boelen T., van Overschee P. Robust advanced PID control (RaPID): PID tuning based on engineering specifications. – IEEE Control Systems Magazine, vol. 26, Issue 1, Feb. 2006, p.15 – 19.
262. Paraskevopoulos P.N., Pasgianos G.D., Arvanitis K.G. New tuning and identification methods for unstable first order plus dead-time processes based on pseudoderivative feedback control. – IEEE Transactions on Control Systems Technology, vol. 12, Issue 3, May 2004, p. 455 – 464.
263. Paulraj A. J., Gore D. A., Nabar R. U. B?lcskei H. An overview of MIMO communications – A key to gigabit wireless. – Proc. IEEE, vol. 92, № 2, 2004. – p. 198–218.
264. Peng Y., Vrancic D., Hanus R. Anti-windup, bumpless, and conditioned transfer techniques for PID controllers. – Control Systems Magazine, IEEE vol. 16, Issue 4, Aug. 1996 p. 48 – 57.
265. Pereira , D.S.; Pinto, J.O.P. Genetic algorithm based system identification and PID tuning for optimum adaptive control. – Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on. 2005, p. 801 – 806.
266. Perez-Aragon J., Recatala G. Distributed architecture for controlling a manufacturing cell. – IEEE International Conference on Systems, Man and Cybernetics, 2003. Vol. 2, 5-8 Oct. 2003, p. 1130 – 1135.
267. Pizzica S. Open systems architecture solutions for military avionics testing. – IEEE Aerospace and Electronic Systems Magazine, vol. 16, Issue 8, Aug. 2001, p. 4 – 9.
268. Popov A., Farag A., Werner H. Tuning of a PID controller Using a Multi-objective Optimization Technique Applied to A Neutralization Plant. – 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05, 12-15 Dec. 2005, p. 7139 – 7143.

269. Prodic A., Maksimovic D. Design of a digital PID regulator based on look-up tables for control of high-frequency DC-DC converters. – 2002 IEEE Workshop on Computers in Power Electronics, Proceedings, 3-4 June 2002, p.18 – 22.
270. Product Support. Mean Time Between Failures (MTBF) – list for SIMATIC products. – <http://support.automation.siemens.com/WW/view/en/16818490>.
271. Profibus RS 485-IS user and installation guideline. PROFIBUS Guideline, ver. 1.1, order №. 2.262. – PROFIBUS Nutzerorganisation, June 2003. – 38 p.
272. Profibus technology and application. System description. – Profibus Nutzerorganization, Oct. 2002. – 42 p.
273. Profibus . Technical description. – PROFIBUS Nutzerorganisation, 1999, 36 p. (www.profibus.com).
274. Programmable control products. Genius modular redundancy for fire and gas applications. – GE Fanuc Automation, GFK-1649A, Sept. 1999. – 50 p.
275. Prokop R., Korbel J., Matusu R. Relay-based autotuning: a second order algebraic design. – IEEE International Workshop on Intelligent Signal Processing, 2005. 1-3 Sept. 2005, p. 98 – 103.
276. Proper grounding for the automation industry. – Pulp and Paper Industry Technical Conference, Portland, OR 2001, 18-22 June 2001, p. 110-113.
277. Prytz G. Redundancy in Industrial Ethernet Networks. – 2006 IEEE International Workshop on Factory Communication Systems, June 27, 2006 p. 380 – 385.
278. Qingcang Yu; Bo Chen; Cheng, H.H. Web based control system design and analysis. – IEEE Control Systems Magazine, vol. 24, No. 3, Jun 2004, p. 45 – 57.
279. Qiu B.; Gooi H.B. Web-based SCADA display systems (WSDS) for access via Internet. – IEEE Transactions on Power Systems, Vol. 15, Issue 2, May 2000, p. 681 – 686.
280. Qiu B., Gooi H. B., Liu Y.; Chan E.K. Internet-based SCADA display system. IEEE Computer Applications in Power, vol. 15, No. 1, Jan 2002, p. 14 – 19.

281. Qu R.T.; Lim, R.; He, H.; Ding, Z.Q.; Aendenroomer, A.; Goh, K.M. A Web-enabled distributed control application platform for industrial automation. IEEE Conference Emerging Technologies and Factory Automation Proceedings. ETFA '03, vol. 2, 16-19 Sept. 2003, p. 129 – 132.
282. Quevedo J., T. Escobet. Digital control: past, present and future of PID control. – Proc. IFAC Workshop, Eds., Terrassa, Spain, Apr. 5–7, 2000.
283. Rad A. B., Lo W. L. Tsang K.M. Self-tuning PID controller using Newton-Raphson search method. – IEEE Transactions on Industrial Electronics, vol. 44, Issue 5, Oct 1997, p. 717 – 725.
284. Radwan F.A.; Martin, T.W. Real-time monitoring and controlling of an Allen-Bradley SLC 500 through the Internet. – 2003 IEEE International Conference on Industrial Technology, v. 1, 10-12 Dec. 2003 p. 387 – 392.
285. Rajsuman R. An overview of the open architecture test system. – Second IEEE International Workshop on Electronic Design, Test and Applications. DELTA 2004, 28-30 Jan. 2004, p. 341 – 346.
286. Reducing Noise in Switching for Test Systems Application Note 1441-2, Agilent Technologies, p. 14.
287. Reliability Consulting:
<http://support.automation.siemens.com/WW/view/en/16818490>.
288. Ren X.M., Rad A.B., Lo W.L., Chan P.T. Adaptive H/sub 2/ optimal control based on Smith predictor for continuous-time systems with unknown time delay. – International Conference on Control and Automation, ICCA '05, vol. 1, 26-29 June 2005, p.159 – 164.
289. Renton P., Bender P., Veldhuis S., Renton D., Elbestawi A., Teltz R., Bailey, T. Internet-based manufacturing process optimization and monitoring system. – Proc. on ICRA '02. IEEE International Conference on Robotics and Automation, 2002, v. 2, 11-15 May 2002, p. 1113 – 1118.
290. Robert Bosch GmbH: CAN Specification ver. 2.0, 1991.

291. RS-422 and RS-485 Application Note. B&B Electronics. June 2006. – 22 p.
292. Sairam K.V.S.S.S.S.; Gunasekaran N.; Redd, S.R. Bluetooth in wireless communication. – IEEE Communications Magazine, Vol. 40, Issue 6, Jun 2002, p. 90 – 96.
293. Sanchez P., M., Cruz P., J. Real time connection of programmable logic controllers to Excel spreadsheets. – Proc. of the 2002 IEEE International Symp. on Industrial Electronics, 2002, ISIE 2002, vol. 1, 2002. – p. 233 – 238.
294. Sanchez -Gasca J.J., Clark K., Miller N.W., Okamoto H., Kurita A., Chow J.H. Identifying linear models from time domain simulations. – IEEE Computer Applications in Power, vol. 10, Issue 2, Apr 1997, p. 26 – 30.
295. Schoen M. P. Dynamic Compensation of Intelligent Sensors. – IEEE Transactions on Instrumentation and Measurement, vol. 56, No. 5, 2007, p. 1992 – 2001.
296. Schrama R.J.P. Accurate identification for control: the necessity of an iterative scheme. – IEEE Transactions on Automatic Control, vol. 37, Issue 7, Jul 1992, p. 991 – 994.
297. Schwab C.; Tangermann M.; Ferrarini L. Web based methodology for engineering and maintenance of distributed control systems: the TORERO approach. – 3rd IEEE International Conference on Industrial Informatics, 2005. INDIN '05. 2005 10-12 Aug. 2005, p. 32 – 37.
298. Shen X., Wang Z., Sun Y. Wireless sensor networks for industrial applications. – Fifth World Congress on Intelligent Control and Automation, 2004. WCICA 2004, Vol. 4, 15-19 June 2004, p. 3636 – 3640.
299. Shen J.-C., Chiang H.-K.. PID tuning rules for second order systems. – 5th Asian Control Conference, vol. 1, 20-23 July 2004, p. 472 – 477.
300. Shestakov A. L. Dynamic error correction method. – IEEE Transactions on Instrumentation and Measurement, Vol. 45, No. 1, 1996. – p. 250 – 255.

301. Shi J., Lee W.S.. Set point response and disturbance rejection tradeoff for second-order plus dead time processes. – 5th Asian Control Conference, 2004, vol. 2, 20-23 July 2004, p. 881 – 887.
302. Shigemasa T., Yukitomo M., Kuwata R. A model-driven PID control system and its case studies. – Proceedings of the 2002 International Conference on Control Applications, 2002, vol. 1, 2002, p. 571 – 576.
303. Silva G. J., Datta A., Bhattacharyya S. P. On the Stability and Controller Robustness of Some Popular PID Tuning Rules. – IEEE Trans. on Automatic Control, vol. 48, № 9, Sept. 2003, p. 1638-1641.
304. Silva G.J., Datta A., Bhattacharyya S.P. New results on the synthesis of PID controllers. – IEEE Transactions on Automatic Control, vol. 47, Issue 2, Feb 2002, p. 241 – 252.
305. SIMATIC Automation System S7-300. Fail-Safe Signal Modules. Manual. – Siemens, Edition 04/2006, 236 p.
306. Sio K.C., Lee C.K. Stability of fuzzy PID controllers. – IEEE Transactions on Systems, Man and Cybernetics, Part A, vol. 28, Issue 4, Jul 1998, p. 490 – 495.
307. Skoczowski S., Domek S., Pietruszewicz, K., Broel-Plater, B. A method for improving the robustness of PID control. – IEEE Transactions on Industrial Electronics, vol. 52, Issue 6, Dec. 2005, p.1669 – 1676.
308. Smit J., Hancke G.P. The design and implementation of a general-purpose, secure, measurement and control network incorporating Internet-based access. – Proceedings of the 20th IEEE Instrumentation and Measurement Technology Conference, 2003. IMTC '03. Vol. 2, May 20-22, 2003, p. 1643 – 1647.
309. Smith O.J.M. Close Control of Loops with Dead Time. – Chemical Engineering Progress, v. 53, 1957, pp. 217-235.
310. Software functional requirements specification. Athena DeviceNet interface module. Rev. 1.40, Feb 2, 2001. – 37 p.

311. Specification of the Bluetooth System. Master Table of Contents & Compliance Requirements. Covered Core Package version: 2.0 + EDR Current Master TOC issued: 4 November 2004. – 1230 p.
312. Stojic M.R., Matijevic F.S., Draganovic L.S. A robust Smith predictor modified by internal models for integrating process with dead time. – IEEE Transactions on Automatic Control, vol. 46, Issue 8, Aug 2001, p. 1293 – 1298.
313. Sun X., Wu J., Wang H. Nonlinear PID controller design based on GA. – Proceedings of the 4th World Congress on Intelligent Control and Automation, 2002, vol. 2, 2002, p.1444 – 1447.
314. Sunder C., Zoitl A., Rainbauer M., Favre-Bulle B. Hierarchical Control Modelling Architecture for Modular Distributed Automation Systems. – IEEE International Conference on Industrial Informatics, 2006, Aug. 2006, p. 12 – 17.
315. Susanu M.; Dumur, D. Open architecture hierarchical predictive control for saturation avoidance. – 9th IEEE International Workshop on Advanced Motion Control, March 2006, p. 33 – 38.
316. Swiniarski R.W. Neural network based self-tuning PID controller with Fourier transformation of temporal patterns. – Industrial Electronics Society, 1990. IECON '90., 16th Annual Conference of IEEE, 27-30 Nov. 1990, p. 1227 – 1232.
317. Takagi S.-I., Oki T., Yamamoto T., Kaneda M. A skill-based PID controller using artificial neural networks. – 1997 IEEE International Conference on Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation', vol. 5, 12-15 Oct 1997, p. 4454 – 4459.
318. Takao K., Yamamoto T., Hinamoto T. Design of a memory-based self-timing PID controller. – 43rd IEEE Conference on Decision and Control, 2004. CDC, vol. 2, 14-17 Dec. 2004, p.1598 – 1603.
319. Takatsu H., Itoh T. Future Needs for Control Theory in Industry – Report of the Control Technology. Survey in Japanese Industry. IEEE Trans. on Control Systems Technology, vol. 7, No.3, May 1999, p. 298-305.

320. Takatsu H., Itoh T. Future needs for control theory in industry-report of the control technology survey in Japanese industry. – IEEE Transactions on Control Systems Technology, vol. 7, Issue 3, May 1999, p. 298 – 305.
321. Tamayo E. Industrial implementation of a predictive IMC controller. – IEEE Canadian Conference on Electrical and Computer Engineering, 1999, vol. 3, 1999, p. 1565 – 1569.
322. Tan Y., Dang X., van Cauwenberghe, A. Generalised nonlinear PID controller based on neural networks. – Information, Decision and Control, 1999. IDC 99. Proceedings. 1999, p. 519 – 524.
323. Tan Y., Dang X., van Cauwenberghe A. A neural network based nonlinear PID controller using PID gradient training. – Proceedings of the 1999 IEEE International Symposium on Intelligent Control/Intelligent Systems and Semiotics, 1999, p. 29 – 33.
324. Tang K.S., Man K.F., Chen G., Kwong S. An optimal fuzzy PID controller. – IEEE Transactions on Industrial Electronics, vol. 48, Issue 4, Aug 2001, p. 757 – 765.
325. Techmation. Protuner. Application manual. www.protuner.com.
326. Temperature -Electromotive Force Reference Functions and Tables for the Letter-Designated Thermocouple Types Based on the ITS-90. Nat. Inst. Stand. Technol. Monograph 175; 1993. – 630 p.
327. Ten ways to bulletproof RS-485 interface. – National Semiconductor Application Note 1057, Oct. 1996. – 10 p.
328. Teng J.-H., Lee R., Liu Y.-H., Leou R.-C. Integration of Internet and virtual instruments to develop an industrial SCADA. – IEEE Power Engineering Society Summer Meeting, 2001, Vol. 3, 2001, p. 1509 – 1514.
329. The Patent Office – <http://gb.espacenet.com>.
330. Thomesse J.-P. Fieldbus Technology in Industrial Automation.- Proc. of the IEEE, Vol. 93, No. 6, June 2005, p. 1073 – 1101.

331. TIA/EIA Telecommunications System Bulletin TSB89, Application Guidelines for TIA/EIA-485-A, Telecommunications Industry Association, June 1998.
332. TIA/EIA standard TIA/EIA-485-A, Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems, Telecommunications Industry Association, March 3, 1998.
333. Tsai P.-Y., Huang H.-C., Chuang S.-J., Chen Y.-J., Hwang R.-C. The model reference control by adaptive PID-like fuzzy-neural controller. – 2005 IEEE International Conference on Systems, Man and Cybernetics, vol. 1, 10-12 Oct. 2005, p. 239 – 244.
334. Vanzura , E.J. Automated system for electromagnetic field generation and immunity testing. – 5th IEEE Instrumentation and Measurement Technology Conference, 1988. IMTC-88, 20-22 Apr 1988, p. 3 – 10.
335. Varol H.A., Bingul Z. A new PID tuning technique using ant algorithm. – Proceedings of the 2004 American Control Conference, vol. 3, 30 June-2 July 2004, p. 2154 – 2159.
336. Vermeulen H.J., Strauss J.M. Off-line identification of an open-loop automatic voltage regulator using pseudo-random binary sequence perturbations. – AFRICON, 1999 IEEE. vol. 2, 1999, p. 799 – 802 vol.2.
337. Vieira M.A.M., Junior D.C.S. Survey on wireless sensor network devices. IEEE Conference «Emerging Technologies and Factory Automation», 2003. Proceedings ETFA '03. Vol. 1, 16-19 Sept. 2003, p. 537 – 544.
338. Vieira J., Mota A. Smith predictor based neural fuzzy controller applied in a water gas heater that presents a large time-delay and load disturbances. – Proceedings of 2003 IEEE Conference on Control Applications, CCA 2003, vol. 1, 23-25 June 2003, p. 362 – 367.
339. Vijayaraghavan G., Brown M., Barnes M. Practical grounding, bonding, shielding, and surge protection. – Elsevier, Newnes, 2004. p. 237.

340. Vinsonneau B., Goodall D.P., Burnham K.J., Brie D. Combined A Priori Knowledge And Structural Model Closed-Loop Identification Of High Temperature Industrial Furnaces. – 44th IEEE Conference on Decision and Control, and European Control Conference CDC-ECC '05, 2005, p. 8138 – 8143.
341. Visioli A. Fuzzy logic based set-point weight tuning of PID controllers. – IEEE Transactions on Systems, Man and Cybernetics, Part A, vol. 29, Issue 6, Nov 1999, p. 587 – 592.
342. Visioli A. Model-based PID tuning for high-order processes: when to approximate. – IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 12-15 Dec. 2005, p. 7127 – 7132.
343. Vyatkin V., Hirsch M., Hanisch H.-M. Systematic Design and Implementation of Distributed Controllers in Industrial Automation. – IEEE Conference on Emerging Technologies and Factory Automation, 2006. ETFA '06, 20-22 Sept. 2006, p. 633 – 640.
344. Wang Q.-G., Hang C.-C., Bi Q. A technique for frequency response identification from relay feedback. – IEEE Transactions on Control Systems Technology, vol. 7, Issue 1, Jan 1999, p. 122 – 128.
345. Wang D., Johnson C.D. A comparison study of some impulse-response identification methods. – Proceedings of Twenty-First Southeastern Symposium on System Theory, 1989, 26-28 Mar 1989, p. 52 – 56.
346. Wang P., Kwok D.P. Optimal fuzzy PID control based on genetic algorithm. – Proceedings of the 1992 International Conference on Industrial Electronics, Control, Instrumentation, and Automation, 1992. 'Power Electronics and Motion Control', 9-13 Nov 1992, p. 977 – 981.
347. Wang L., Tan K.C. Modern Industrial Automation Software Design. Principles and Real-World Applications. – John Wiley & Sons Inc., Hoboken, New Jersey. 2006. – 313 p.

348. Wang L. Open architecture-based factory automation systems. *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, No. 2, Feb. 2004, p. 14 – 17.
349. Weaver A.C. Factory monitoring and control using the internet. – *IECCON'01: The 27th Annual Conf. of the IEEE Industrial Electronic Society*, 2001, p. 1639 – 1645.
350. Webb J.W., Reis R.A. Programmable logic controllers. Principle and applications. – 5th Ed., Pearson Education Inc., Upper Saddle River, New Jersey, 2003, 460 p.
351. Wei T., Songjiao S., Mengxiao W. Model identification and PID control of long time-delay processes. – *Proceedings of the 4th World Congress on Intelligent Control and Automation*, 2002, vol. 2, 2002, p. 900 – 904.
352. Wei T., Songjiao S., Mengxiao W.. A novel Smith predictor with double-controller structure. – *Proceedings of the 2002 American Control Conference*, vol. 6, 2002, p. 5074 – 5079.
353. Wells C.H., Larson R.E. Application of combined optimum control and estimation theory to direct digital control. – *Proceedings of the IEEE*. vol. 58, Issue 1, Jan. 1970, p.16 – 22.
354. Wiberg P.-A., Bilstrup U. Wireless technology in industry – Applications and user scenarios. – *Proc. IEEE Int. Conf. Emerging Technologies and Factory Automation (ETFA '01)*, pp. 123–133.
355. Wikipedia . – <http://en.wikipedia.org>.
356. Willig A.; Matheus, K.; Wolisz, A. Wireless technology in industrial networks. – *Proc. of the IEEE*, Vol. 93, Issue 6, June 2005, p. 1130 – 1151.
357. Willig , A. Redundancy Concepts to Increase Transmission Reliability in Wireless Industrial LANs. – *IEEE Transactions on Industrial Informatics*, vol. 1, № 3, 2005, p. 173 – 182.
358. Wolischlaeger M.; Neumann, P.; Bangemann, T. Web services for remote maintenance of fieldbus based automation systems. *IEEE AFRICON. 6th Africon Conference in Africa*, 2002, vol. 1, 2-4 Oct. 2002, p. 247 – 252.

359. Wu S., Watanabe K., Muramatsu E., Ariga Y., Endo S. Robust stability of approximate Smith predictor control systems. – SICE 2004 Annual Conference. vol. 2, 4-6 Aug. 2004, p. 1522 – 1527.
360. Wu H., Ming J., Yang Y., Zhu S. Integrating embedded-Web technology and real-time Ethernet for modern distributed control. – Fifth World Congress on Intelligent Control and Automation, 2004. WCICA 2004. Vol. 2, June 15-19, 2004, p. 1323 – 1325.
361. Xi B., Fang Y., Chen M., Liu J. Use of Ethernet for Industrial Control Networks. – ICIEA – 1st IEEE Conference on Industrial Electronics and Applications, May 2006. – p. 1 – 4.
362. Xia F., Yin H., Wang Z., Sun Y. Function block oriented architecture for open distributed automation. – Fifth World Congress on Intelligent Control and Automation, vol. 3, June 15-19, 2004, p. 2706 – 2710.
363. Xu J., Feng X. Design of adaptive fuzzy PID tuner using optimization method. – Fifth World Congress on Intelligent Control and Automation, 2004. WCICA 2004, vol. 3, 15-19 June 2004, p. 2454 – 2458.
364. Yamamoto T., Kaneda M., Oki T., Watanabe E., Tanaka K. Intelligent tuning PID controllers. – IEEE International Conference on Systems, Man and Cybernetics, 1995. ‘Intelligent Systems for the 21st Century’, vol. 3, 22-25 Oct. 1995, p. 2610 – 2615.
365. Yamamoto T., Fujii K., Kaneda M. A self-tuning PID controller and its application for an ethylene cracking furnace. – 1996 IEEE Conference on Emerging Technologies and Factory Automation, EFTA ‘96. Proceedings, vol. 1, 18-21 Nov 1996, p. 275 – 281.
366. Yang T.-M., Xiong S.-B. Neural network based online self-learning adaptive PID control for automatic ranging cutting height of shearer. – Proc. 2002 International Conference on Machine Learning and Cybernetics, vol. 2, 2002, p. 857 – 859.
367. Yang P., Peng D.-G., Yang Y.-H., Wang Z.-P.. Neural networks internal model control for water level of boiler drum in power station. – Proceedings of 2004 International Conference

- on Machine Learning and Cybernetics, vol. 5, 26-29 Aug. 2004, p.3300 – 3303.
368. Yang Y., Xue Y., Huang J.. Simultaneous PID self-tuning and control. – Proceedings of the 2004 IEEE International Symposium on Intelligent Control, 2-4 Sept. 2004, p.363 – 367.
369. Yang Q., Xue Y., Wu Q.. Stabilization and robust control of a class of processes with long time delay. – 30th Annual Conference of IEEE Industrial Electronics Society, IECON, vol. 2, 2-6 Nov. 2004, p.1304 – 1308.
370. Yang Y., Wang W.-G., Yu D.-J., Ding G.. A fuzzy parameters adaptive PID controller design of digital positional servo system. – 2002 International Conference on Machine Learning and Cybernetics, vol. 1, 2002, p.310 – 314.
371. Yang S.H., Tan L.S., Chen X. Requirements specification and architecture design for Internet-based control systems. – Proceedings. 26th Annual International Computer Software and Applications Conference, 2002. COMPSAC 2002, 26-29 Aug. 2002, p. 75 – 80.
372. Yao L., Lin C.-C.. Design of a self tuning fuzzy PID controller by the accumulated genetic algorithm. – 2002 IEEE International Conference on Industrial Technology, 2002. IEEE ICIT '02, vol. 1, 2002, p.649 – 654.
373. Yesil E.; Guzelkaya M.; Eksin I. Internal model control based fuzzy gain scheduling technique of pid controllers. – World Automation Congress, 2004. Proceedings. Vol. 17, 28 June – 1 July 2004, p.501 – 506.
374. Yongquan Y., Ying H., Bi Z.. A PID neural network controller. – Proceedings of the International Joint Conference on Neural Networks, vol. 3, 20-24 July 2003, p. 1933 – 1938.
375. Yongquan Y., Ying H., Tao Z. To tune the dynamic parameters of neural PID controller by the intelligent learning algorithm. – 2005 IEEE International Conference on Mechatronics and Automation, vol. 3, 29 July-1 Aug. 2005, p. 1521 – 1526.
376. Yongquan Y., Ying H., Bi Z.. The dynamic fuzzy method to tune the weight factors of neural fuzzy PID controller. – 2004

- IEEE International Joint Conference on Neural Networks, 2004, vol. 3, 25-29 July 2004, p. 2397 – 2402.
377. Yongquan Y., Ying H., Minghui W., Bi Z., Guokun Z.. Fuzzy neural PID controller and tuning its weight factors using genetic algorithm based on different location crossover. – 2004 IEEE International Conference on Systems, Man and Cybernetics, vol. 4, 10-13 Oct. 2004, p.3709 – 3713.
378. Yourui H., Xiuqiao C., Yihua H. Optimization for Parameter of PID Based on DNA Genetic Algorithm. – International Conference on Neural Networks and Brain, 2005. ICNN&B '05, vol. 2, 13-15 Oct. 2005, p. 859 – 861.
379. Yu W.-S., Lu T.-C. PID controller design using dynamical neural networks. Neural Networks Proceedings. – IEEE World Congress on Computational Intelligence6 vol. 3, 4-9 May 1998, p. 2131 – 2135.
380. Yukitomo M., Baba Y., Shigemasa T., Ogawa M., Akamatsu K., Amano S. A model driven PID control system and its application to chemical processes. – Proceedings of the 41st SICE Annual Conference, vol. 4, 5-7 Aug. 2002, p. 2656 – 2660.
381. Yukitomo M., Shigemasa T., Baba Y., Kojima F. A two degrees of freedom PID control system, its features and applications. – 5th Asian Control Conference, vol. 1, 20-23 July 2004, p.456 – 459.
382. Zadeh L.A. Fuzzy sets. – Information and Control. 1965, №8, p.338-353.
383. Zhang Y., Chen Z.-Q., Yang P., Yuan Z.-Z.. Neural network-based PID predictive control for nonlinear time-delay systems. – Proceedings of 2004 International Conference on Machine Learning and Cybernetics, vol. 2, 26-29 Aug. 2004, p. 1014 – 1018.
384. Zhang J., Yu D., Qi S. Structural research of fuzzy PID controllers. – International Conference on Control and Automation, 2005. ICCA '05, vol. 2, 26-29 June 2005, p. 1248 – 1253.
385. Zheng W.X. A modified method for closed-loop identification of transfer function models. – IEEE Transactions on Circuits

- and Systems I: Fundamental Theory and Applications, vol. 49, Issue 4, Apr 2002, p. 556 – 562.
386. Zhixun X.; Yuejin H. Power System Technology. – Proceedings of 2002 Int. Conf. on PowerCon, vol. 4, 2002, p. 2448 – 2451.
 387. Zhu G., Skelton R.E., Li P.. Q-Markov cover identification using pseudo random binary signals. American Control Conference, 1994. vol. 1, 29 June-1 July 1994, p. 515 – 519.
 388. Zhu Y., Liu X., Qian J.. Adaptive predictive PID control scheme of ideal internal thermally coupled distillation columns. – Fifth World Congress on Intelligent Control and Automation, 2004. WCICA 2004, vol. 4, 15-19 June 2004, p. 3441 – 3444.
 389. Ziegler J. G., Nichols N. B. Optimum settings for automatic controllers. – Trans. ASME, vol. 64, p. 759–768, 1942.
 390. ZigBee specification. Document 053474r13. – ZigBee Standards Organization, Dec. 1, 2006. – 534 p.
 391. Zipse D.W., Earthing-Grounding methods: a primer. – Industrial and Commercial Power Systems Technical Conference, 2002, May 5-8, 2002 p. 158 – 177.
 392. Ziyuan H., Minrui F., Zhiqiang H.. Smith online identification predictive control and its application in STM. – Proceedings of the 4th World Congress on Intelligent Control and Automation, vol. 3, 2002, p. 2254 – 2257.
 393. Zoitl A., Grabmair G., Auinger F., Sunder C. Executing real-time constrained control applications modeled in IEC 61499 with respect to dynamic reconfiguration. – 3rd IEEE International Conference on Industrial Informatics, 2005. INDIN '05, 10-12 Aug. 2005, p. 62 – 67.
 394. Zoitl A., Sunder C., Terzic I. Dynamic Reconfiguration of Distributed Control Applications with Reconfiguration Services based on IEC 61499. – IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications, 15-16 June 2006. p. 109 – 114.
 395. Zoitl A., Smodic R., Sunder C., Grabmair G. Enhanced real-time execution of modular control software based on IEC

61499. – Proceedings 2006 IEEE International Conference on Robotics and Automation, May 15-19, 2006, p. 327 – 332.
396. Аблин И.Е. С MasterSCADA – шаг за шагом. – «Промышленные АСУ и контроллеры», №10, 2003, с. 4 – 7.
397. Александровская Л.Н., Афанасьев А.П., Лисов А.А. Современные методы обеспечения безотказности сложных технических систем. – М.: Логос, 2001. – 206 с.
398. Анашкин А.С., Кадыров Э.Д., Харазов В.Г. Техническое и программное обеспечение распределенных систем управления. Под редакцией Харазова В.Г. – Санкт-Петербург: Изд-во «Р-2», 2004. – 367 с.
399. Бабенко Ю., Денисенко В. Цифровая система термометрии элеватора «Грейн». – Хлебопродукты, №11, 2007. – с. 46-48.
400. Бажанов В.Л. Универсальный микропроцессорный регулятор с USWO алгоритмом управления. – Приборы и системы. Управление, контроль, диагностика, 2000, № 1, с. 35 – 39.
401. Бажанов В.Л., Вайшнарас А.В. Программа «ММ-настройка» для определения параметров ПИД-регуляторов по методу масштабирования. – Автоматизация в промышленности», №6, 2007. – с. 60-64.
402. Барнс Д. Электронное конструирование: методы борьбы с помехами. М.: Мир, 1990, 239 с.
403. Баскаков С.И. Радиотехнические цепи и сигналы. М.: Высшая школа, 1983. – 536 с.
404. Белевцев А.В., Богатов В.В., Каржавин А.В., Петров Д.В., Улановский А.А. Термоэлектрические преобразователи температуры. Теория, практика, развитие (обзор). – Обнинск, ПК «Тесей», 2003 г. – 57 с.
405. Бессонов Л.А. Теоретические основы электротехники. М.: Высшая школа, 1973. – 752 с.
406. Богданов С.В. Умный дом. – Наука и техника, Санкт-Петербург, 2003, 112 с.

407. Боломытцев В. Замена элементов управляющей вычислительной системы без отключения питания. – СТА, №2, 2000, с. 72-77.
408. Букингем М. Шумы в электронных приборах и системах. – М.: Мир, 1986. – 399 с.
409. Бурдонов И.Б., Косачев А.С., Пономаренко В.Н. Операционные системы реального времени. Препринт Института системного программирования РАН. – www.citforum.ru.
410. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++, 2-е изд./Пер. с англ. – М.: «Издательство Бином», СПб.: «Невский диалект», 2000. – 560 с.
411. Буч Г. Язык UML. Руководство пользователя/ Грейди Буч, Джеймс Рамбо, Айвар Джекобсон: Пер. с англ. Слинкин А.А. – 2-е изд., М.: ДМК Пресс, Питер, 2004. – 432 с.
412. Воронов А.А. Устойчивость, управляемость, наблюдаемость. М.: Наука, 1979, 336 с.
413. Гарманов А. Принципы обеспечения электросовместимости измерительных приборов. – СТА, №4, 2003, с. 64-72 и №1, 2004 г., с. 62-68.
414. Гельднер К., Кубик С. Нелинейные системы управления. М.: Мир, 1987, 368 с.
415. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. – Москва, Физматлит, 2006, 320 с.
416. Гмурман В.Е. Теория вероятностей и математическая статистика. – М.: Высшая школа, 2001. – 479 с.
417. Грибанов М.В., Калачев Д.П., Третьяков С.А. CAL – CAN протокол прикладного уровня для промышленных приложений. – МКА, №4, 1999.
418. Гроп Д. Методы идентификации систем. М.: Мир, 1979, 302 с.
419. Гулько С.В., Джоврей Н. Обзор стандарта IEC 61499. – ПиКАД, №4, 2005 г.

420. Ивѣн Б., Берес Д. Visual Basic .NET. Библия пользователя.: Пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 1024 с.
421. Изерман Р. Цифровые системы управления. М.: Мир, 1984, 541 с.
422. Ицкович Э.Л. Современные алгоритмы автоматического регулирования и их использование на предприятиях. – Автоматизация в промышленности, №6. 2007. – с. 39-44.
423. Ицкович Э.Л. Конкурентоспособность российских производителей контроллеров на рынке средств автоматизации производства. – Промышленные контроллеры АСУ, №2. 2008. – с. 4 – 10.
424. Кисилев В. Промышленный Ethernet в стиле Hirschmann. – СТА, №2, 2005, с. 6-12.
425. Клир Жд. Системология. Автоматизация решения системных задач. М.: Радио и связь, 1990. – 544 с.
426. Корнеев В.В., Киселев А.В. Современные микропроцессоры. – 3-е изд. – СПб.: БХВ-Петербург, 2003. – 448 с.
427. Косарев Е.Л. Методы обработки экспериментальных данных. М.: 2003. – 256 с.
428. Кругляк К. Локальные сети Ethernet в АСУ ТП : быстрее, дальше, надежнее. – СТА, №1, 2003. – с. 6 – 13.
429. Кузнецов Р.Г. Кабели для современных сетей промышленной автоматизации. – Автоматизация в промышленности, №8, 2005.
430. Куцевич И.В. Системный интегратор в современном мире систем управления. – МКА, №3, 2004, с. 76-84.
431. Липаев В.В. Технологические процессы и стандарты обеспечения функциональной безопасности в жизненном цикле программных средств. М.: Jet Info №8(135), 2004.
432. Липаев В.В. Функциональная безопасность программных средств. – М.: Синтег, 2004. – 340 с.
433. Ротач В.Я. Теория автоматического управления. М.: МЭИ, 2004, 400 с.

434. Ротач В.Я. Настройка регуляторов модифицированным методом Циглера-Николса. – Промышленные контроллеры АСУ, №2, 2008. – с. 38-42.
435. Ротач В.Я. Об адаптивных системах управления с текущей идентификацией объекта.
436. Ротач В.Я. К расчету оптимальных параметров реальных ПИД-регуляторов по экспертным критериям// Промышленные АСУ и контроллеры. №2, 2006. с. 22-29. В. Надежность программных средств. – Синтез, 1998 г. 232 стр.
437. Фрайден Дж. Современные датчики. Справочник. – М.: Техносфера, 2005. – 592 с.
438. Хазарадзе Т.О., Куликов А.И. Построение масштабных АСУ ТП: опыт решения проблемы. – Мир компьютерной автоматизации, №5, 2002, с. 37-45.
439. Шехтман М.Б., Цесарев И.М., Прошин А.И. Крупномасштабная распределенная система управления газофракционирующей установки. – Промышленные АСУ и контроллеры, 2006, №5. с. 12 – 16.
440. Штейнберг Ш.Е. Идентификация в системах управления. М.: Энергоатомиздат, 1987, 80 с.
441. Штейнберг Ш.Е., Сережин Л.П., Залуцкий И.Е., Варламов И.Г. Проблемы создания и эксплуатации эффективных систем регулирования //Промышленные АСУ и контроллеры. 2004. №7. с. 1-7.
442. Шубладзе А.М., Гуляев С.В., Малахов В.А., Ольшванг В.Р., Кузнецов С.И. Автоматическая настройка и адаптация в промышленных ПИД регуляторах. – Приборы и системы. Управление, контроль, диагностика. №13, 2007, с. 26 – 30.
443. Шубладзе А.М., Гуляев С.В., Шубладзе А.А. Оптимальные автоматически настраиваемые общепромышленные регуляторы// Приборы и системы. Управление, контроль, диагностика. 2002. №10. с. 30-33.

444. Эрглис К.Э. Интерфейсы открытых систем. – М.: «Горячая линия – Телеком», 2000. – 256 с.

Опис основних елементів інтерфейсу

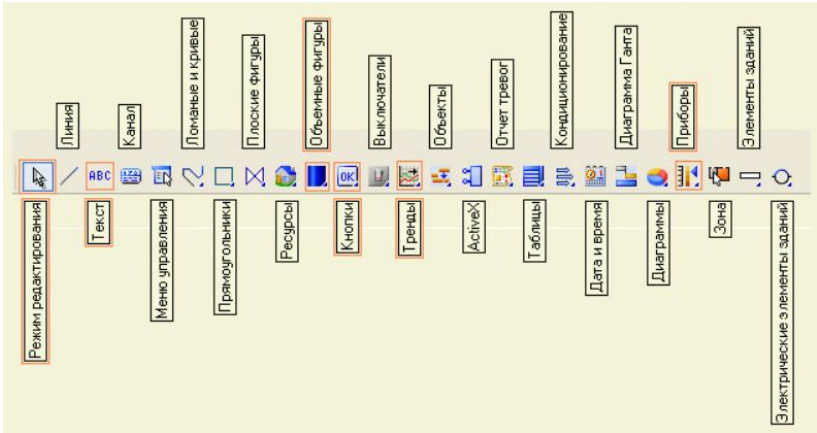


Рисунок А.1 – Графічні інструменти редактора TRACE MODE

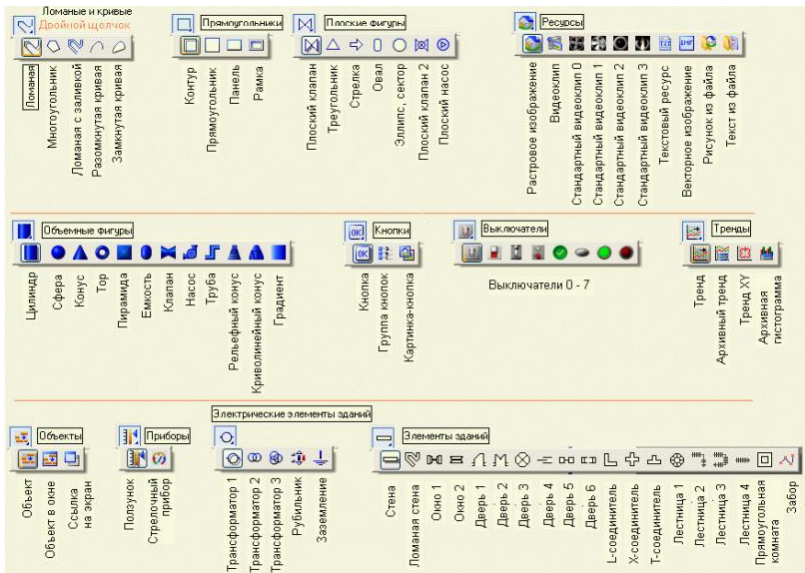


Рисунок А.2 – Графічні елементи редактора

Арифметичні FBD блоки

Додаток Б

Таблиця Б

Арифметичні FBD блоки

Назва	Позначення	Дія	Входи та виходи
1	2	3	4
Додавання	$X+Y$	Додавання X і Y	X, Y – доданок, O – сума
Додавання чотирьох елементів	$+$	Додавання A, B, C, D	A, B, C, D – доданок, O – сума
Віднімання	$X-Y$	Віднімання Y з X	X – зменшуване, Y – від’ємник, O – різниця
Множення	$X*Y$	Множення X на Y	X, Y – множники, O – добуток
Ділення	X/Y	Ділення X на Y	X – ділене, Y – дільник, O – часте
Залишок від ділення	$X\%Y$	Залишок від ділення X на Y	X – ділене, Y – дільник, O – залишок від ділення
Зведення в ступінь	$X**Y$	Зведення X в ступінь Y	X – основа, Y – показник ступеня, O – результат зведення в ступінь
Абсолютне значення	ABS	Повертає абсолютне значення	X – вхід, O – вихід
Інверсія знака	$-X$	Змінює знак вхідного значення	X – вхід, O – вихід
Ціла частина	FLOOR	Повертає цілу частину числа	X – вихідне число, O – ціла частина
Зворотня величина	$1/X$	Обчислює зворотню величину: $O = 1/X$	X – вхідне значення, O – зворот-на

			величина
--	--	--	----------

Продовження табл. Б

1	2	3	4
Квадратний корінь	SQRT	Витягує квадратний корінь X	X – вхідне значення, O – квадратний корінь з X
Зведення в квадрат	X**2	Зведення X в квадрат	X – вхідне значення, O – квадрат X
Сума квадратів	HYPOТ	Обчислює суму квадратів	X, Y – вхідні значення, O – сума квадратів
Масштабування	K*X+C	Виконує масштабування вхідного значення	X – вхідне значення, K – множник, C – зміщення

Додаток В

FBD блоки порівняння

Таблиця В

Назва	Позначення	Опис	Входи та виходи
1	2	3	4
Рівність	==	Повертає 1, якщо IN1 дорівнює IN2, в іншому випадку повертає 0	IN1, IN2 – значення, що порівнюються, EQ – результат порівняння
Нерівність	<>	Повертає 1, якщо IN1 НЕ дорівнює IN2, в іншому випадку повертає 0	IN1, IN2 – значення, що порівнюються, NE – результат порівняння
Більше	>	Повертає 1, якщо IN1 більше IN2, в іншому випадку повертає 0	IN1, IN2 – значення, що порівнюються, GT – результат порівняння
Менше	<	Повертає 1, якщо IN1 менше IN2, в іншому випадку повертає 0	IN1, IN2 – значення, що порівнюються, LT – результат порівняння
Більше або дорівнює	>=	Повертає 1, якщо IN1 не менш IN2, в іншому випадку повертає 0	IN1, IN2 – значення, що порівнюються, GE – результат порівняння
Менше або дорівнює	<=	Повертає 1, якщо IN1 не більш IN2, в іншому випадку повертає 0	IN1, IN2 – значення, що порівнюються, LE – результат порівняння
Рівність нулю	==0	Повертає 1, якщо INP дорівнює 0, в іншому випадку повертає 0	INP – значення, що порівнюються, ==0 – результат порівняння
Нерівність нулю	<>0	Повертає 1, якщо INP не дорівнює 0, в іншому випадку повертає 0	INP – значення, що порівнюються, <>0 – результат порівняння

Продовження табл. В

1	2	3	4
Знакова функція	SIGN	Якщо $INP > 0$, то $SN+=1$, а $SN-=0$; якщо $INP < 0$, то $SN+=0$, а $SN-=1$. Если $INP = 0$, то $SN+ = SN- = 0$.	INP – вхідне значення, $SN+$ і $SN-$ – вихідне значення
Аналіз на рівність	CMР	Порівнюється вхід INP з входами PDK, повертається найменший з номерів входів PDK, чиї значення рівні INP, якщо всі значення PDK нерівні INP, то значення Q не змінюється.	INP, PDK – входи, Q – вихід
Аналіз збігів	CMPN	Якщо INP не дорівнює Pd, то при кожному виклику QT збільшується на 1, якщо $QT > PT$, то QE повертає 1, збільшення QT припиняється навіть при INP нерівному PD. Виходи QT, QE приймає 0 при будь-якій зміні INP.	INP, PD, PT – входи, QT, QE – виходи
Аналіз збігів	CMPE	Відрізняється від CMPN аналізом збігів INP з PD, скиданням QT і QE в 0 при будь-якій зміні як INP, так і PD	INP, PD, PT – входи, QT, QE – виходи

Продовження табл. В

1	2	3	4
Управління по астрономічному у часу	ALARM	Формує 1 при збігу поточного астрономічного часу з заданим на входах	MON – місяць; DAY – день місяця; DOF – день тижня; H – години; M – хвилини; S – секунди

Додаток Г

FBD блоки вибору

Таблиця Г

Назва	Позначення	Дія	Входи та виходи
1	2	3	4
Вибір з двох	SEL	Обирає IN0, якщо IG= 0 і IN1, якщо IG= 1	IG – критерій вибору, IN0, IN1 – значення, SEL – результат вибору
Вибір з п'яти	nSEL	Повертає значення одного з входів і номер входу. Якщо REG = 0 повертається мінімальне значення, якщо REG = 1- найбільше. Перші п'ять бітів числа BLK визначають участь входів виборі. Якщо біт дорівнює 0, то відповідний йому за номером вхід розглядається, якщо 1 – ігнорується	REG – вхід, що визначає умову вибору, BLK – вхід, що визначає участь входів I0...I5 в виборі; I0...I5 – входи для вибору; VAL – обране значення; NUM – номер обраного каналу
Вибір максимального	MAX	Повертає найбільше значення з вхідних	IN1, IN2 – вхідні значення, MAX – максимальне з вхідних значень
Вибір мінімального	MIN	Повертає найменше значення з вхідних	IN1, IN2 – вхідні значення, MIN – мінімальне з вхідних значень

Продовження табл. Г

1	2	3	4
Обмеження	LIMIT	Кліпує вхідний сигнал	INP – вхідний сигнал вхід MAX-максимальне значення вхідного сигналу, вхід MIN-мінімальне значення вхідного сигналу
Вибір з трьох	MUX	Повертає INNUM, якщо NUM = 0,1,2 в іншому випадку вихід Q не змінюється	Вхід NUM-номер входу для вибору; IN0...IN2 – входи для вибору; Q – результат вибору
Вибір з семи	MUX7	Аналогічний MUX, відрізняється кількістю входів IN	Вхід NUM – номер входу для вибору; IN0 ... IN6 – входи для вибору; Q – результат вибору
Інтервал	NLIM	Повертає 1, якщо $INP > MAX$; повертає 0, якщо $MIN \leq INP$; повертає 2, якщо $INP < MIN$	Вхід MIN – мінімальне значення; вхід INP – вхідне значення; вхід MAX – максимальне значення
Запізнення	LTN	Реалізація запізнювання: $Q0i = INPi$; $Q1i = INPi-1$; $Q2i = INPi-2$; $Q3i = INPi-3$, де i – номер поточного виклику блоку	INP – вхід; Q0...Q3 – затриманий сигнал INP

Продовження табл. Г

1	2	3	4
Передбачення	FRWD	Реалізує екстраполяцію вхідного значення по першій і другій похідним, виходу Q присвоюється передбачуване значення INP при наступному виклику блоку	INP – вхід; Q – передбачуване значення, яке надходить на вхід блоку при наступному виклику

Додаток Д

FBD блоки – генератори

Таблиця Д

Назва	Позначення	Дія	Входи та виходи
1	2	3	4
Меандр	G01	Генерує прямокутний сигнал з максимальним значенням рівним 1	1/0 – результат генерації
Одиниця, що переміщується	G1	8 – бітовий вихід приймає послідовно значення: 0, 20, 21, 22,...,27, 26,..., 20, 21 і т. д.	Q – результат генерації сигналу
Випадкова величина в діапазоні [0;1]	RND	Генерує випадкову величину в діапазоні [0; 1] з нормальним законом розподілу	Q – результат генерації сигналу
Пилкоподібний сигнал	PILA	Генерується пилкоподібний сигнал з максимальним значенням PV	PV – максимальне значення; Q – результат генерації сигналу
Одиниця з заданою вірогідністю	GP01	Генеруються 0 і 1 з імовірністю генерації 1 рівній PRB. PRB – ціле число в діапазоні від 0 до 1000	PRB – ймовірність генерації 1; 0/1 – результат генерації
Астрономічний час	TIME	Повертає поточний астрономічний час	S – секунди; M – хвилини; H – години
Астрономічна дата	DATE	Повертає поточне значення дати	DAY – день місяця; MON – місяць;

			YR – поточний рік
Період виклику програми	TSTEP	Вимірює період виклику програми в мілісекундах	ПМ – період виклику програми

Продовження табл. Г

1	2	3	4
Синусоїдальний сигнал	GSIN	Генерує синусоїдальний сигнал одиничної амплітуди з періодом коливачь РТ (кількість викликів блоку, які припадають на один період)	РТ– період коливачь; SIN– результат генерації

НАУКОВЕ ВИДАННЯ

Невлюдов І.Ш., Андрусевич А.О., Євсєєв В.В.,
Максимова С.С., Стародубцев М.Г., Невлюдова В.В.

**Автоматизована система керування
технологічними процесами в SCADA системі
TRACE MODE 6**

*Рекомендовано Вченою радою
Харківського національного університету радіоелектроніки*

Коректор В.Г. Андріященко

Комп'ютерна верстка О.О. Роголюк

Підписано до друку 11.10.2018 р. Формат 60×84 1/16.
Папір офсет. Гарнітура Таймс 20,0 Умов. Друк. Арк..
14,7 Умов. вид. арк. Тираж 300 прим. Зам. № 97.

Видано на замовлення ХНУРЕ