



Національний університет
водного господарства та
природокористування

Міністерство освіти і науки України
Національний університет водного господарства та
природокористування
Кафедра автоматизації, електротехнічних та комп'ютерно-
інтегрованих технологій

04-03-97

Методичні вказівки

до виконання лабораторних робіт № 11-15
з дисципліни „Автоматизація технологічних процесів і
виробництв” студентами за напрямом підготовки
6.050202 "Автоматизація та комп'ютерно-інтегровані
технології" денної та заочної форм навчання

Рекомендовано методичною
комісією за напрямом
підготовки 6.050202
"Автоматизація та
комп'ютерно-інтегровані
технології"
Протокол № 7 від 21.03.2014р.

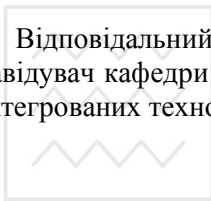
Рівне 2014



Методичні вказівки до виконання лабораторних робіт № 11 – 15 з дисципліни „Автоматизація технологічних процесів і виробництв” студентами за напрямом підготовки 6.050202 "Автоматизація та комп'ютерно-інтегровані технології" денної та заочної форм навчання / М.М. Клепач, Д.Т. Реут – Рівне: НУВГП, 2014 – 49 с.

Упорядники: **М. М. Клепач** – старший викладач кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій,
Д. Т. Реут – асистент кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій,

Відповідальний за випуск **В.В. Древецький**, д.т.н., професор, завідувач кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій.





Робота 11. Розробка програмного забезпечення ПЛК автоматизації типового періодичного технологічного процесу в середовищі Codesys 2.3

11.1. Мета роботи

Ознайомитися з принципами реалізації програмного забезпечення автоматизованих систем управління об'єктами періодичної дії в середовищі розробки Codesys 2.3. Навчитись створювати типові програми реалізації управляючих функцій ПЛК.

11.2. Теоретичні відомості

11.2.1. Комплекс програмування ПЛК Codesys 2.3

Комплекс програмування ПЛК Codesys застосовується для програмування контролерів різних виробників, зокрема Moeller, Festo, Wago (Німеччина), Parker Hannifin (США), Mitsubishi Electric (Японія), АВВ (Швеція, Швейцарія), ОВЕН (Російська Федерація).

Комплекс Codesys включає власне Codesys як середовище розробки так і систему виконання Codesys SP (рис. 11.1).

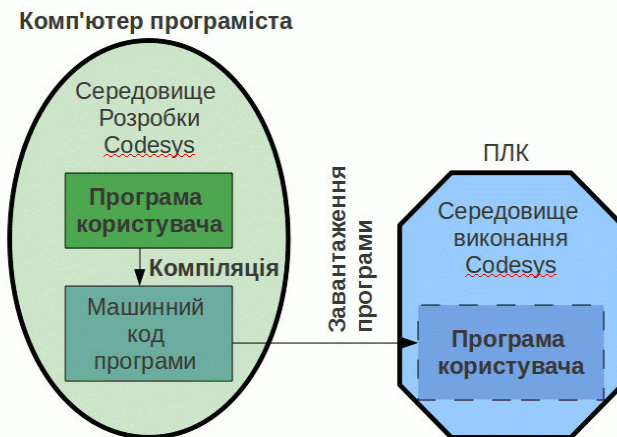


Рис. 11.1. Складові комплексу програмування Codesys



Вона виконує функції операційної системи, здійснюючи обслуговування входів, виходів, комунікаційних інтерфейсів, виклик задач користувацької програми та перемикання між ними, вирішення помилок, що виникають при роботі користувацької програми. За необхідності весь контроль за роботою ПЛК може бути переданий користувацькій програмі (обслуговування переривань, обробка помилок, керування планувальником задач і т.п.).

Для опису можливостей вводу-виводу конкретного ПЛК використовуються target-файл, який необхідно встановити в середовище розробки для програмування даного типу ПЛК. У цьому файлі описаний адресний простір вводу-виводу, вектори переривань та інша інформація, що характеризує апаратні можливості ПЛК.

Також до комплексу Codesys входить Codesys HMI, який дозволяє створювати нескладні графічні інтерфейси користувача, Codesys ENI Server, що дозволяє взаємодіяти з системами контролю версій, організувати багатокористувацький доступ до проекту, та Codesys SoftMotion — набір засобів для керування рухом, який дозволяє ПЛК виконувати роль блока ЧПУ. Дистрибутив також містить OPC-сервер, системи виконання Codesys SP RTE (багатозадачна система виконання жорсткого реального часу під Windows NT, 2000, XP) та Codesys SP PLCWinNT (м'якого реального часу, запускається як процес у ОС Windows на ядрі NT).

Codesys SP передає користувацькій програмі образ входів, а не поточний їх стан (рис.11.2).

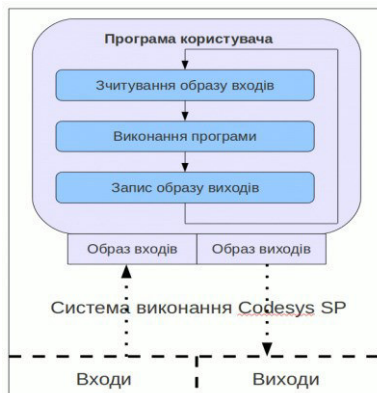


Рис. 11.2. Алгоритм роботи ПЛК з Codesys SP



Створення нового проекту

При виборі «Файл – Создать» отримуємо вікно вибору цільової платформи. Перелік доступних платформ залежить від встановлених target-файлів. Після вибору ПЛК необхідно створити програму PLC_PRG, що буде єдиною програмою в однозадачному проекті, аналогом функції *main()* в мові С.

Типи програмних компонентів (POU)

В Codesys є три типи програмних компонентів (POU):

1) функція – це POU, який повертає лише одне значення та не має внутрішньої пам'яті; ім'я функції використовується як вихідна змінна, якій присвоюється результат;

2) функціональний блок – це POU, який приймає та повертає вибране число значень. Фактично функціональний блок є аналогом класу (рис. 11.5.), а екземпляр блоку — аналогом об'єкту класу; кожен з екземплярів має власні вхідні, вихідні, внутрішні змінні, незалежні від інших екземплярів;

3) програма – це POU, який може формувати довільне значення під час виконання. Значення усіх змінних зберігаються між її викликами. Якщо програма, викликана з одного POU, змінить значення своїх змінних, то при наступному виклику з іншого POU ці змінні матимуть ті самі значення.

Для реалізації роботи програми за необхідним алгоритмом рекомендовано використовувати найменш складний POU: якщо для реалізації достатньо функції, то небажано використовувати для цього програму чи функціональний блок. Наприклад, якщо необхідно додати можливість ручного керування, то достатньо просто викликати у PLC_PRG по умові “біт ручного керування скинутий” функцію автоматичного керування. У випадку реалізації автоматичного керування не функцією, а програмою, на додавання можливості перемикання режимів керування знадобилося б значно більше змін в кодї програми та відповідно більше затраченого часу.

Типи даних

В Codesys підтримуються наступні типи даних:

- BOOL — логічний, може приймати 2 значення: TRUE/FALSE (істинно/хибно).
- Цілочисельні: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, які відрізняються діапазоном чисел, що зберігаються в них (табл.11.1).



Цілочисельні типи даних

Тип	Нижня межа	Верхня межа	Розмір пам'яті, біт
BYTE	0	255	8
WORD	0	65535	16
DWORD	0	4294967295	32
SINT	-128	127	8
USINT	0	255	8
INT	-32768	32767	16
UINT	0	65535	16
DINT	-2147483648	2147483647	32
UDINT	0	4294967295	32

Типи BYTE, WORD, DWORD використовуються, як правило, для бітових полів, а інші типи — власне для цілих чисел.

- Рациональні: REAL і LREAL, які займають відповідно 32 та 64 біти пам'яті та можуть зберігати число в діапазонах від $1.175494351e-38$ до $3.402823466e+38$ та від $2.2250738585072014e-308$ до $1.7976931348623158e+308$ відповідно.
- STRING — рядок символів; потребує вказання довжини рядка в дужках одразу після типу, наприклад

r:STRING(30):="рядок довжиною до 30 символів"

Якщо довжина не вказана, вона приймається рівною 80 символів. Довжина рядка не обмежена, проте стандартна бібліотека роботи з рядками може працювати лише з довжинами від 1 до 255 символів.

- TIME — містить інтервал часу в мілісекундах, максимальне значення — T#49d17h2m47s295ms.
- TOD (TIME_OF_DAY) — час доби, від TOD#00:00:00 до TOD#23:59:59.999.
- DATE — календарна дата, починаючи з 1 січня 1970 року.



Діапазон значень: від DATE#1970-00-00 до DATE#2106-02-06.

- DT (DATE_AND_TIME) — час в секундах, починаючи з 0 годин 1 січня 1970 року. Діапазон значень: від DT#1970-00-00-00:00:00 до DT#2106-02-06-06:28:15.

Також в Codesys доступні користувацькі типи даних, що формуються на основі вищеперерахованих елементарних типів: масив з можливістю контролю дотримання меж, вказівник з можливістю контролю виходу за допустиму область пам'яті, перелік, структура, псевдонім елементарного типу, елемент з обмеженим діапазоном значень.

У Codesys немає автоматичного перетворення типів. Усі операції перетворення повинні бути явно виконані функціями виду *INTYPE_TO_OUTTYPE*, де *INTYPE* – тип аргументу функції, *OUTTYPE* – тип значення, яке повертає функція. Наприклад,

```
string1:=REAL_TO_WORD(3.14)
```

Мови програмування

Codesys підтримує наступні мови програмування, стандартизовані в IEC 61131-3:

- Instruction List (IL) - список інструкцій, асемблероподібна мова;
- Structured Text (ST) - структурований текст, подібна до мови Паскаль високорівнева мова;
- Sequential Function Chart (SFC) - послідовні функціональні діаграми – графічна мова, описує послідовність станів та умов переходу;
- Function Block Diagram (FBD) - функціональні блокові діаграми, графічна мова, в якій програма створюється з'єднанням функціональних блоків, кожен з яких реалізовує певну підпрограму;
- Ladder Diagram (LD) - драбинкові діаграми, програмна реалізація релейно-контактних схем.

Також підтримується мова Continuous Function Chart (CFC), що відрізняється від стандартної FBD можливістю створення зворотних зв'язків та виконанням блоків у визначеному користувачем порядку.

За даними технічної підтримки виробника Codesys найбільше звертань надходить від користувачів мови ST (70%), по 30% - SFC



і FBD/CFC, по 10% - IL та LD. Відповідно можна судити про використання тієї чи іншої мови, зробивши поправку на те, що LD є більш вживаною серед досвідчених інженерів, які раніше працювали з релейно-контактними схемами й рідко звертаються за допомогою, тому відсоток використання LD насправді більший. Сумарний відсоток перевищує 100%, а отже частина користувачів використовують в проекті одночасно дві або більше мов.

Мови CFC та FBD зручно використовувати для реалізації нерозгалужених неперервних алгоритмів (різного типу регуляторів), SFC – для керування процесами, які відбуваються в декілька стадій, на кожній з яких алгоритм керування відрізняється, ST – для реалізації складних математичних операцій, циклів, роботи з масивами даних, символами, алгоритмів з великою кількістю розгалужень.

Оскільки мова CFC відрізняється від FBD можливістю вільного розміщення елементів, створення зворотніх зв'язків та вибору порядку виконання блоків, то зручнішою для початківця буде саме вона.

В Codesys існує асистент вводу, який можна викликати клавішею F2. У лівому списку відображаються категорії доступних для вставки в проект елементів, в правому – список або ієрархія (якщо встановлений прапорець “структурно”) елементів категорії. Прапорець “З аргументами” дозволяє здійснювати вставку елемента одразу з іменами аргументів, звільняючи програміста від необхідності вводити їх вручну. Результатом виконання вставки з аргументами функціонального блоку PID (рис. 11.3) наступний рядок:

```
PID( ACTUAL:= , SET_POINT:= , KP:= , TN:= , TV:= ,  
Y_MANUAL:= , Y_OFFSET:= , Y_MIN:= , Y_MAX:= , MANUAL:= ,  
RESET:= , Y=> , LIMITS_ACTIVE=> , OVERFLOW=> );
```

Для оголошення змінної в проекті немає необхідності записувати її вручну в розділі оголошень. Достатньо використати ім'я змінної в кодї й тоді при переході на наступний рядок буде викликано діалог проголошення змінної (рис. 11.4), в якому необхідно обрати клас змінної (звичайна змінна *VAR*, вхідна змінна *VAR_INPUT*, вихідна змінна *VAR_OUTPUT*, глобальна змінна *VAR_GLOBAL*), її ім'я та



тип. Також можна вказати початкове значення змінної та задати самостійно адресу пам'яті, яка буде виділена під змінну. Викликати діалог проголошення змінної можна також комбінацією клавіш *Shift+F2*.

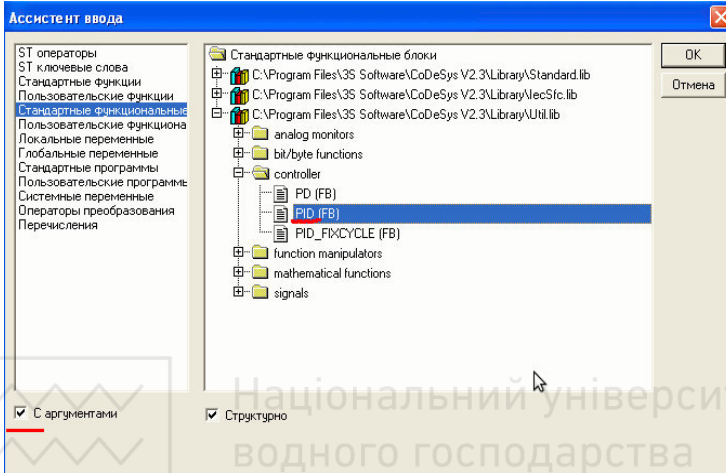


Рис.11. 3. Ассистент ввода

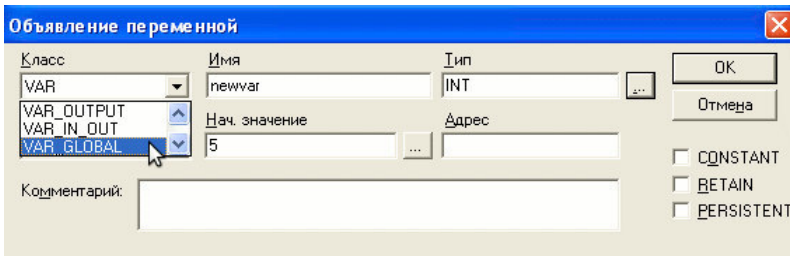


Рис. 11.4 Вікно діалогу оголошення змінної

Взаємодія ПЛК з об'єктом управління

Можливості вводу-виводу ПЛК описуються в target-файлі. За необхідності можна вибрати інший ПЛК зміною налаштувань цільової платформи на вкладці "Ресурси". За замовчуванням входи та виходи ПЛК адресуються виразами %AB.N, де А визначає напрям: вхід (I) або вихід (Q), В – тип змінної (R – дійсний, W –

двійкове слово, X – байт), N – номер біта. Але зазвичай програмісту зручніше працювати з осмисленими назвами змінних, тому в вікні “Конфігурація ПЛК” можна задати власні імена входів та виходів, а також сконфігурувати їх: вибрати тип вхідного сигналу, діапазон значень, період опитування, “безпечне” значення (в яке встановлюється вихід, поки йому ще не присвоїли значення). Змінні, доступні для обміну в мережі, також відносяться до адресного простору вводу-виводу.

11.3. Програмне та апаратне забезпечення роботи

1. Персональний комп'ютер
2. Середовище програмування Codesys 2.3, встановлене на ПК;
3. Програмований логічний контролер ОВЕН ПЛК-154-У-М;
4. Target-файли для ПЛК ОВЕН.
5. Перетворювач інтерфейсів RS 485/USB.

11.4. Програма роботи

1. Ознайомитися з технічними характеристиками та функціональними можливостями ПЛК.
2. Ознайомитися з основними принципами роботи в середовищі Codesys 2.3.
3. Розробити допоміжну документацію для реалізації програмного управління типовим періодичним технологічним процесом.
4. Розробити програмне забезпечення для ПЛК.
5. Перевірити правильність роботи розробленого програмного забезпечення шляхом програмної емуляції та із завантаженням програми в ПЛК.

11.5. Порядок виконання роботи

1. Ознайомитися із зовнішнім виглядом, будовою та функціональними можливостями використовуваного обладнання.
2. Перед початком роботи із середовищем Codesys 2.3 переконатись у наявності встановлених необхідних Target-файлів. Для цього потрібно запустити відповідне програмне забезпечення (Старт-меню Windows => Програми => 3S Software => CoDeSys

V2.3 => InstallTarget) та знайти модель ПЛК у списку встановлених (рис.11.5). Якщо відповідний Target-файл відсутній у списку, його потрібно встановити. Target-файли поставляються виробником обладнання та можуть бути отримані з супутніх електронних носіїв або з центрів завантажень на електронних ресурсах виробника. При цьому вони можуть поставлятися у вигляді XML-документів, тоді їх потрібно встановлювати вручну, з використанням вищезгаданого програмного забезпечення, або у вигляді інсталяційних пакетів, після запуску котрих target-файли автоматично заносяться до бази.

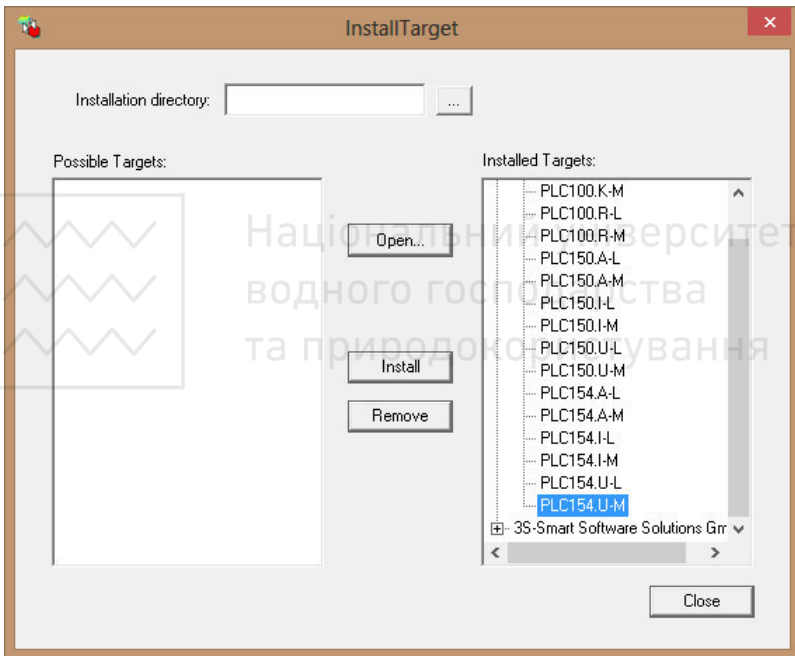


Рис.11.5. Інтерфейс програмного забезпечення для роботи з Target-файлами

3. Запустити середовище програмування (Програми => 3S Software => CoDeSys V2.3 => CoDeSys V2.3) та створити новий проект. У діалоговому вікні (рис.11. 6) обрати цільову платформу (у нашому випадку ПЛК-154-У-М), решту налаштувань можемо залишити за замовчуванням.

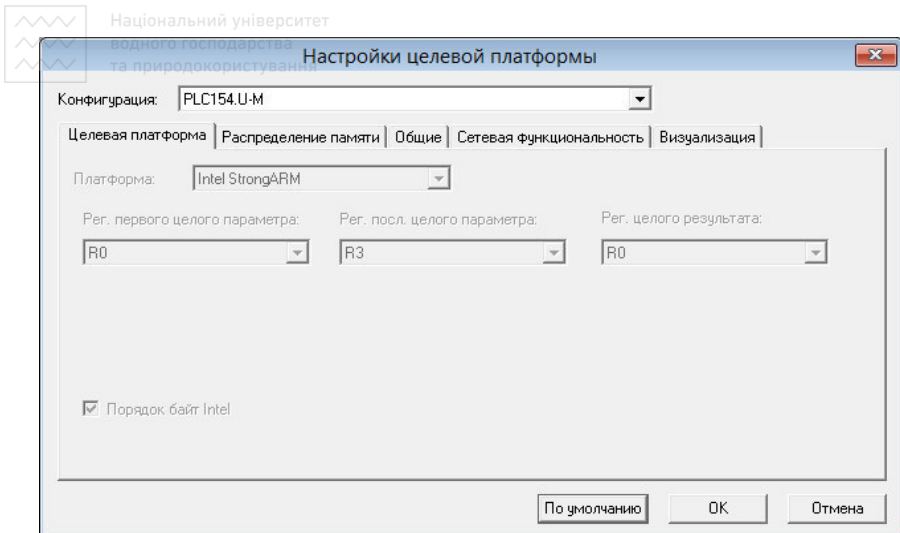


Рис.11. 6. Вікно вибору та налаштувань цільової платформи

Далі потрібно створити новий програмний компонент, при цьому задати назву, тип та мову його реалізації. (рис.11.7).

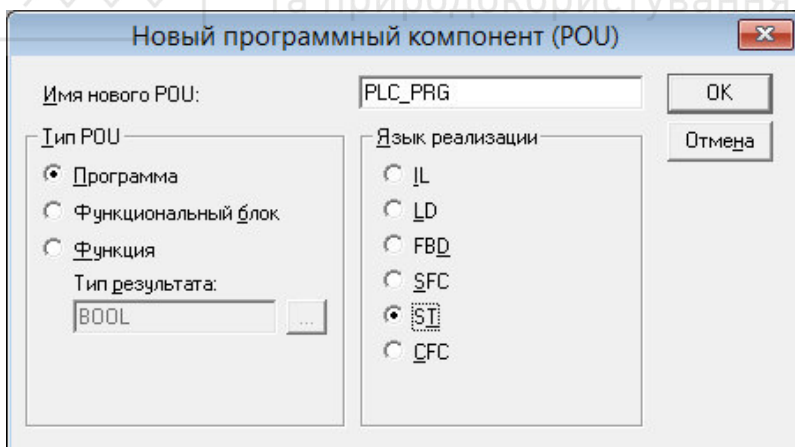


Рис.11.7 Діалогове вікно створення програмного компоненту

Для реалізації простої програми із використанням лише дискретних входів та виходів створимо головну програму (при

цьому вона повинна називатися саме PLC_PRG). Зазвичай, особливо при реалізації великих проектів, програмний алгоритм бажано розбивати на модулі, а головну програму створювати максимально лаконічно. Такий підхід в подальшому дозволить уникнути плутанини із великим кодом та значно пришвидшити налагодження отриманої програми. Крім того, створені раніше POU можна буде повторно використати у інших проектах для вирішення типових задач. Для головних програм зазвичай використовують мови SCF, CFC, ST, однак, чітких правил щодо цього немає. В нашому випадку використаємо мову ST. Результатом буде перехід до вікна написання програмного коду (рис.11.8).

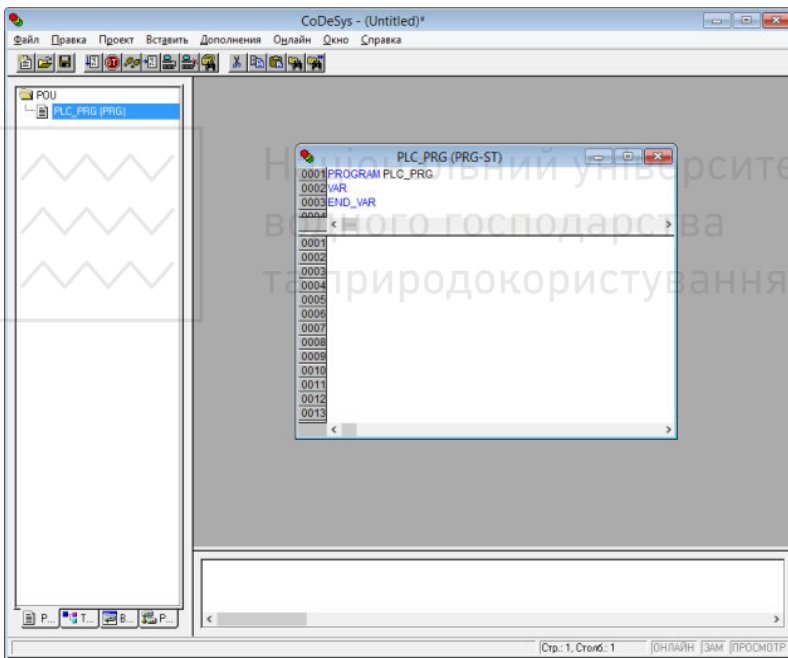


Рис.11.8. Загальний вигляд вікна для вводу програмного коду

4. Розглянемо способи вирішення декількох нескладних, типових для періодичних процесів задач.

Задача №1. Об'єктом регулювання є резервуар, у котрому потрібно підтримувати постійний рівень рідини (рис. 11.9). Рідина

надходить у резервуар 1 через трубопровід 2, на котрому встановлено електромагнітний клапан 3, що відкривається при подачі живлення. Для контролю рівня рідини встановлено поплавковий рівнемір 4, котрий у верхньому положенні замикає контакт.

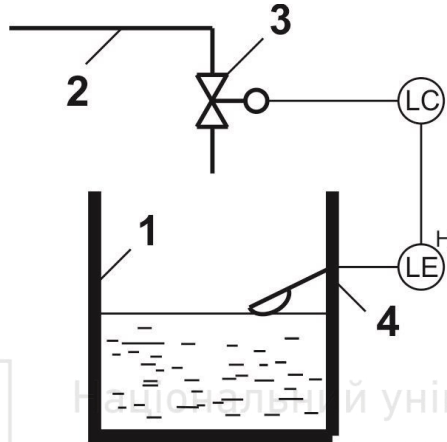


Рис.11.9. САР рівня води у резервуарі

Опис роботи системи: При заданому рівні рідини у резервуарі поплавок знаходиться у верхньому положенні, при цьому він замикає контакт SA1, напруга на електромагнітному клапані відсутня, що відповідає його закритому стану. При зменшенні рівня рідини в резервуарі поплавок опускається, контакт 1 розмикається, подається живлення на електромагнітний клапан, який відкривається та відновлюється подача рідини.

Для реалізації програмного забезпечення ПЛК створимо таблицю змінних проекту (табл.11.2)

Таблиця 11.2.

Таблиця змінних

№	Параметр	Ім'я	Тип	Клас	Адреса
1	Верхній рівень	LN	Дискретна	глобальна	%IX0.0
2	Клапан	VL	Дискретна	глобальна	%QX1.0

Тепер повернемося до Codesys та створимо підпрограму



керування рівнем в резервуарі. Для цього на вкладці ROU правий клік => Додати об'єкт... (рис.11.9).

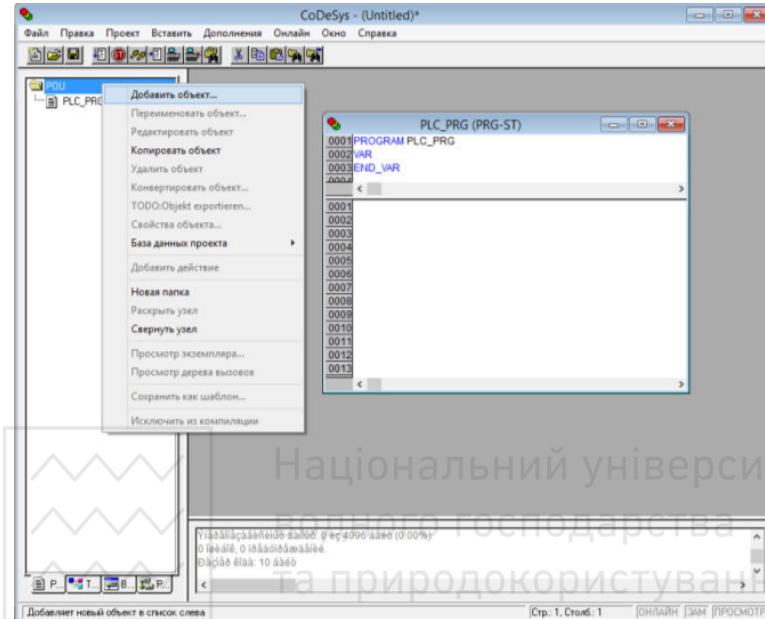


Рис.11.9. Додавання нового ROU

У діалоговому вікні виберемо тип ROU - програма, мова - LD та назвемо її, наприклад «Lev_one». Хоча тип програмного компоненту потрібно обирати якомога простіший, однак, для першого прикладу створимо одразу програму.

Прив'яжемо змінні згідно таблиці до входів та виходів ПЛК, при цьому не обов'язково оголошувати їх вручну або за допомогою діалогового вікна, можна перейти на вкладку «Ресурси» та у конфігураторі ПЛК задати назви відповідних входів та виходів (рис.11.10).

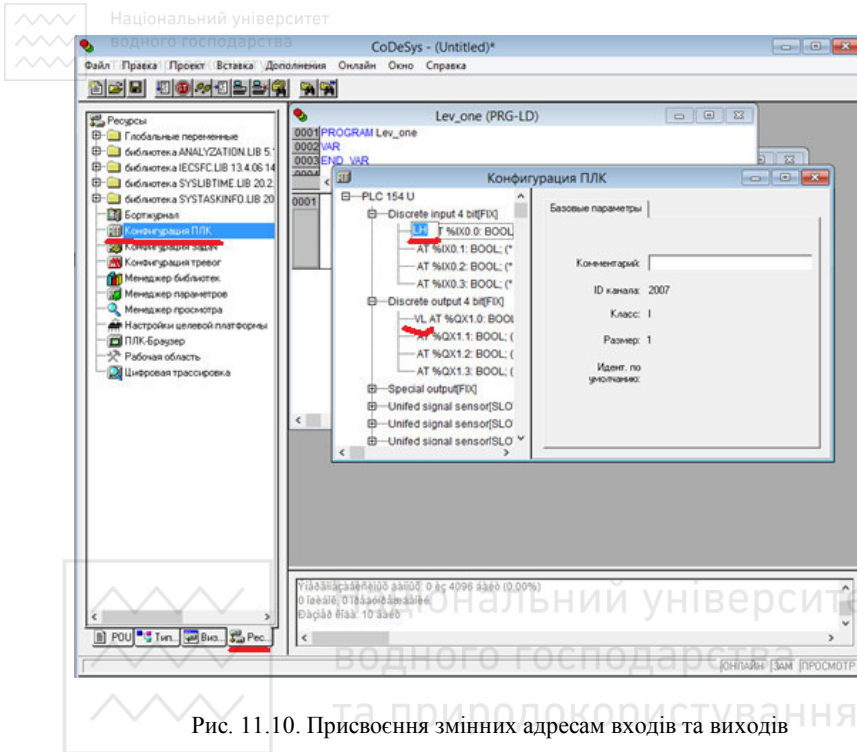


Рис. 11.10. Присвоєння змінних адресам входів та виходів

Повернемося до підпрограми Lev_one. Програми мовою драбінкової логіки виконуються лініями зверху вниз, зліва на право. Вхідні змінні, або умови логіки щабля позначаються контактами (нормально замкнутими або розімкнутими), вихідні – котушками (обмотками) реле. Таким чином, змінній LH (типу BOOL), що відповідає стану дискретного входу додамо відповідний контакт із панелі інструментів ($\overline{I1} \overline{I2} \overline{I3} \overline{I4}$), а вихідній змінній – котушку ($\rightarrow Q1$). При цьому для написання умови активації котушки використаємо нормально замкнений контакт (при розімкненому контакті SA1, тобто поплавков знаходиться нижче заданого рівня, при цьому змінній LH відповідає значення FALSE, в цей час повинен бути активним дискретний вихід для активації електромагнітного клапана, а змінна VL повинна мати значення TRUE). Готову програму зображено на рис.11.11.

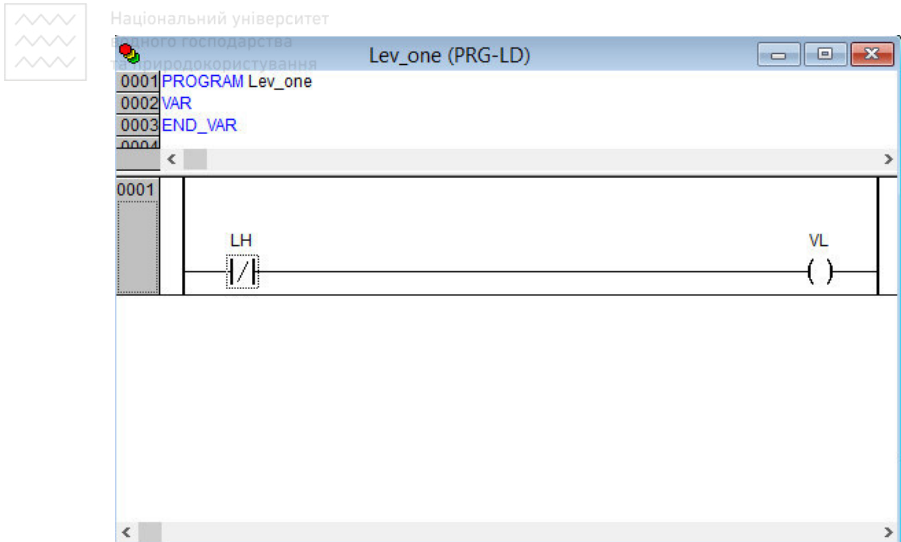


Рис. 11.11. Підпрограма управління рівнем у резервуарі

Для використання створеної підпрограми Lev_one потрібно викликати її з головної програми PLC_PRG. Для цього потрібно додати рядок «Lev_one;» у вікні редагування коду головної програми.

Тепер у запусимо програму у режимі емуляції, для чого у режимі емуляції (Онлайн-> Режим емуляции) під'єднаємося до емульованого ПЛК та запусимо програму за допомогою вкладки Онлайн головного меню (або ALT+F8) та Старт (або F5).

Якщо наш проект не містить помилок, розпочнеться виконання програми в режимі емуляції, при цьому сигнальна лінія синього кольору у LD-програмі свідчитиме про проходження сигналу.

Перейдемо до конфігуратора ПЛК та змінимо стан дискретного виходу натисканням на піктограму його стану та прослідкуємо правильність виконання програми (рис.11.12).

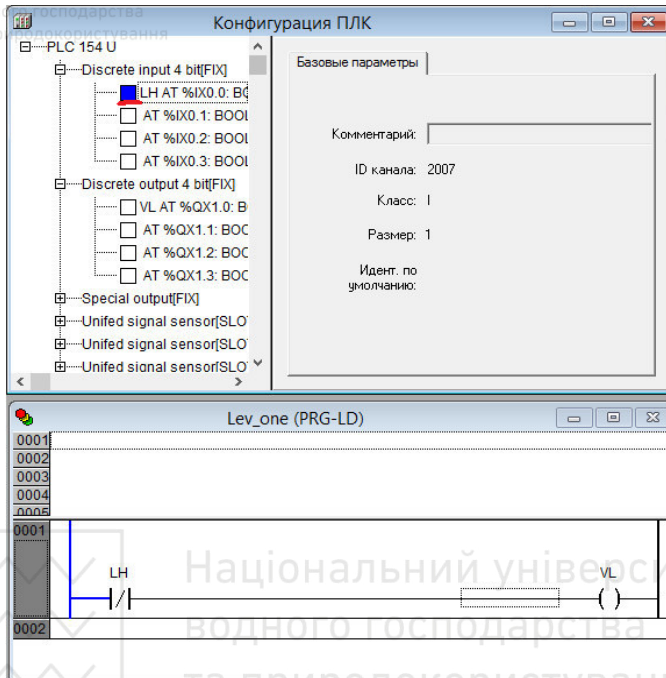


Рис.11.12. Режим емуляції роботи ПЛК

Тепер, коли проект готовий, перевіримо його роботу в ПЛК, для цього потрібно:

- Виконати відключення ПЛК та вийти із режиму емуляції;
- Використовуючи вкладку Онлайн головного меню, виконати налаштування підключення до ПЛК;
- З'єднати контролер із ПК та виконати підключення;
- Завантажити програму до оперативної пам'яті ПЛК;
- Натиснути кнопку «Старт» на передній панелі контролера;
- Перевірити роботу системи.

Задача №2. Об'єктом регулювання є резервуар для об'ємного дозування рідини (рис.11.13). Рідина надходить у резервуар 1 через трубопровід 2, на котрому встановлено електромагнітний клапан 3, що відкривається при подачі живлення. Для контролю рівня рідини встановлено поплавковий рівнемір 4, котрий у верхньому положенні



водного господарства та природокористування
замикає контакт SA1, а у нижньому – SA2. Скидання продукту відбувається клапаном 5.

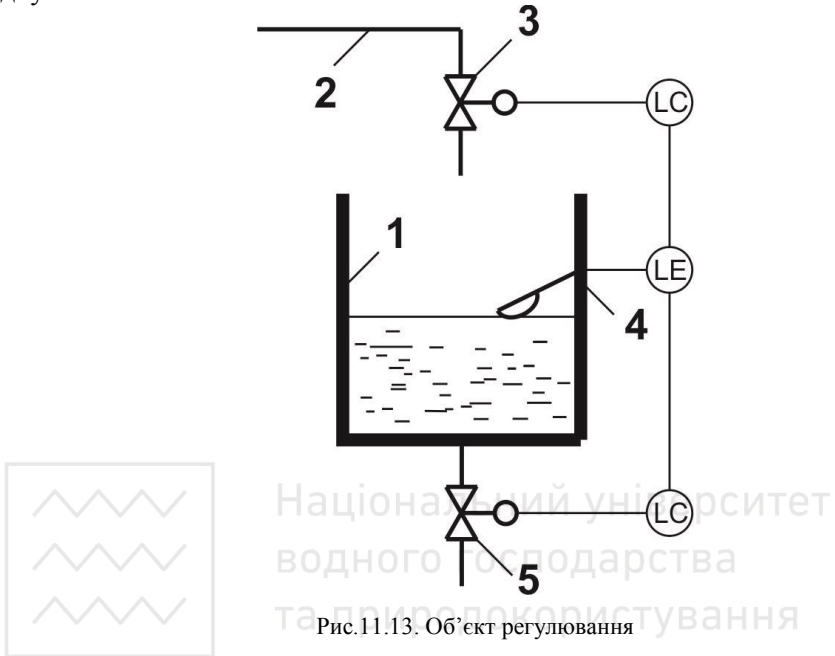


Рис.11.13. Об'єкт регулювання

Опис процесу: При досягненні верхнього рівня відбувається замикавання контакту SA1, при цьому потрібно припинити подачу продукту до ємності клапаном 3 та відкрити скидний клапан 5. При досягненні нижнього рівня потрібно закрити скидний клапан 5 та відкрити клапан подачі продукту 3.

Складемо таблицю змінних процесу (табл. 11. 3):

Таблиця 11.3

Таблиця змінних

№	Параметр	Ім'я	Тип	Клас	Адреса
1	Верхній рівень	LH	Дискретна	глобальна	%IX0.0
2	Нижній рівень	LL	Дискретна	глобальна	%IX0.1
3	Клапан 3	VL1	Дискретна	глобальна	%QX1.0
4	Клапан 5	VL2	Дискретна	глобальна	%QX1.1
5	Початкова умова	ON	Дискретна	локальна	-



Після прив'язки їх до входів та виходів ПЛК перейдемо до створення програмного забезпечення. Загалом існує два підходи до вирішення подібної задачі. Перший базується на реалізації умов виконання певної дії. В нашому випадку – це умови відкриття або закриття клапанів та реалізуються перевіркою станів елементів системи. Так, клапан 3 може бути відкритим лише за умови, коли нижній клапан 5 закритий, а рідина не досягла верхнього рівня. І навпаки, нижній клапан 5 може бути відкритим за умови, що верхній клапан 3 закритий та не замкнутий контакт нижнього рівня. Програма, що реалізує даний алгоритм, зображена на рис.11.14.

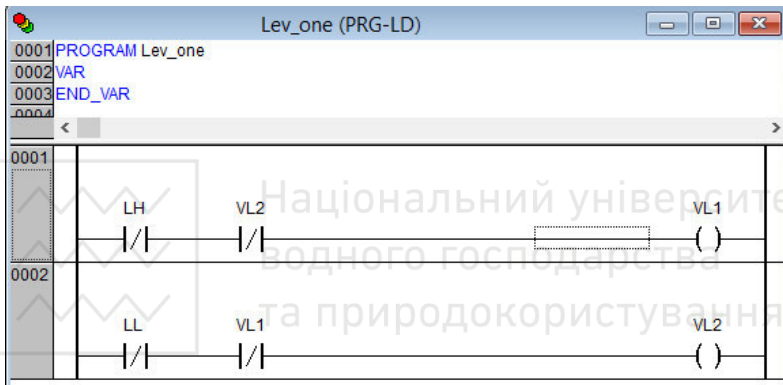


Рис.11.14. Програма роботи ПЛК для об'ємного дозування рідини

Інший підхід передбачає використання певної події у якості умови. У нашому випадку це може бути спрацювання контактів верхнього або нижнього рівня, при цьому потрібно необхідно закрити або відкрити відповідні клапани. Такий алгоритм доцільно реалізувати використанням Set і Reset котушок, стан яких зберігається при зникненні сигналу до моменту, коли надійде протилежна команда, однак, в цьому випадку потрібно передбачити додаткову змінну (ON) для відкриття хоча б одного клапана та ініціалізації алгоритму, якщо у початковий момент часу жоден з контактів не був замкнений, або при оголошенні змінної одного із дискретних виходів задати його стан, як активний (рис.11.15).

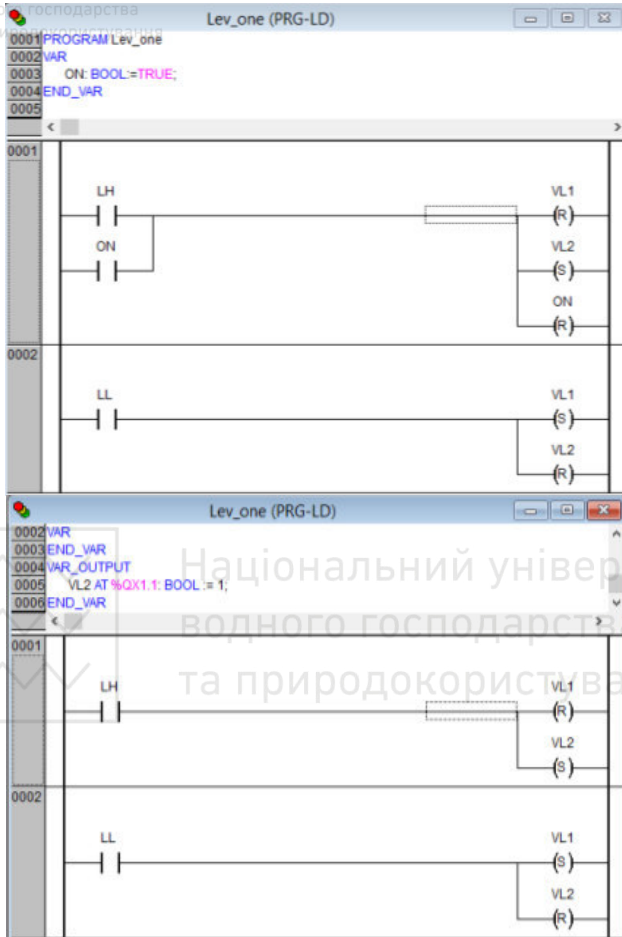


Рис. 11.15. Програма роботи ПЛК для об'ємного дозування рідини

Задача №3 Умова збігається з умовою Задачі №2. Потрібно додатково передбачити затримку 3 секунди після досягнення нижнього рівня.

Для реалізації такого програмного забезпечення контролера потрібно додати функціональний блок таймера (правий клік на шаблі програми, «Функціональний блок...», або Ctrl+B), оголосити екземпляр функціонального блоку та задати час на вході блоку PT у

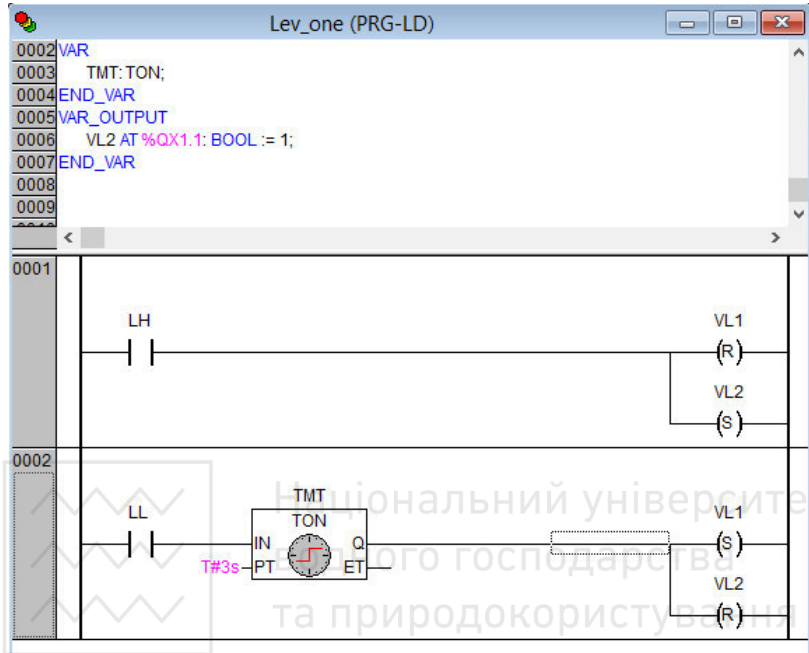


Рис. 11.16. Програма роботи ПЛК для об'ємного дозування рідини з таймером

Задача №4. Розробити програмне забезпечення процесу об'ємного дозування двох рідких компонентів (рис.11.17)

Звіт про виконання роботи повинен містити:

1. Функціональну схему і технологічні вимоги до системи управління.
2. Таблицю змінних.
3. Знімки екранів розробленого ПЗ.
4. Пояснення роботи розробленого програмного забезпечення.



Національний університет
водного господарства
та природокористування

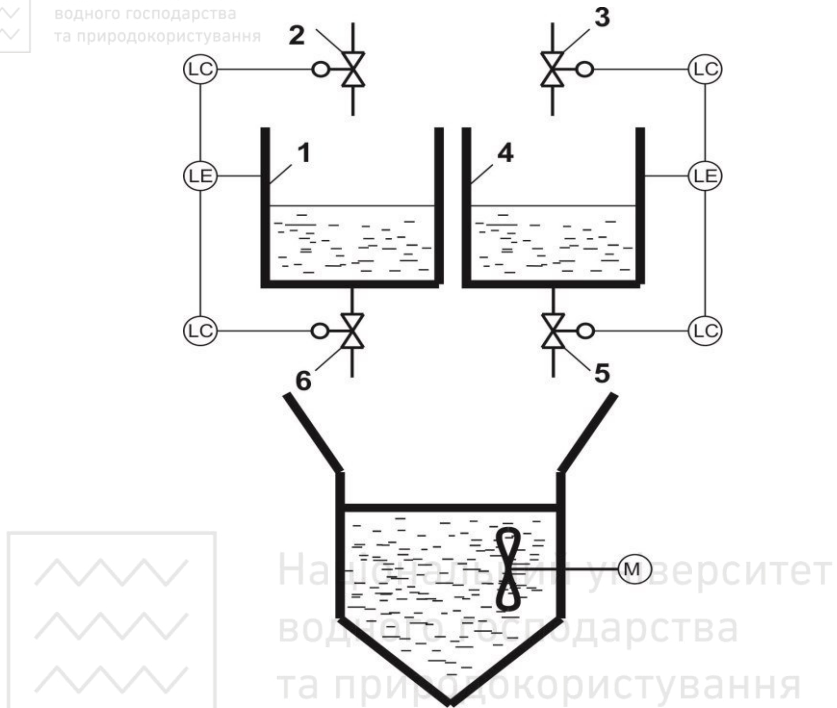


Рис.11.17 Схема установки для утворення двокомпонентної суміші

11.6. Контрольні запитання

1. Яке призначення та склад програмного комплексу Codesys?
2. Які мови програмування підтримуються середовищем Codesys 2.3?
3. Який порядок розроблення програмної реалізації управляючих функцій ПЛК в середовищі Codesys 2.3?
4. Які типи змінних використовуються середовищем Codesys 2.3?
5. Що таке Target-файли?
6. Для чого використовуються Target-файли?
7. Які організації розробляють Target-файли?
8. Які типи змінних використовуються в ПЛК?
9. Яка технологія флеш-пам'яті використовуються в ПЛК?
10. Які інформаційні ресурси ПЛК Овен?

WHILE	Команда циклу	WHILE i > 0 DO n := n * 2; END WHILE;
REPEAT	Команда циклу	REPEAT k := k + i; UNTIL i < 20; END REPEAT;
EXIT	Вихід із циклу	EXIT;
RETURN	Вихід із підпрограми	RETURN;
;	Пуста команда	::

12.3. Програмне та апаратне забезпечення роботи

1. Персональний комп'ютер.
2. Середовище програмування Codesys 2.3.
3. OPC-сервер Lectus.
4. Програмований логічний контролер ОВЕН ПЛК-154-У-М;
5. Віддалені модулі розширення вводу/виводу ОВЕН;
6. Перетворювач інтерфейсів ОВЕН ЕКОН-134.

12.4. Опис технологічного процесу утворення розчину живильних солей у виробництві кормових дріжджів

Об'єктом автоматизації є однофункціональна тристадійна мішалка періодично дії, що використовується для приготування як освітленого, так і неосвітленого розчину живильних солей у виробництві кормових дріжджів (рис.12.1).

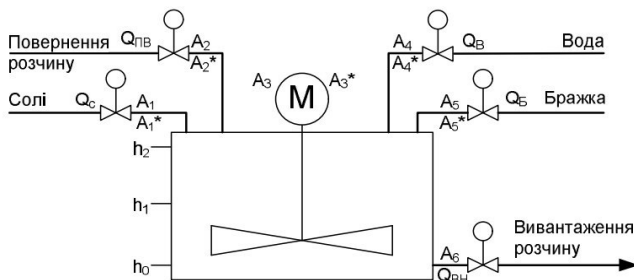


Рис. 12.1. Схема процесу змішування періодичної дії



Цикл роботи мішалки включає наступні стадії:

1. Завантаження розчинника;
2. Завантаження солей та початок перемішування;
3. Закінчення завантаження та розчинення;
4. Закінчення перемішування та початок освітлення;
5. Закінчення освітлення та початок вивантаження освітленого розчину;
6. Закінчення вивантаження освітленого розчину та дозавантаження розчинника;
7. Дозавантаження солей та початок перемішування;
8. Закінчення дозавантаження та розчинення;
9. Закінчення розчинення та початок вивантаження розчину з частковим його поверненням для кращого перемішування;
10. Закінчення перемішування та закінчення повернення розчину;
11. Закінчення вивантаження розчину.

12.5. Порядок виконання роботи

1. Побудувати логіко-динамічну схему алгоритму функціонування мішалки.
2. Побудувати граф-схему алгоритму функціонування мішалки.
3. Засобами середовища CodeSys 2.3 створити програмне забезпечення ПЛК для управління технологічним процесом.

Звіт про виконання роботи повинен містити:

1. Логіко-динамічну схему алгоритму функціонування мішалки.
2. Граф-схему алгоритму функціонування мішалки.
3. Знімки екранів проведених налаштувань у ході роботи та знімок або текст програмного коду ПЛК з поясненнями.
4. Знімок екрану роботи OPC-сервера та лог передачі даних між ПЛК та ПК.

12.6. Контрольні запитання

1. Які існують елементи мови структурованого тексту?
2. Які переваги і недоліки мови структурованого тексту?



3. Які способи представлення алгоритму функціонування програмної системи?
4. Яке призначення OPC-сервера?
5. Як налаштовують OPC-сервер?
6. В чому суть емуляції ПЗ?
7. Які функції емулятора CodeSys 2.3?
8. Які функції перетворювача інтерфейсів?
9. Які функції автоматизації реалізовано програмно?
10. Як реалізовано блокування?

Робота 13. Дослідження розподіленої системи управління

13.1. Мета роботи

Ознайомитися з принципами побудови промислових мереж, промисловими протоколами передачі даних між пристроями автоматизації. Навчитись зв'язувати ПЛК з верхнім рівнем АСУ та віддаленими модулями вводу/виводу за протоколом Modbus RTU.

13.2. Теоретичні відомості

Можливості вводу-виводу конкретного ПЛК описуються в target-файлі. За необхідності можна вибрати інший ПЛК зміною налаштувань цільової платформи на вкладці “Ресурси”. За замовчуванням входи та виходи ПЛК адресуються виразами %AV.N, де А визначає напрям: вхід (I) або вихід (Q), В – тип змінної (R – дійсний, W – двійкове слово, X – байт), N – номер біта. Але зазвичай програмісту зручніше працювати з осмисленими назвами змінних, тому в вікні “Конфігурація ПЛК” можна задати власні імена входів та виходів, а також зконфігурувати їх: вибрати тип вхідного сигналу, діапазон значень, період опитування, “безпечне” значення (в яке встановлюється вихід, поки йому ще не присвоїли значення).

Змінні, доступні для обміну в мережі, також відносяться до адресного простору вводу-виводу, тому для створення відповідної змінної потрібно в конфігурації ПЛК викласти контекстне меню

кореневого елемента (як правило, його назва збігається з назвою ПЛК) та додати піделемент з назвою мережевого протоколу (рис. 13.1). У випадку вибору Modbus в параметрах модуля новоствореного елемента необхідно задати мережеву адресу ПЛК (рис. 13.2) та додати піделемент, що відповідатиме апаратному інтерфейсу, через який ПЛК підключений до мережі (рис.13.3). На вкладці “Параметри модуля” вибраного інтерфейсу задається тип фрейму (ASCII або RTU) та налаштування фізичного рівня (рис.13.4). На цьому конфігурація власне протоколу завершується та можна додавати змінні, доступні через цей інтерфейс (рис. 13.5).

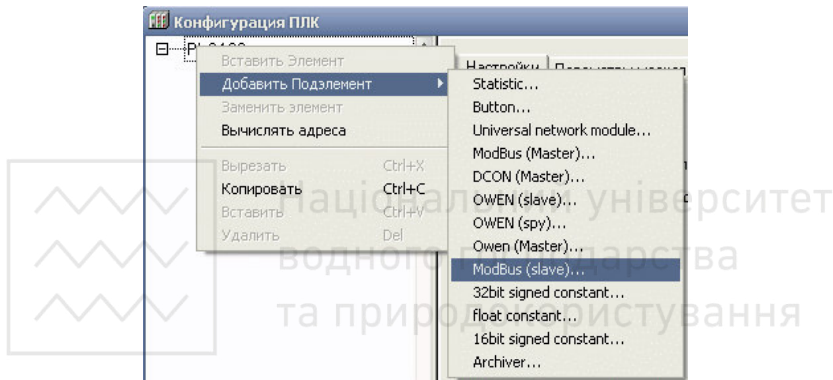


Рис.13.1. Вибір мережевого протоколу

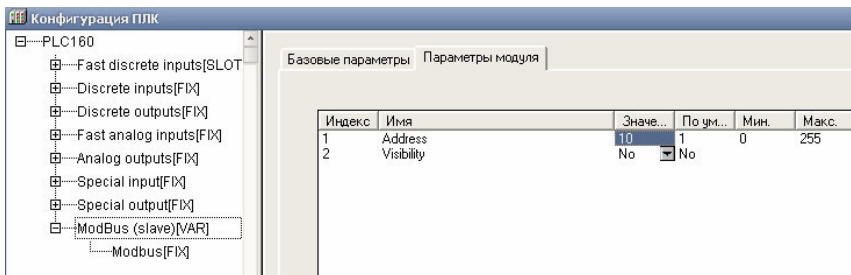


Рис. 13.2. Задання мережевої адреси ПЛК



Національний університет
водного господарства
та природокористування

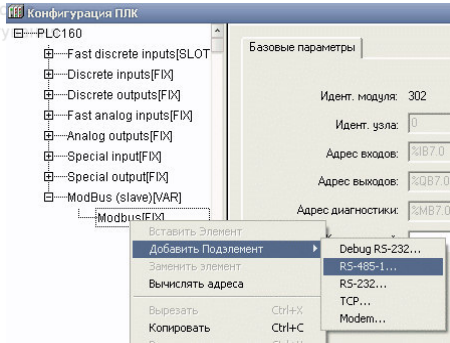


Рис. 13.3. Вибір апаратного інтерфейсу

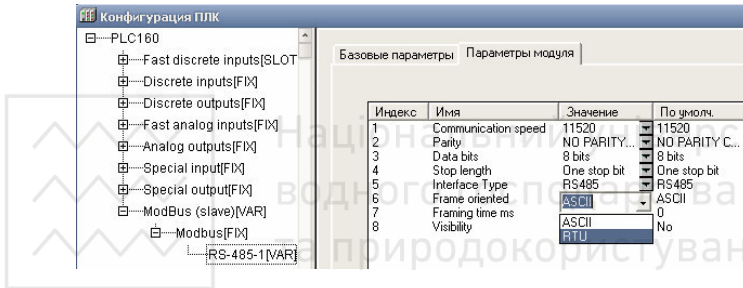


Рис. 13.4. Вибір типу фрейму та налаштувань фізичного рівня

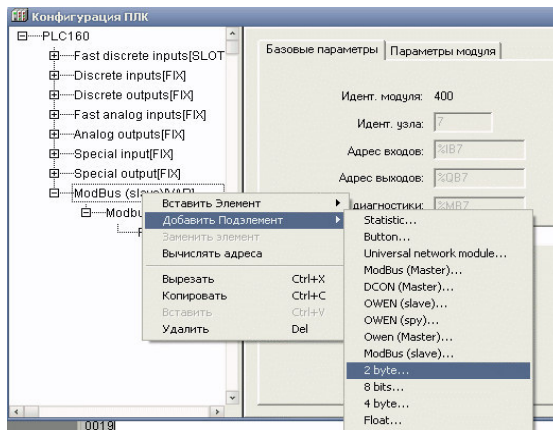


Рис. 13.5. Додавання нової змінної розміром 2 байти



Відповідно до специфікації протоколу Modbus на прикладному рівні дані представлені в одиничних котушках (містять один біт інформації) та в 16-бітних регістрах. З точки зору Master-пристрою Slave-пристрій являє собою неперервну область пам'яті (рис. 13.6), що адресується по регістрах з адресою від 0.

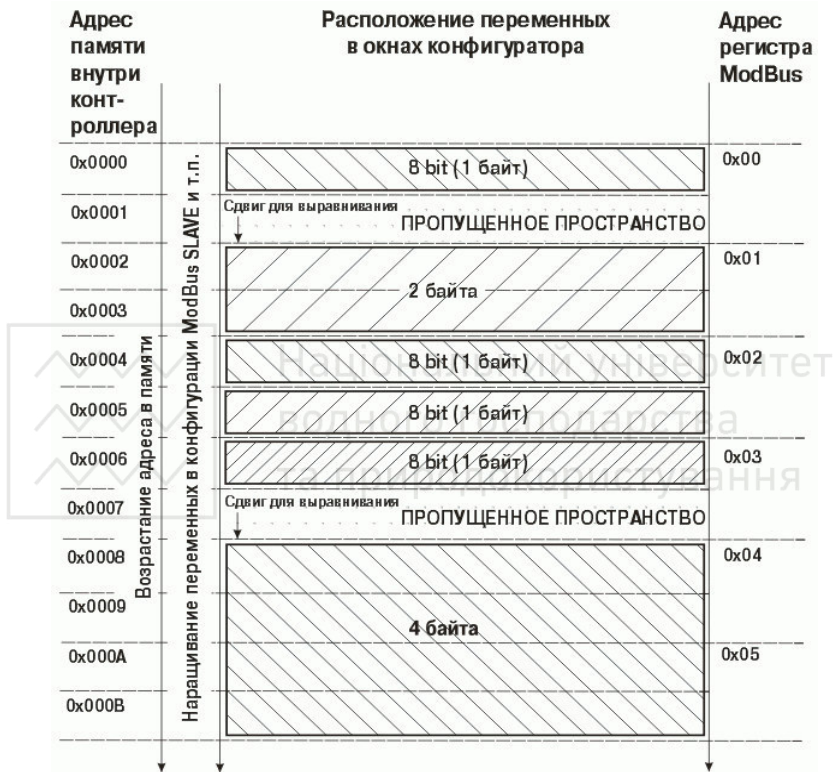


Рис. 13.6. Вирівнювання змінних, доступних через Modbus

Як правило, виникає необхідність передавати змінні різного розміру, тому для структурування пам'яті використовується вирівнювання змінних. Його суть полягає в тому, що адреса змінної повинна бути кратна її розміру в байтах, тобто однобайтові змінні можуть розміщатись з комірок пам'яті за адресами 0x0000, 0x0001, 0x0002, 0x0003, двохбайтові — 0x0000, 0x0002, 0x0004, 0x0006, чотирьохбайтові — 0x0000, 0x0004, 0x0008, 0x0012. Відповідно,



якщо за адресою 0x0006 знаходилась одnobайтова змінна, то наступна чотирьохбайтова змінна розміститься не в наступній вільній комірці 0x0007, а за адресою 0x0008.

13.3. Програмне та апаратне забезпечення роботи

1. Персональний комп'ютер
2. Середовище програмування Codesys 2.3, встановлене на ПК.
3. OPC-сервер Lectus.
4. Програмований логічний контролер ОВЕН ПЛК-154-У-М.
5. Віддалені модулі розширення вводу/виводу ОВЕН.
6. Перетворювач інтерфейсів ОВЕН ЕКОН-134.

13.4. Програма роботи

1. Конфігурування перетворювача інтерфейсів ЕКОН-134;
2. Конфігурування мережевих параметрів віддалених модулів розширення вводу/виводу;
3. Підключення модулів до ПЛК;
4. Конфігурування ПЛК для роботи з модулями вводу/виводу та OPC-сервером;
5. Написання простого програмного забезпечення для взаємодії ПЛК з модулями та OPC-сервером;
6. Перевірка роботи розробленої системи.

13.5. Порядок виконання роботи

1. Під'єднати перетворювач інтерфейсів ЕКОН-134 до ПК та провести налаштування мережевої карти.
2. За допомогою «конфігуратора віртуальних портів» провести налаштування портів перетворювача інтерфейсів;
3. По черзі під'єднати модулі та провести їх конфігурування за допомогою ПЗ «Конфігуратор М110»;
4. З'єднати модулі з ПЛК;
5. Провести конфігурування ПЛК для роботи в мережі. Створити конфігурацію ролі Master для роботи з модулями вводу/виводу та конфігурацію Slave – для опитування ПЛК OPC-сервером.



6. Створити програмне забезпечення ПЛК для перевірки правильності його роботи у складі системи.
7. Сконфігурувати OPC-сервер для обміну даними з ПЛК.

Звіт про виконання роботи повинен включати:

1. Знімки екранів проведених налаштувань у ході роботи.
2. Знімок або текст програмного коду ПЛК з поясненнями.
3. Знімок екрану роботи OPC-сервера та лог передачі даних між ПЛК та ПК.

13.6. Контрольні запитання

1. Які характеристики промислової мережі RS485?
2. Яка максимальна довжина шини RS485?
3. Для чого кінець шини закорочують резистором?
4. Яка процедура вирішення колізій на шині?
5. Які функції пристрою Master?
6. Які функції пристрою Slave?
7. Які налаштування шини RS485?
8. Як сконфігурувати пристрої для роботи в мережі?
9. Які функції програмного модуля ПЛК для роботи в мережі?
10. Яке призначення віртуальних портів?

Робота 14. Дослідження системи автоматизації процесу дозування рідин

14.1. Мета роботи

Вивчити технологію процесу дозування. Вивчити будову і принцип дії комп'ютерно-інтегрованої системи автоматизації процесу дозування рідин

14.2. Теоретичні відомості

Процес дозування широко поширений в галузях хімічної, харчової, легкої промисловості.



При автоматизації процесів дозування основна увага приділяється регулюванню співвідношення витрат заданих рецептурою компонентів, при цьому пред'являються підвищені вимоги до точності дозування.

Дозування може здійснюватися за рівнем, за об'ємом (наприклад за допомогою багатокомпонентного насоса-дозатора), або за масою (за допомогою вагових пристроїв).

Дозатори рідин – пристрої для автоматичного відмірювання (дозування) заданої маси або об'єму рідини. Вони забезпечують видачу дози одного або декількох продуктів (відповідно одно- і багатокомпонентні дозатори) одному або різним споживачам (одно- і багатоканальні дозатори); дозують речовини в заданій часовій або логічній послідовності (програмні дозатори). Найбільша ефективність використання дозаторів досягається, якщо регулятором або його основою служать мікропроцесорні пристрої (ПЛК, мікроконтролери), що дозволяють компенсувати вплив зовнішніх збурюючих дій (напр., параметрів технологічного режиму процесу), вести дозування за заданою програмою, зручно представляти інформацію операторові та передавати результати дозування (наприклад, загальний об'єм використаного продукту) на наступний рівень управління.

Дозування за масою. Устаткування, що здійснює дозування за масою (вагове дозування) є достатньо повільним, в порівнянні з описаним вище устаткуванням, проте воно виявляється незамінним у випадках, коли величина дози повинна складати десятки, або навіть сотні кілограмів. Дозування в таку велику тару за рівнем або за об'ємом є недоцільним.

Вагові дозатори застосовують для дозування твердих сипучих матеріалів, рідше – рідин. Дози від декількох *грамів* до сотень *кілограмів*, продуктивність – від сотень до десятків *т/год.*, похибка дозування – від 0,1 до 0,5%. З дозаторів дискретної дії найбільше поширені в хімічній промисловості такі, в яких завантажувана ємність встановлена на силовимірювальних перетворювачах – тензометричних або платформних вагах.

14.3. Програма роботи

1. Ознайомитися з сучасним методами дозування рідин.



2. Вивчити будову і принцип дії лабораторної установки дозування рідин. Детально ознайомитися з призначенням всіх елементів системи.

3. Побудувати структурну і функціональну схеми досліджуваної системи автоматизації процесу дозування рідин.

4. Ознайомитися з програмним забезпеченням ПЛК та операторської станції.

14.4. Опис лабораторної установки

Схема лабораторної установки зображена на рис. 14.1.

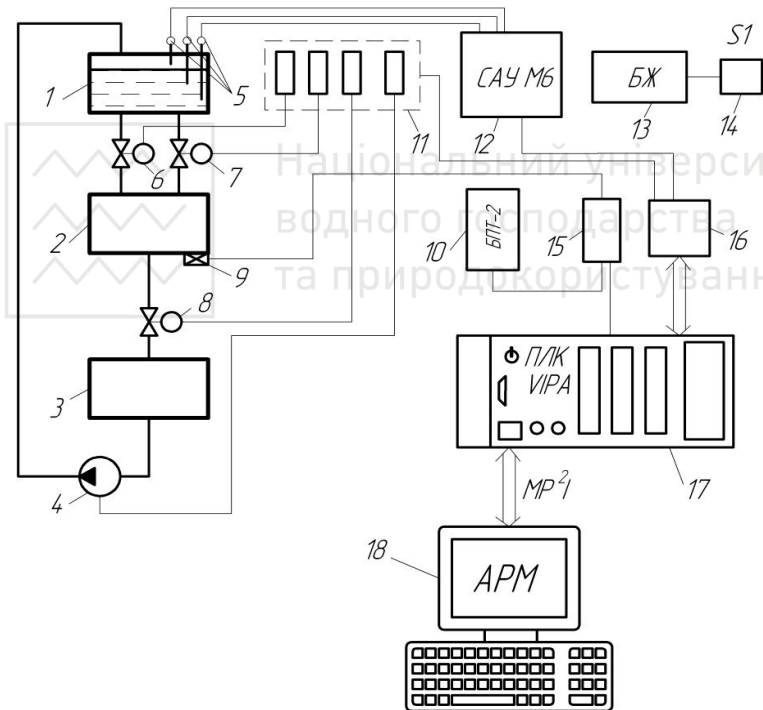


Рис. 14.1. Структурна схема лабораторної установки


Лабораторна установка складається із резервуара дозованої рідини 1, резервуара для зважування заданої дози рідини 2, резервуара, в який проводиться дозування 3, насоса 4, електродних

давачів рівня 5, електромагнітних клапанів 6 – 8, тензодавача 9, блока перетворення сигналів з тензодавачів 10 (БПТ-2), релейних пристроїв 11, сигналізатора рівня рідини 12 (ОВЕН САУ-М6), блока живлення 13, вимикача 14, клемних колодок 15 – 16, програмованого логічного контролера 17 (VIPA 114) та операторської станції 18.

Дана автоматизована система управління процесом дозування рідини реалізована на базі програмованого логічного контролера німецького виробництва VIPA 114. Серія контролерів VIPA 100V – це ПЛК малого класу з можливістю розширення, призначені для використання в якості віддалених інтелектуальних вузлів у розподілених мережах технологічного рівня, а також для автономної роботи в якості пристроїв управління невеликими установками.

Для візуалізації і дистанційного управління процесом даний ПЛК підключений до операторської станції (АРМ оператора) за допомогою інтерфейсу MP²I. АРМ оператора реалізоване на базі SCADA-системи TRACE MODE 6. Для зв'язку ПЛК зі SCADA-системою використовується OPC-сервер фірми VIPA.

14.5. Порядок виконання роботи

1. Ознайомитися з будовою, принципом дії лабораторної установки та призначенням всіх її елементів.
2. Увімкнути комп'ютер, завантажити інтегроване середовище SCADA-системи TRACE MODE 6 та відкрити збережений проект.
3. Перевірити налаштування OPC-сервера. Скопіювати в звіт налаштування змінних OPC-сервера.
4. Запустити відлагодження проекту в TRACE MODE, натиснувши кнопку .
5. З дозволу викладача увімкнути установку.
6. Натиснути кнопку "Змінити завдання" на мнемосхемі АРМ оператора та задати значення дози рідини (100 г).
7. Натиснути кнопку "Розпочати дозування".
8. Спостерігати за процесом дозування до повторення циклу.
9. Скопіювати в звіт мнемосхему і графік процесу дозування.
10. Змінюючи значення завдання, провести ще 3 дозування та заповнити таблицю 14.1.



Дослідження точності дозування рідини

№ п/п	Завдання, г	Маса отриманої дози, г	Абс. похибка, г	Відн. похибка, %
1	100			
2	150			
3	200			
4	250			

12. Завантажити середовище програмування WinPLC7.

13. Відкрити програму для даної лабораторної установки, написану мовою драбинкових діаграм (LAD), ознайомитися з призначенням блоків у програмі та їх функціями.

14. Скласти блок-схему програми дозування.

15. Побудувати структурну і функціональну схеми досліджуваної системи автоматизації процесу дозування рідини. Описати принципи дії лабораторної установки.

16. Оформити звіт про виконання роботи, вказавши фактори, що впливають на точність дозування у даній роботі. Зробити висновки.

14.6. Контрольні запитання

1. З чого складаються автоматичні вагові дозатори рідин?
2. Які існують методи дозування?
3. В чому полягає дозування за рівнем?
4. Який принцип роботи об'ємного поршневого дозатора дискретної дії.
5. Який принцип роботи вагового дозатора рідин дискретної дії?
6. Який принцип роботи тензодавача?
7. Які функції автоматизованої системи вагового дозування?
8. Які функції пристрою БПТ-2 експериментальної установки?
9. Якою мовою реалізовано програмне забезпечення управління дозуванням?
10. Як встановити залежність машинної змінної від маси рідини?



Робота 15. Дослідження автоматизованої системи дозування за рецептом

15.1. Мета роботи

Вивчити технологію процесу дозування двокомпонентної суміші. Вивчити будову і роботу комп'ютерно-інтегрованої системи автоматизації процесу дозування двох компонентів. Ознайомитись з програмним забезпечення системи дозування за рецептом для приготування заданого складусуміші.

15.2. Теоретичні відомості

Дана автоматизована система управління процесом дозування рідини реалізована на базі програмованого логічного контролера німецького виробництва VIPA 114. Серія контролерів VIPA 100V – це ПЛК малого класу з можливістю розширення, призначені для використання в якості віддалених інтелектуальних вузлів у розподілених мережах технологічного рівня, а також для автономної роботи в якості пристроїв управління невеликими установками.

Для візуалізації і дистанційного управління процесом даний ПЛК підключений до операторської станції (АРМ оператора) за допомогою інтерфейсу MP²I. АРМ оператора реалізоване на базі SCADA-системи TRACE MODE 6. Для зв'язку ПЛК зі SCADA-системою використовується OPC-сервер фірми VIPA.

OPC (OLE for Process Control) – сімейство програмних технологій, що представляють собою єдиний інтерфейс для управління об'єктами автоматизації та технологічними процесами.

Інтерфейс користувача відкритого проекту в середовищі WinPLC V4 зображено на рис. 15.1. Створення програм в середовищі WinPLC7 є досить зручним та клопітким процесом.

Програмування ПЛК може відбуватися за допомогою використання трьох мов стандарту MEK (6) 1131-3: STL, FBD, LAD. STL – мова структурованого тексту, це текстова високорівнева мова загального призначення (по синтаксису схожа на Паскаль). FBD – графічна мова функціональних блоків, подібна до функціональних схем пристроїв. LAD – графічна мова, що є стандартизованим

варіантом мов релейно-контактних схем. Зазвичай програма написана на одній з мов легко переводиться на інші мови, проте не кожна програма написана на STL може бути відображена за допомогою FBD або LAD.

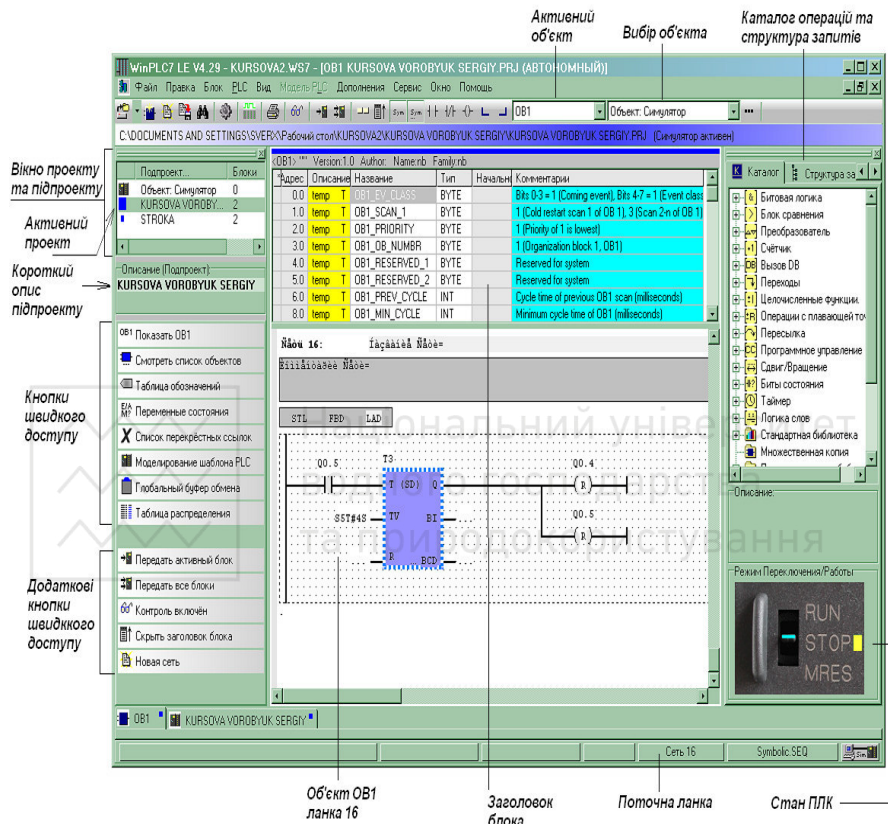


Рис. 15.1. Інтерфейс інструментального середовища WinPLC7

WinPLC7 має вбудований симулятор, що дає змогу розробляти та відлагоджувати програмний продукт без наявності апаратного забезпечення. В симуляторі реалізовано можливість під'єднання модуля цифрового вводу на 8 портів, цифрового вводу на 16 портів, двійково-десятькового вводу на 8 портів, двійково-десятькового вводу на 16 портів, аналогового вводу, цифрового виводу на 8 портів, цифрового виводу на 16 портів, двійково-десятькового виводу на 8



портів, двійково-десятькового виводу на 16 портів, аналогового виводу. Для аналогових модулів можна вибрати вхідний та вихідний діапазон вимірювання (за струмом та напругою) та задати адресу кожного модуля.

Використання симулятора є доцільним при розробці та відлагодженні логіки роботи програм та алгоритмів обробки внутрішніх змінних.

В лабораторній роботі використовується контролер VIPA 114-6BJ02, що відноситься до серії System 100V. Контролери даної серії можуть бути використані в:

- машинах для фасування і упаковки;
- верстатах різного призначення;
- машинах для харчової промисловості;
- машинах для легкої промисловості;
- конвеєрах, підйомниках, ліфтах;
- насосах, компресорах;
- системах сигналізації і охорони;
- системах обліку електроенергії;
- сучасних вимірювальних системах.

Серія System 100V побудована за модульним принципом. Це означає, що користувач має можливість оптимально підбирати склад модулів для вирішення своєї задачі і гнучко модифікувати його при розширенні або при зміні вимог до системи. Всі модулі введення-виводу і інтерфейсні модулі володіють властивістю універсальності, тобто їх можна застосовувати разом з будь-яким модулем CPU даної серії. При цьому є можливість вибору процесорного модуля з оптимальною продуктивністю для вирішення конкретного завдання. До складу сімейства входять декілька моделей контролерів з вбудованими каналами введення-виводу і з підтримкою функцій формування сигналів переривання, швидких лічильників і імпульсних виходів. Кількість каналів введення-виводу може бути збільшено за допомогою модулів розширення.

Також варто зазначити, що System 100V – єдине в світі сімейство процесорних модулів контролера такого класу, спеціально розроблених для роботи в мережі Profibus як ведені. Окрім "інтелектуальних" процесорних модулів в лінійці System 100V є широкий вибір непрограмованих модулів віддаленої периферії



Profibus DP і CANOpen (ведених) з інтегрованими дискретними і аналоговими каналами введення/виводу.

15.3. Програма роботи

1. Вивчити технологію приготування суміші з двох компонентів.
2. Вивчити будову і принцип дії лабораторної установки дозування рідин.
3. Побудувати структурну і функціональну схеми досліджуваної системи автоматизації процесу дозування рідин.
4. Ознайомитися з програмним забезпеченням ПЛК та операторської станції.
5. Навчитись змінювати склад суміші.
6. Дослідити точність дозування кожного компоненту.
7. Дослідити схему дозування переповнення резервуару.
8. Описати технологію процесу керування приготуванням суміші.

15.4. Опис лабораторної установки

Лабораторна установка (рис. 15.2) складається із резервуара дозованої рідини 1, резервуара для зважування заданої дози рідини 2, резервуара, в який проводиться дозування 3, насоса 4, електродних давачів рівня 5, електромагнітних клапанів 6 – 8, тензодавача 9, блока перетворення сигналу з тензодавача 10 (БПТ-2), релейних пристроїв 11, сигналізатора рівня рідини 12 (ОВЕН САУ-М6), блока живлення 13, вимикача 14, клемних колодок 15 – 16, програмованого логічного контролера 17 (VIPA 114) та операторської станції 18. Реле 11 призначені для підсилення керуючого сигналу з ПЛК за потужністю, оскільки піковий струм через клапан перевищує максимально допустимий для дискретного виходу ПЛК.

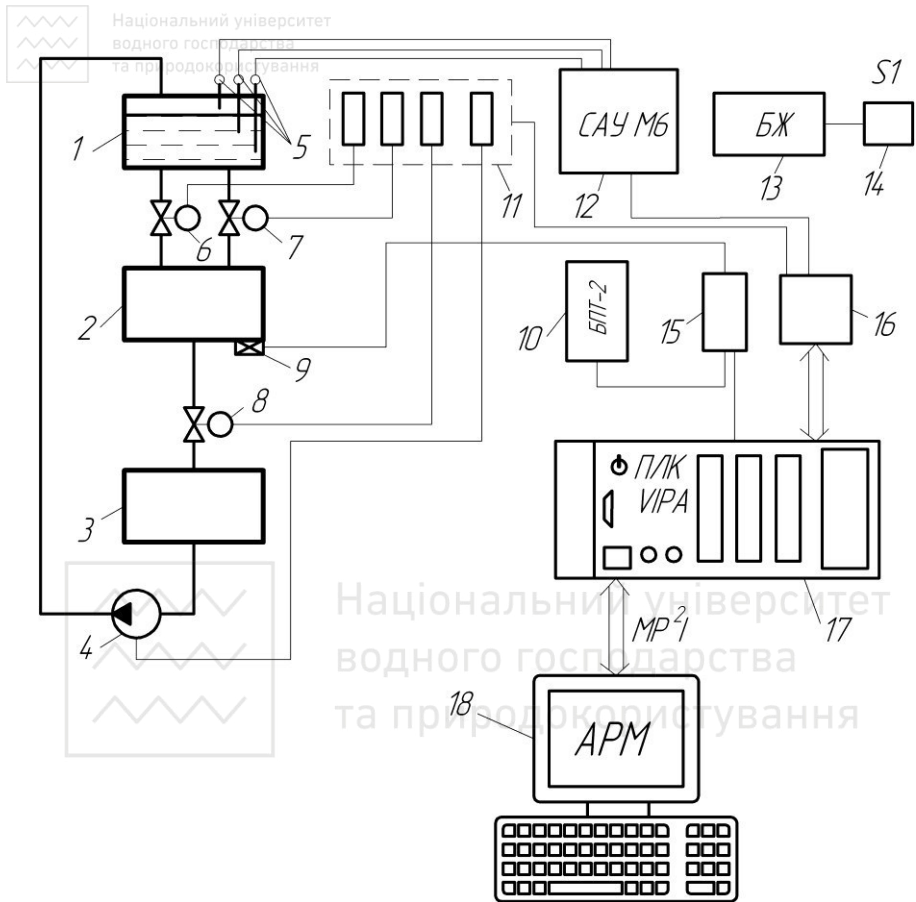


Рис. 15.2. Структурна схема лабораторної установки

Мнемосхема процесу дозування двох компонентів відображає вигляд головного екрану оператора (рис. 15.3). На виробництві після дозування отримана суміш подається в наступний технологічний апарат, проте в лабораторній установці він відсутній, тому отримана суміш перекачується насосом знову в резервуар дозованої рідини, який служить резервуаром і першого, і другого компонента.

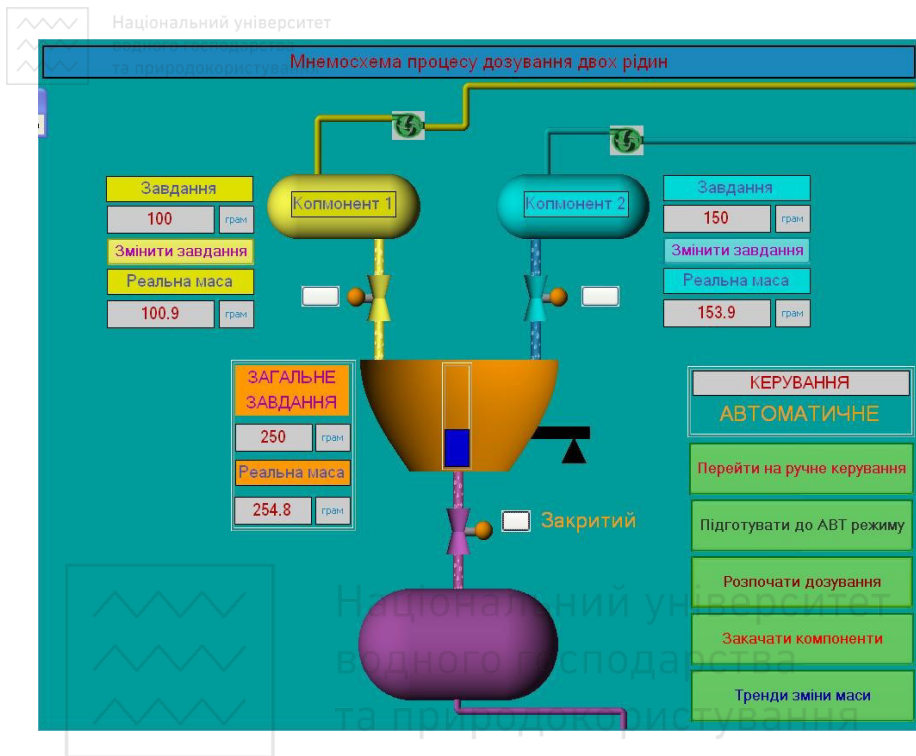


Рис. 15.3. Мнемосхема процесу дозування двох рідин

Лабораторний стенд дозування двох компонентів може працювати в двох режимах роботи – автоматичному та ручному. На головному екрані автоматизованого робочого місця відображено індикацію поточного режиму роботи та розташовано кнопку програмного перемикача режимів.

Після переведення роботи лабораторного стенду дозування двох компонентів в автоматичний режим оператор може змінити завдання першого та другого компонентів, за допомогою кнопок "Змінити завдання", виділених відповідним кольором. Натиснувши кнопку, потрібно ввести значення завдання маси в грамах та натиснути «Готово». Після натиснення «Готово» значення маси передається зі SCADA-системи OPC-серверу, який в свою чергу передає значення маси в ПЛК. ПЛК записує отримане значення у відповідний регістр пам'яті даних. За замовчуванням завдання на дозування першого компоненту встановлене 150 грамів та другого



компоненту - 200 грамів. Автоматичне дозування розпочинається після натиснення кнопки “Розпочати дозування”. Відразу після цього відкривається перший клапан і відбувається дозування компоненту з бункера, в якому він знаходиться, на вагу. Вимірювання й індикація навантаженої на вагу маси відбувається неперервно в часі за сигналом з тензодавача. ПЛК порівнює значення маси, отримане з тензодавача, із завданням маси та, коли перше стане рівним або перевищить друге, видає сигнал на закриття клапана. Через дискретний вихід, підключений до обмотки реле, припиняє текти струм. Контакти реле розмикаються і з електромагнітного клапана зникає напруга, що спричиняє його закриття. У програмі ПЛК вмикається таймер затримки дозування другого компоненту. Після закінчення дозування першого компоненту програмований логічний контролер запам’ятовує остаточну масу та передає її на АРМ, де відбувається її індикація. Коли пройде час затримки, вихід таймера встановлюється в логічну одиницю подається сигнал на відкриття другого клапана. Сигнал з тензодавача з цього моменту несе інформацію про сумарну масу обох компонентів, тому він порівнюється із сумою завдань першого та другого компонентів. Коли маса на вазі перевищить суму завдань, коло живлення електромагніта клапана розмикається та він перекриває подачу другого компоненту. Вмикається таймер затримки вивантаження. По закінченні часу, заданому в таймері, відбувається автоматичний злив надозованої суміші компонентів подачею напруги на електромагніт клапана зливу 8. Коли маса на дозаторі стане меншою 25 г, вмикається таймер затримки закриття клапану зливу суміші, величина затримки вибирається такою, щоб з резервуара 2 встигла витекти вся рідина. Переглянути графіки зміни маси кожного з компонентів та загальне значення зміни маси на вазі можна натиснувши кнопку “Тренди зміни маси”. При цьому на моніторі відобразиться екран із відповідними трендами (рис. 15.4).

За їх допомогою можна визначити зміну маси кожного компоненту в часі, зміну маси рідини на вазі в часі та тривалість виконання дозування. Графіки зміни маси кожного з компонентів зображені на одному тренді, позначені різними кольорами, для того щоб було легко відслідкувати початок та кінець дозування кожного з компонентів. Графік загальної зміни маси при цьому буде сумою графіків зміни кожного з компонентів. Для заповнення ємностей з



яких відбувається дозування служить кнопка “Закачати компоненти” (рис. 15.3).

В ручному режимі керування дозуючими кнопками та клапаном зливу відбувається за допомогою розташованих біля них кнопочних вимикачів. Індикація стану клапанів реалізована із використанням зміни кольору, динамічних зображень та надписів. На екрані також відображено загальне завдання маси, що являє собою суму завдань двох компонентів та реальну масу цієї суми (рис. 3). Кнопка “Підготувати до АВТ режиму” служить для перекачки надозованої рідини із ємності в яку скидається в ємність із якої вона здозовується. Перебіг процесу перекачування триває певний визначений в програмі час, або робота насоса блокується при досягненні верхнього електрода в резервуарі дозованої рідини. Блокування – фіксація робочих частин апаратів та механізмів у певному положенні, яке зберігається незалежно від того, усунуто чи ні блокуючий вплив.

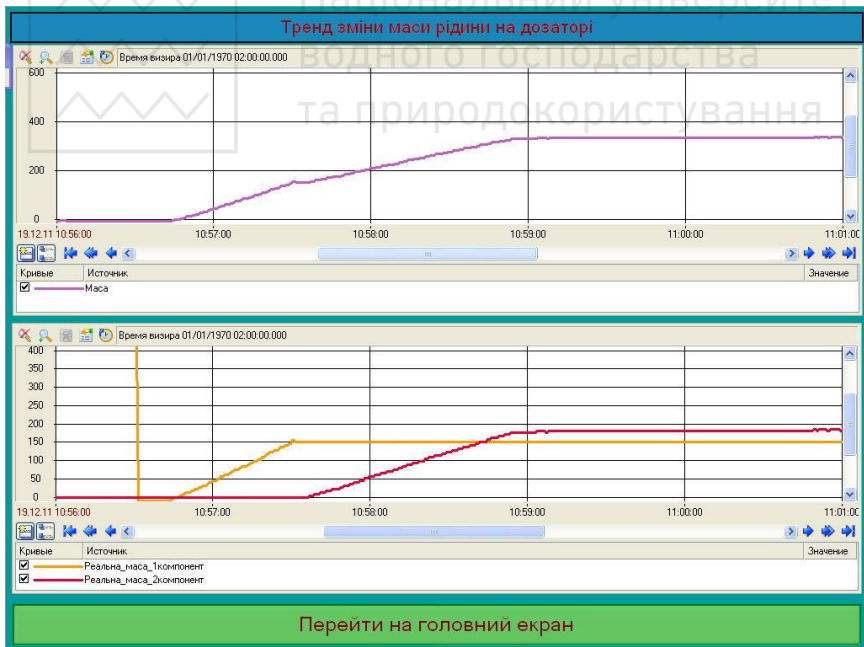


Рис. 15.4. Тренди зміни маси рідин на дозаторі



Алгоритм роботи програми в автоматичному режимі починається із масштабування змінної та перетворення масштабованого значення у фізичну величину (рис. 15.5). Далі програма запам'ятовує завдання на дозування компонентів. Програма постійно перевіряє чи надійшов сигнал на запуск дозування. Якщо така команда надійшла перевіряється умова при якій надозується перший компонент. Відповідно до неї змінюється стан 1-ого клапана. Далі програма перевіряє умову дозування 2-ого компоненту та відбувається його надозування відповідно до встановленої маси. При цьому програма керує станом 2-ого клапана. Після закінчення дозування обох компонентів відраховується час записаний в таймері T2, після якого запускається злив надозованої суміші у підготовлену для неї ємність. Про закінчення процесу зливу сигналізує зміна маси на дозаторі менше 25 грамів. При цьому програма відраховує час необхідний для повного спустошення дозатора та закриває клапан зливу. Після закриття клапану зливу програма готова до сприйняття нового завдання та відпрацювання нового циклу дозування в автоматичному режимі.

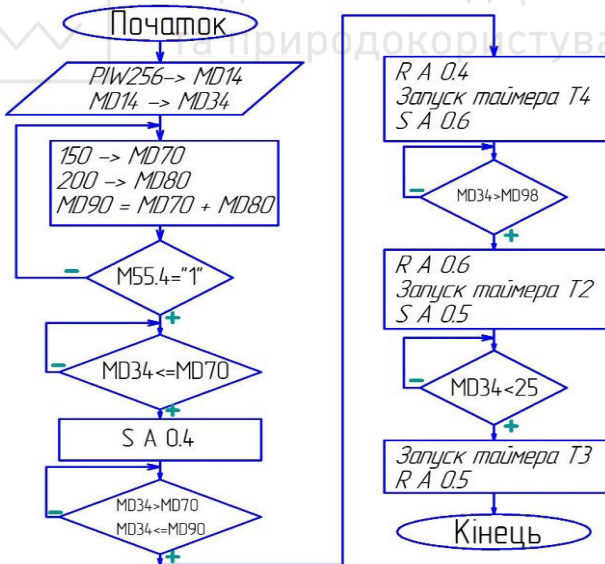


Рис. 15.5. Блок-схема алгоритму роботи програми стану дозування в автоматичному режимі

Змінні та позначення на блок-схемі: PIW256 – апаратна адреса аналогового входу, підключеного до виходу БПТ-2; MD14 – масштабоване значення змінної, апаратне значення маси; MD34 – поточне значення маси на дозаторі; MD70 – завдання по замовчуванню на дозування першого компоненту; MD80 - завдання по замовчуванню на дозування другого компоненту; MD90 – сума завдань на дозування; MD98 – сума завдань на дозування компонентів із врахуванням похибки при дозуванні першого компоненту; M55.4 – змінна яка відповідає за запуск процесу дозування; S A 0.4 – відкриття клапану дозування першого компоненту; R A 0.4 – закриття клапана 1; S A 0.6 – відкриття клапана 2; R A 0.6 – закриття клапана 2; S A 0.5 – відкриття клапана зливу; R A 0.5 – закриття клапана зливу.

15.5. Порядок виконання роботи

1. Ознайомитися з будовою, принципом дії лабораторної установки та призначенням всіх її елементів.
2. Увімкнути комп'ютер, завантажити інтегроване середовище SCADA-системи TRACE MODE 6 та відкрити збережений проект.
3. Перевірити налаштування OPC-сервера. Для цього в меню «Програми» вибрати *VIPA OPC Edit* та в меню відкриття файлу обрати поточну конфігурацію серверу. Скопіювати в звіт з лабораторної роботи налаштування змінних OPC-сервера. Запустити монітор реального часу TRACE MODE, натиснувши кнопку .
4. З дозволу викладача увімкнути установку.
5. Натиснути кнопку "Змінити завдання" на мнемосхемі АРМ оператора та задати значення дози першого компоненту відповідно до таблиці 15.1.
6. Натиснути кнопку "Розпочати дозування".
7. Спостерігати за процесом дозування до повторення циклу. Опис процесу занести до звіту з лабораторної роботи.
8. Зкопіювати в звіт мнемосхему і графік процесу дозування.
9. Змінюючи значення завдання, провести іще 3 дозування та заповнити таблицю 15.1.
10. Дослідити блокування ввімкнення насоса подачі рідини в



резервуар дозованої рідини при досягненні верхнього рівня. Для цього натисніть кнопку «Підготувати до АВТ режиму» та почекайте, доки рівень в резервуарі досягне верхнього електрода. Спробувати в ручному режимі ввімкнути насос кнопкою «Закачати компоненти». Переконавшись, що насос не вмикається, здійснити ще одне дозування, щоб рівень в резервуарі зменшився. Повторно спробувати включити насос в ручному режимі.

11. Завантажити середовище програмування WinPLC7, обравши в меню «ПУСК» *Програми – VIPA GmbH – VIPA WinPLC7*.
12. Відкрити програму для даної лабораторної установки (Пункт меню *File - Open*), написану мовою драбинкових діаграм (LAD), ознайомитися з призначенням блоків у програмі та їх функціями.
13. Побудувати функціональну схему досліджуваної системи автоматизації процесу дозування рідин. Описати принцип дії лабораторної установки.
14. Оформити звіт про виконання роботи, вказавши фактори, що впливають на точність дозування у даній роботі. Зробити висновки.

Таблиця 15.1

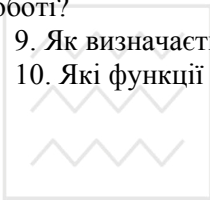
Дослідження точності дозування рідини

№ п/п	Задання маси компонента		Маса отриманої дози компонента, г		Абс. похибка, г		Відн. похибка, %	
	1	2	1	2	1	2	1	2
1	m_{11}	m_{21}						
2	m_{12}	m_{22}						
3	m_{13}	m_{23}						
4	m_{14}	m_{24}						



15.6. Контрольні запитання

1. Яке призначення OPC-сервера?
2. Які мови програмування контролерів підтримує середовище WinPLC7?
3. Визначити яку мінімальну зміну вхідного струмового сигналу в діапазоні 0-20 мА може розрізнити аналоговий вхід із розрядністю 12 біт.
4. Для чого використовується симулятор в середовищі WinPLC7?
5. Для яких задач зазвичай використовуються контролери серії System 100V?
6. Який принцип лежить в основі побудови серії VIPA System 100V і що він означає?
7. Для чого призначений блок САУ- М6?
8. Від чого залежить точність дозування в даній лабораторній роботі?
9. Як визначається загальне завдання маси рідин на дозаторі?
10. Які функції автоматичного блокування в даній системі?





Література

1. Пальчевський Б.О. Автоматизація технологічних процесів: Навч. посіб.-Львів: Світ,2007. - 392с.
2. Бородин И.Ф., Судник Ю.А. Автоматизация технологических процессов: Учебник. - Москва: КолосС, 2005. - 344с.
3. А.О.Бобух. Автоматизовані системи керування технологічними процесами. Нав.посібник. – Харків ХНАМГ, 2006. - 186 с.
4. Ключев А.С. и др. Проектирование систем автоматизации технологических процессов. Справочное пособие. – М.: Энергия, 1980.
5. ГОСТ 21.408-93. Правила виконання робочої документації автоматизації технологічних процесів
6. Програмування промислових контролерів у середовища Unity Pro. Пупена О.М., Ельперін І.В. Навч. посібник.,— К.: Видавництво Ліра . – К. . – 2013. . – 340с.
7. Клепач М.І. Методичні вказівки до виконання курсового проекту з курсу “Автоматизація періодичних технологічних процесів”. – Рівне: УДУВГП, 2002. – 15 с.