

Міністерство освіти і науки України
Запорізький національний університет

Пупена О.М., Ельперін І.В., Ніколаєнко А.М.

Програмування промислових контролерів

*Методичні вказівки до лабораторного практикуму для студентів ЗНУ
спеціальності 151
“Автоматизація та комп’ютерно-інтегровані технології”*

*Рекомендовано для видання
на засіданні кафедри АУТП,
протокол № від 2020р.*

Запоріжжя
2020

Програмування промислових контролерів: Методичні вказівки до лабораторного практикуму для студентів спеціальності 151“Автоматизація та комп’ютерно-інтегровані технології”./Уклад.:О.М.Пупена, І.В.Ельперін, А.М.Ніколаєнко 2020. — 60 с.

Укладачі: О.М. Пупена, канд. техн.наук
І.В. Ельперін, канд.техн.наук
А.М.Ніколаєнко, канд. техн.наук

Відповідальний за випуск
зав. каф. АУТП ЗНУ д. т. н., проф. Пазюк М.Ю.

ЛАБОРАТОРНА РОБОТА № 1

Конфігурування ПЛК і робота зі змінними I/O.

Тривалість: 2 акад. години .

Мета: ознайомлення з середовищем UNITY PRO, принципами конфігурування ПЛК M340, принципами створення змінних та їх локалізацією.

Лабораторна установка.

Програмне забезпечення. UNITY PRO V \geq 4.0.

Загальна постановка задачі. Необхідно створити конфігурацію ПЛК M340 відповідно до поставленої задачі.

Послідовність виконання роботи.

Перед виконанням лабораторної роботи ознайомтесь з додатком 1.

1) Запустити на виконання UNITY PRO. Створити новий проект для M340 VMXR342000.

3) Створити конфігурацію ПЛК та його модулів, відповідно до рис.1.1 та таблиці 1.1.

4) Змінити конфігурацію процесорного модуля, відповідно до рис. Д1.5.

5) Проконтролювати розподіл споживаної потужності для всіх модулів живлення. Зробити висновки про необхідність заміни модуля живлення.

6) Створити локалізовані змінні, прив'язані до вхідних та вихідних каналів, відповідно до рис.1.2 (Project Browser->Variables & FB instances->Elementary Variables). Після створення, знайти кожен змінну у відповідних до каналів полях "Symbol" вікон конфігурації модулів.

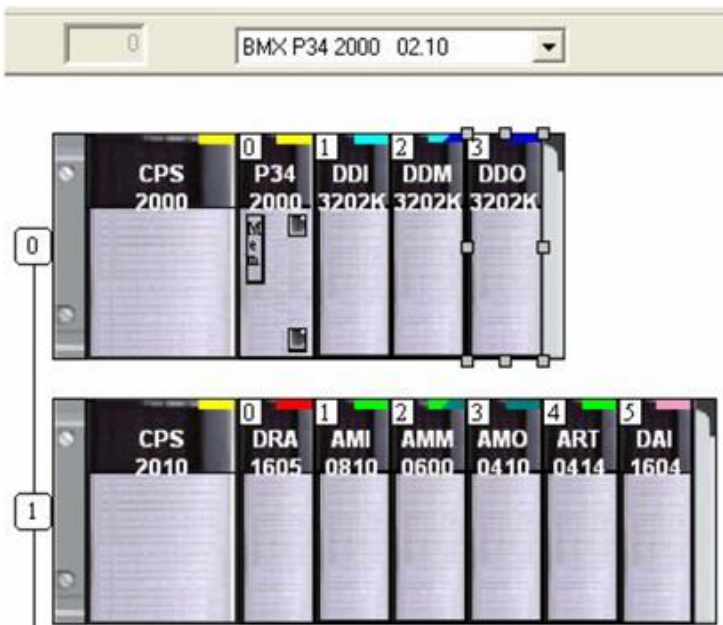


Рис.1.1 Компонування M340

| Name | .. | Ad.. |
|-------------|-------|----------|
| VR1 | INT | %QW1.2.4 |
| M1 | EBOOL | %Q0.3.13 |
| V1 | EBOOL | %Q0.2.31 |
| TE1 | INT | %IW1.4.0 |
| StopSystem | EBOOL | %I0.2.13 |
| StartSystem | EBOOL | %I0.1.5 |
| TIC1_SP | REAL | |
| TIC1_FV | REAL | |
| TIC1_OUT | REAL | |
| TIC1_MAN | BOOL | |

Рис 1.2 Створення локалізованих змінних I/O

Таблиця 1.1.

| Поз. мод | Канали/ модуль | Назва властивості | Значення | Коментар | |
|----------|---------------------|---------------------|-----------------------------|--|--|
| 1 | 2 | 3 | 4 | 5 | |
| 0.1 | BMX DDI3202K | | | | |
| | 0-15 | Function | Discrete Inputs | канали використовуються | |
| | | Task | MAST | опитування каналів з задачею MAST | |
| | 16-23 | Function | Discrete Inputs | канали використовуються | |
| | | Task | FAST | опитування каналів з задачею FAST | |
| 24-31 | Function | None | канали не використовуються | | |
| 0.2 | BMX DDM3202K | | | | |
| | 0-15 (вхідні) | Всі властивості | за замовченням | | |
| | 16-23 (вихідні) | Function | Discrete Outputs | канали використовуються | |
| | | Task | MAST | запис виходів в кінці задачі MAST | |
| | | Supply Monitoring | опція виставлена | контроль живлення каналів активований, при виході напруги за межі буде сигналізуватися помилка | |
| | | Reactivate | Programmed | реактивація виходів при короткому замиканні (КЗ) відбувається програмним шляхом | |
| | | Fallback Mode | Maintain | при зупинці контролера, стан виходів залишається без змін | |
| | 24-31 (вихідні) | Function | Discrete Outputs | канали використовуються | |
| | | Task | MAST | запис виходів в кінці задачі MAST | |
| | | Supply Monitoring | опція виставлена | контроль живлення каналів активований, при виході напруги за межі буде сигналізуватися помилка | |
| | | Reactivate | Automatic | реактивація виходів при короткому замиканні відбувається кожні 10 с до зникнення помилки КЗ | |
| | | Fallback Mode | Fallback | при зупинці ПЛК, вихідні канали перейдуть в стан, визначений властивістю Fallback Value | |
| | | Fallback Value | 0 | при зупинці ПЛК, вихідні канали перейдуть в стан "0" | |
| | 0.3 | BMX DDO3202K | | | |
| | | 0-31 | Всі властивості | за замовченням | |
| 1.0 | BMX DRA1605 | | | | |
| | 0-15 | Всі властивості | за замовченням | | |
| 1.1 | BMX AMI0810 | | | | |
| | 0-3 | Task | MAST | опитування каналів з задачею MAST | |
| | | Cycle | Normal | модуль проводить повну обробку всіх каналів | |
| | | Used Range | опція виставлена 0...10V | канали використовуються (обробляються) вибраний діапазон вхідного сигналу 0-10В | |
| | | Scale | за замовченням | для цих каналів діапазон масштабування за замовченням 0-10000 | |
| | | Filter | 1 | мала ступінь фільтрації | |
| | | | | | |
| 1 | 2 | 3 | 4 | 5 | |

| | | | | | |
|----------------|--------------------|--------------------|---|--|-----------------------------------|
| 1.1 | 4-7 | Task | FAST | опитування каналів з задачею FAST | |
| | | Cycle | Normal | модуль проводить повну обробку всіх каналів | |
| | | Used | опція виставлена | канали використовуються (обробляються) | |
| | | Range | 1...5V/4...20mA | вибраний діапазон вхідного сигналу 4-20mA | |
| | | Scale | за замовченням | для цих каналів діапазон масштабування за замовченням 0-10000 | |
| | | Filter | 0 | без фільтрації сигналу | |
| 1.2 | BMX AMM0600 | | | | |
| | 0-3 (вхідні) | Task | MAST | опитування каналів з задачею MAST | |
| | | Cycle | Fast | модуль обробляє тільки канали з виставленою опцією Used, фільтрація не використовується | |
| | | Used | опція виставлена тільки для каналів 0,1 | вхідні канали 0 та 1 використовуються (обробляються), канали 2 та 3 – не використовуються (відключені) | |
| | | Range | 0...10V | вибраний діапазон вхідного сигналу 0-10V | |
| | | Scale | за замовченням | для цих каналів діапазон масштабування за замовченням 0-10000 | |
| | | Filter | не має значення | фільтрація не використовується, так як вибраний Cycle=Fast | |
| | 4-5 (вихідні) | Task | MAST | запис виходів в кінці задачі MAST | |
| | | Range | 4...20mA | вибраний діапазон вихідного сигналу 4-20mA | |
| | | Scale | за замовченням | для цих каналів діапазон масштабування за замовченням 0-10000 | |
| | | Fallback | опція знята | при зупинці контролера, стан виходів залишається без змін | |
| | | Wiring CTRLC | опція знята | модуль не контролює обрив, перевантаження або КЗ на вихідних каналах | |
| | 1.3 | BMX AMO0410 | | | |
| | | 0-3 | Task | MAST | запис виходів в кінці задачі MAST |
| Range | | | 4...20mA | вибраний діапазон вихідного сигналу 4-20mA | |
| Scale | | | за замовченням | для цих каналів діапазон масштабування за замовченням 0-10000 | |
| Fallback | | | опція виставлена | при зупинці контролера, стан виходів визначається | |
| Fallback Value | | | 5000 | при зупинці контролера, на виходах буде виставлено 50% від діапазону | |
| Wiring CTRLC | | | опція виставлена | модуль контролює обрив, перевантаження або КЗ на вихідних каналах та сигналізує при цих несправностях | |
| | | | | | |

| 1 | 2 | 3 | 4 | 5 |
|-----|--------------------|-----------------|-------------------|---|
| 1.4 | BMX ART0414 | | | |
| | 0-3 | Cold Junction | Internal Telefast | компенсація холодних кінців для термопар проводиться в модулі Telefast, для термометрів опору властивість не має значення |
| | | Rejection | 50Hz | модуль буде фільтрувати гармонійні паразитні завади, створені живленням частотою 50 Гц |
| | | Used | опція виставлена | канали використовуються (обробляються) |
| | | Range | Pt100...(3 wires) | використовується термометр опору Pt100 з 3-провідною схемою підключення |
| | | Scale | 1/10°C | 1 = 0.1°C; -175°C...825°C -> -1750...8250 |
| | | Filter | 0 | без фільтрації сигналу |
| 1.5 | BMX DA1604 | | | |
| | 0-16 | Всі властивості | за замовченням | |

Перевірка виконання роботи та питання до захисту.

Викладачем перевіряється виконання поставленого завдання.

Студент повинен пояснити призначення кожного пункту виконаного завдання.

1. З яких основних конструктивних елементів складається M340?

2. Яку максимальну кількість шасі підтримує M340? Яким чином об'єднуються між собою шасі M340, як адресуються шасі та модулі в ньому?

3. Які способи підключення до дискретних та аналогових модулів є в M340?

4. По яким характеристиками відрізняються між собою процесорні модулі M340?

5. Прокоментуйте призначення кожного розділу проекту в UNITY PRO. Де проводиться конфігурування апаратного забезпечення?

6. Прокоментуйте настройки параметрів процесорного модуля на прикладі рис.Д1.5.

7. Прокоментуйте яким чином в середовищі UNITY PRO можна проконтролювати споживану потужність по кожному виходу модуля живлення.

8. Які задачі можуть функціонувати в M340? Як канали прив'язуються до конкретної задачі? Покажіть на прикладі.

9. Яким чином для аналогових модулів вказується діапазон сигналу та масштабування? Покажіть на прикладі одного вхідного та вихідного модуля.

10. Яким чином для вихідних модулів вказується значення виходу при зупинці ПЛК? Покажіть на прикладі дискретного та аналогового вихідних модулів.

11. Яким чином функціонує захист при КЗ та перевантаженні для дискретних захищених вихідних модулів? Продемонструйте як конфігурується ре активація виходів.

12. Яким чином конфігурується фільтрація для аналогових вхідних сигналів?

13. Яким чином адресуються канали вводу/виводу? Прокоментуйте рис.1.2.

ДОДАТОК 1. Робота в середовищі UNITY PRO.

Д1.1. Створення проекту та навігація в ньому. При створенні нового проекту (File->New), середовище UNITY PRO пропонує вибрати тип ПЛК (М340/Premium/Quantum) та процесорний модуль. У вікні вибору (рис.Д1.1) доступні короткі описи модулів та версія операційної системи (ОС). Для доступу до вибору процесорних модулів з старшими ОС необхідно виставити опцію "Show all versions".

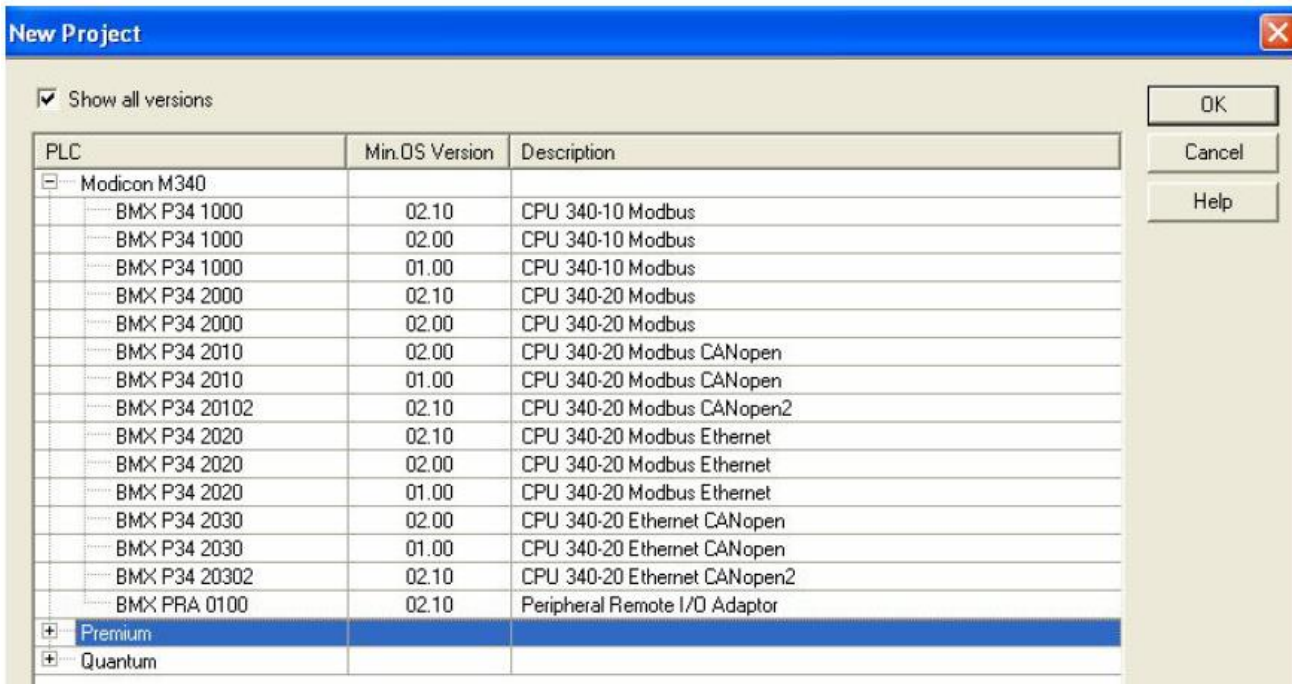


Рис.Д1.1. Вікно вибору типу ПЛК та процесорного модуля.

Проект UNITY PRO включає: конфігурацію апаратного та програмного забезпечення ПЛК, програму користувача, всі похідні типи даних, анімаційні таблиці (для перегляду та зміни значень змінних в табличному вигляді) та анімаційні екрани (для перегляду та зміни значень змінних в графічному вигляді), документацію до проекту.

Навігацію за проектом здійснюється з використанням *Провідника Проекту - Project Browser* (Tools->Project Drowser). На рис.Д1.2 показаний зовнішній вигляд Провідника Проекту в структурному вигляді.

Д1.2. Апаратна конфігурація шасі. Для створення апаратної конфігурації М340, необхідно вибрати подвійним кліком пункт PLCbus в розділі Configuration. При цьому відкриється вікно апаратної конфігурації (рис.Д1.3) та опціонально Hardware Catalog. Додавлення та видалення модулів може проводитися різними варіантами: через пункти контекстного меню позиції модуля, подвійним кліком по позиції модуля, або перетягуванням необхідного модуля в потрібну позицію із Hardware Catalog.

Д1.3. Апаратна конфігурація модулів. До вікна конфігурації модуля та його каналів можна досягти шляхом подвійного кліка по самому модулю. Конфігурація самого модуля проводиться шляхом вибору назви модуля у вікні

конфігурації (рис.Д1.4(а)), а каналів — шляхом вибору групи каналів (рис.Д1.4(б)). Зокрема для всіх модулів у вікні конфігурації доступна загальна інформація про модуль (вкладка Overview), а також перелік об'єктів вводу/виводу, які доступні в даному модулі (вкладка I/OObjects).

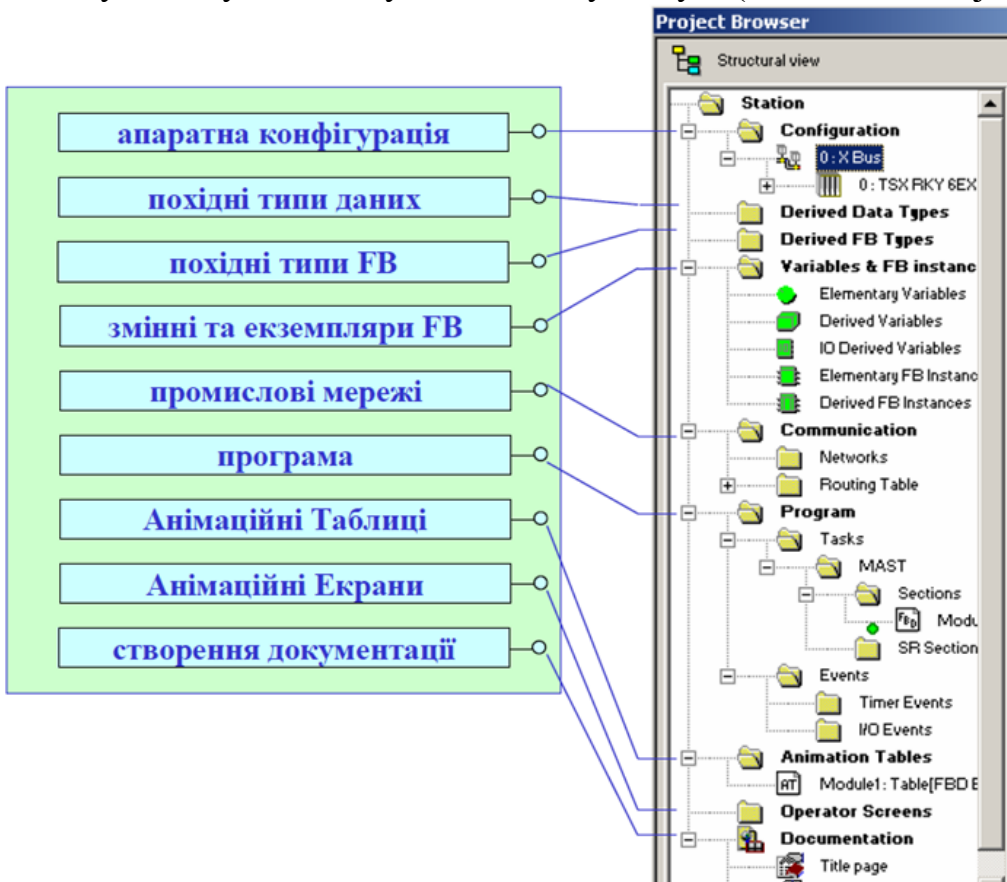


Рис. Д1.2 Структура проекту

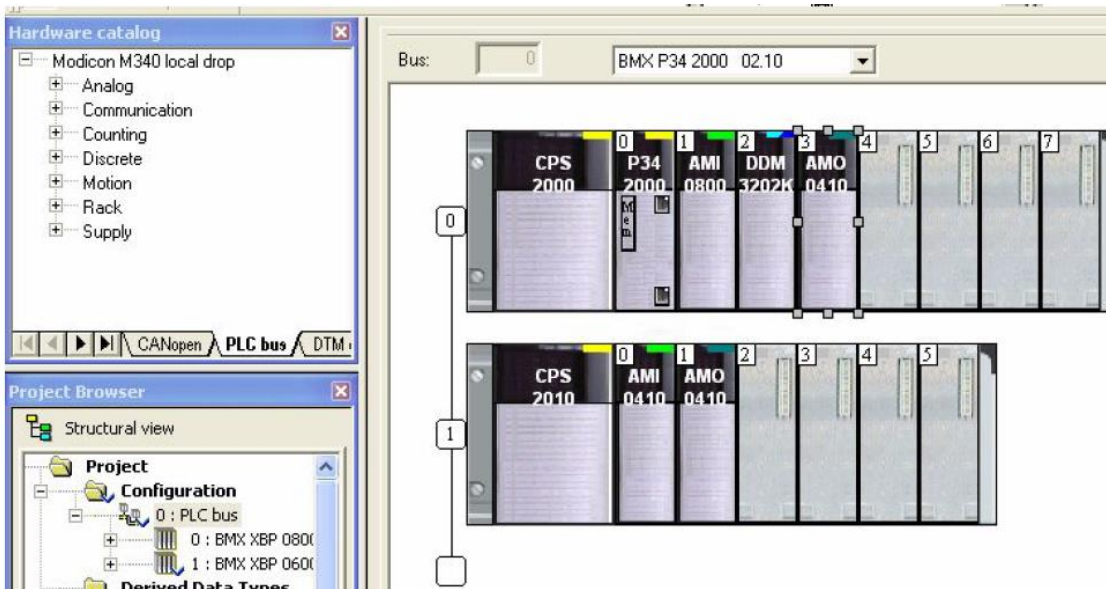


Рис. Д 1.3 Вікно апаратної конфігурації

Конфігураційні властивості каналів залежать від типу модуля та вибраних в ньому каналів. На рис.Д1.4(б) показані вихідні канали, починаючи з 16-го, для змішаного дискретного модуля BMX DDM 32D2K.

Для цих типів каналів вказується використовувана функція (Function), прив'язка опитування каналів до задачі (Task), необхідність контролю за живленням каналів (Supply Monitoring), спосіб реактивації виходів (Reactivate), режим виставлення виходів при зупинці ПЛК (Fallback Mode). Для інших типів каналів настройки будуть відрізнятися.

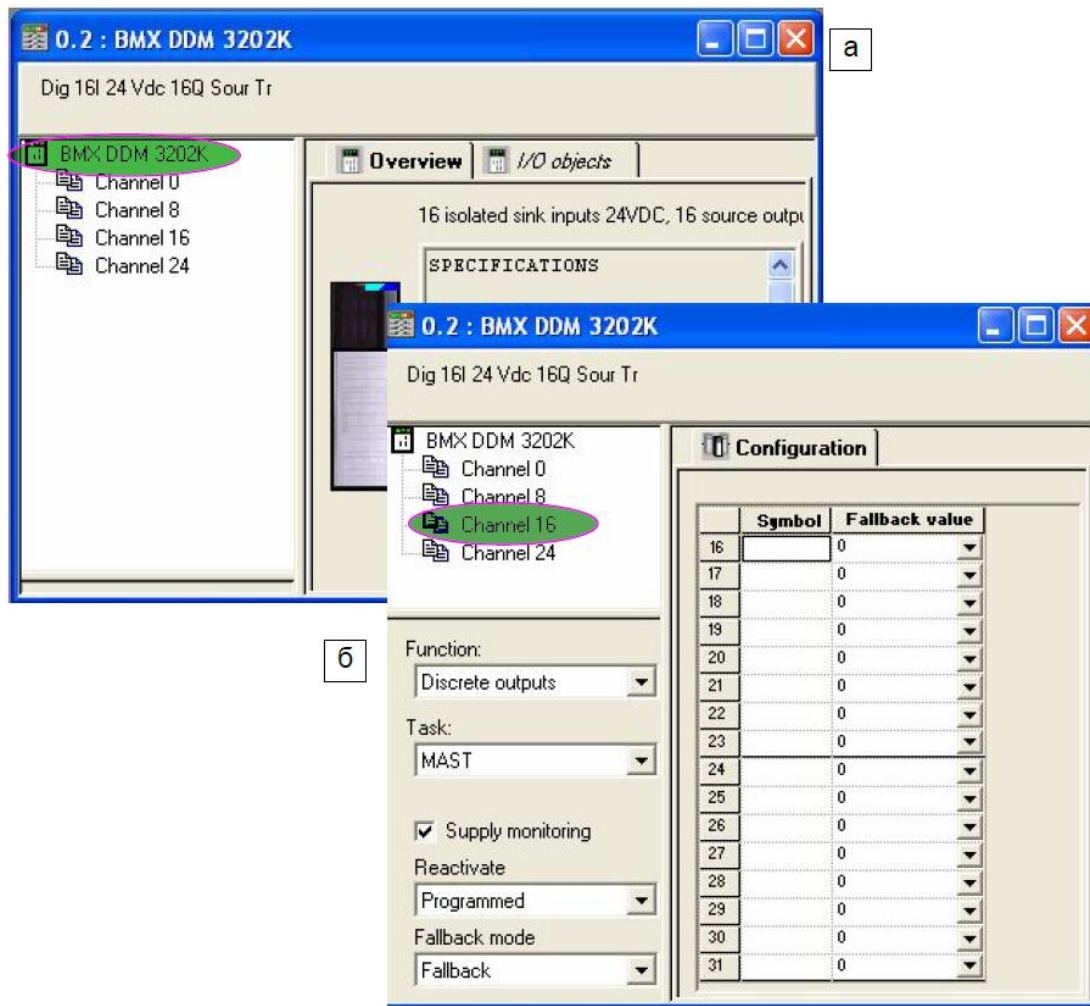


Рис. Д1.4. Вікно конфігурації модуля

Д1.4. Апаратна конфігурація процесорного модуля.

Для процесорного модуля, крім вкладок Overview та I/O Objects доступні вкладки Animation та Configuration. Вкладка Animation використовується в онлайн режимі для можливості діагностування, зміни дати та часу годинника реального часу ПЛК, а також для його ідентифікації. На вкладці Configuration проводиться конфігурація операційного режиму (Operating Mode) процесорного модуля а також пам'яті, яка виділяється під локалізовані дані (%M, %MW, %KW). Для M340 доступні наступні настройки операційного режиму (рис.Д1.5):

- Run/Stop Input: при активації даної опції, ПЛК буде знаходитись в RUN, якщо вхідна змінна, яка вказана в полі адреси напроти опції, будедорівнювати "1", в іншому випадку ПЛК буде в STOP;
- Memory Protect: при активації опції, запис на карту пам'яті можливий тільки якщо вхідна змінна, яка вказана в полі адреси напроти опції, буде дорівнювати "1".
- Automatic start in Run: при активації опції, ПЛК після вмикання живлення автоматично переводиться в режим RUN.
- Initialize %MWi on cold start: при активації опції змінні %MW будуть обнулятися при включенні ПЛК.

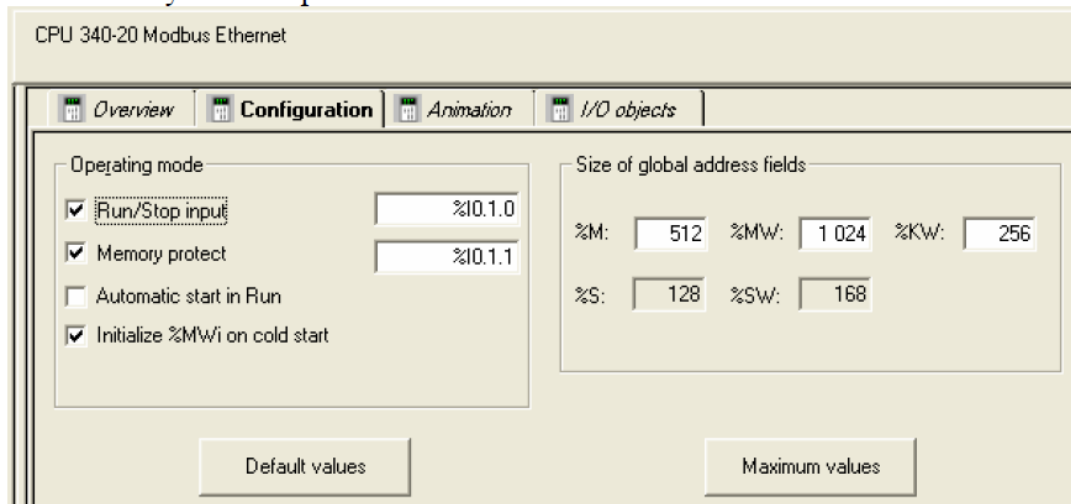


Рис. Д1.5. Вікно конфігурації процесорного модуля

Д1.5. Розподіл споживаної потужності модулями живлення.

Для модуля живлення доступне вікно розподілу споживаної потужності по всім типам живлення: внутрішніх 24VR та 3,3V, а також зовнішніх датчиків 24V. Вікно доступне через контекстне меню модуля живлення, пункт "Power Supply and IO Budget"(рис.Д1.6).

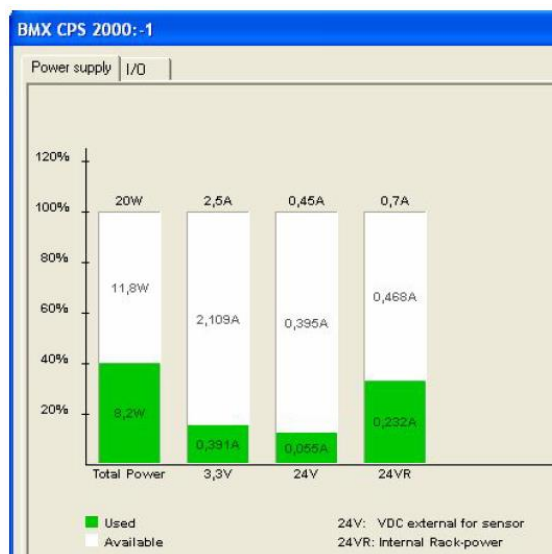


Рис.Д1.6. Вікно розподілу споживної потужності

ЛАБОРАТОРНА РОБОТА № 2

Основи програмування ПЛК в середовищі UNITY PRO

Тривалість: 2 акад. години

Мета: ознайомлення з принципами створення та налагодження програм користувача для ПЛК М340 в середовищі UNITY PRO мовами ST та LD.

Лабораторна установка.

Перед виконанням лабораторної роботи ознайомтесь з додатком 2.

Програмне забезпечення. UNITY PRO V \geq 4.0.

Загальна постановка задачі. Необхідно створити змінні та програму користувача для ПЛК М340 відповідно до наступної задачі(рис.2.1). При натисканні кнопки ПУСК відкривається клапан набору першого продукту. Після досягнення середнього рівня клапан 1-го продукту закривається, відкривається клапан набору 2-го продукту. Після спрацювання сигналізатору верхнього рівня включається клапан зливу. Після відключення сигналізатору нижнього рівня, цикл повторюється у випадку якщо кнопка СТОП не натиснута. Якщо СТОП натиснута клапан зливу закривається.

Написати програму імітатора роботи рівня в окремій секції мовою LD. Логіку виконання програми реалізувати в окремій секції мовою ST. Програму перевірити та відлагодити з використанням таблиць анімацій та операторських екранів.

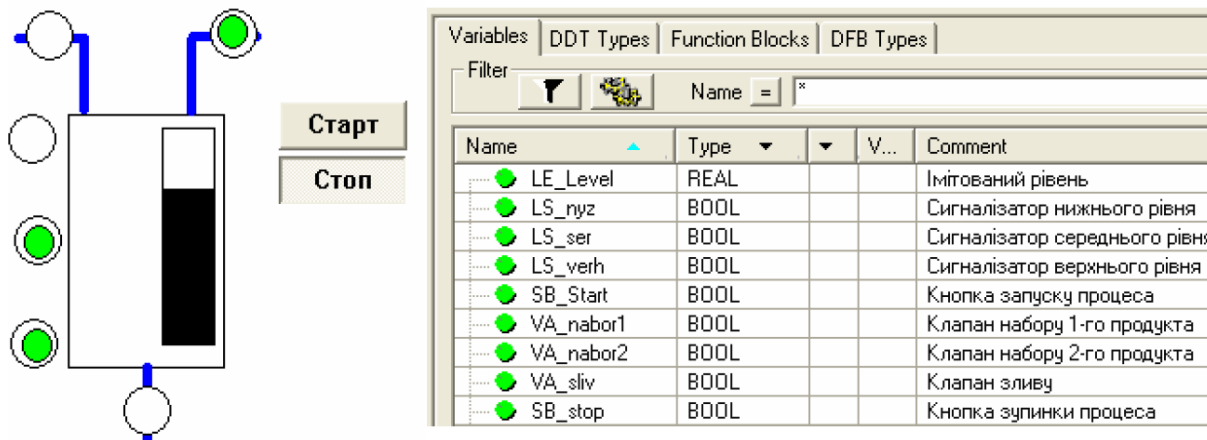


Рис.2.1. Приклад операторського екрану (зліва), та змінні (з права) до поставленої задачі.

Послідовність виконання роботи.

1) Запустити на виконання UNITY PRO. Створити новий проект з ПЛК М340 будь-якої конфігурації.

2) В редакторі даних проекту створити нелокалізовані змінні, відповідно до задачі, як показано на рис.2.1.

3) Створити секцію в задачі MAST мовою LD з назвою "ImitationL". Додаткові властивості для секції не вказувати. Створити в секції програму імітації рівня як на рис.2.2.

4) Викликати команду Analyze і перевірити проект на синтаксичні

помилки.

У випадку наявності помилок, виправити їх в проекті. При відсутності помилок зробити повну компіляцію проекту і в on-line режимі записати програму в Імітатор ПЛК. Під час процедури звернути увагу на стан індикаторів панелі статусу.

5) Створити Анімаційну таблицю зі змінними. Перевірити роботу програми імітації шляхом зміни значень клапанів. Подивитися на роботу редактору LD в анімаційному режимі.

6) Не виходячи з on-line режиму, створити секцію в задачі MAST на мові ST з назвою "ControlL". Додаткові властивості для секції не вказувати. Створити в секції програму логіки управління як на рис.2.3.

7) Компілювати зміни в проекті. В анімаційній таблиці перевірити роботу програми.

8) Створити операторський екран як на рис.2.1. Перевірити роботу програми з використанням операторського екрану.

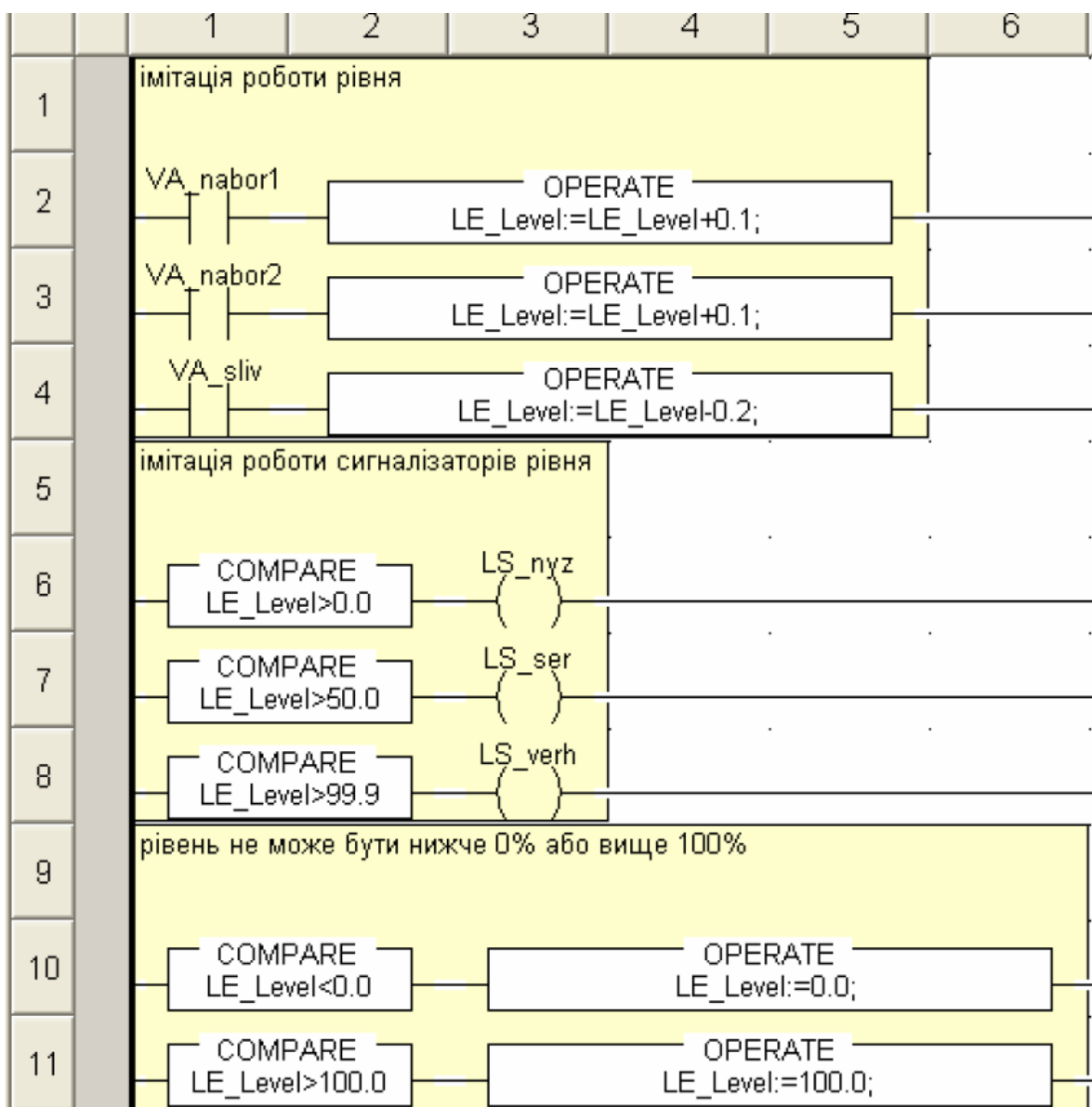


Рис.2.2. Секція імітації роботи рівня LD-мовою


```

if SB_start and not LS_nyz then (*якщо команда ПУСК і емність порожня*)
    VA_nabor1:=true;          (*відкрити КЛ набору 1*)
end_if;
if LS_ser and VA_nabor1 then  (*якщо спрацював датчик сер. рівня*)
    VA_nabor1:=false;        (*закрити КЛ набору 1*)
    VA_nabor2:=true;        (*відкрити КЛ набору 2*)
end_if;
if LS_verh then              (*якщо спрацював датчик вер. рівня*)
    VA_nabor2:=false;        (*закрити КЛ набору 2*)
    VA_sliv:=true;          (*відкрити КЛ зливу*)
end_if;
if not LS_nyz and VA_sliv then (*якщо відкль датчик ниж. рівня і йде злив*)
    VA_sliv:=false;         (*закрити КЛ зливу*)
    (*якщо СТОП не нажата - відкрити КЛ набору 1*)
    if not SB_stop then VA_nabor1:=true;end_if;
end_if;

```

Рис.2.3. Секція логіки управління рівня ST-мовою

Перевірка виконання роботи та запитання до захисту.

Викладачем перевіряється виконання поставленого завдання. Студент повинен пояснити програму та призначення кожного пункту виконаного завдання.

1. В яких режимах по відношенню до ПЛК може знаходитись UNITYPRO?

2. Чим відрізняється повна компіляція проекту від часткової? В якому випадку можлива зміна проекту в ПЛК не зупиняючи його роботу?

3. Які задачі доступні в M340? Розкажіть про принципи функціонування цих задач та їх настройки: виклик задач, сторожовий таймер. Покажіть де в проекті створюються та конфігуруються параметри задач.

4. Яким чином в UNITY PRO створюються секції? В якій послідовності вони виконуються в межах однієї задачі? Чи може MAST задача складатися з секцій написаних на всіх 5-ти мовах LD, ST, IL, FBD та SFC?

5. Поясніть відмінність локалізованих від нелокалізованих даних в ПЛК з UNITY.

6. Які зони локалізованої пам'яті даних Ви можете назвати?

7. Де в M340 зберігається програма користувача та дані?

8. Яким чином в UNITY PRO створюються змінні? Які типи змінних Ви знаєте?

9. Яким чином в UNITY PRO створюються локалізовані змінні? Чим локалізовані змінні відрізняються від нелокалізованих?

10. Які особливості Ви можете назвати при програмуванні секцій мовами LD та ST в середовищі UNITY PRO?

ДОДАТОК 2. Розробка та відладка програм в UNITY PRO.

Д2.1. Режими роботи. UNITY PRO по відношенню до ПЛК може знаходитись в 2-х режимах:

offline: в цьому режимі відсутній зв'язок середовища UNITY PRO з ПЛК, всі зміни в проекті відображаються тільки в файлі проекту на ПК;

online: в цьому режимі відбувається обмін між UNITY PRO та ПЛК.

Перехід в режим on-line відбувається через команду в меню PLC->Соппест. При цьому, UNITY PRO з'єднається з тим ПЛК, адреса якого прописана у вікні налаштування зв'язку UNITY PRO. Вікно налаштування адрес ПЛК та імітатора ПЛК викликається в меню PLC->Set Address...

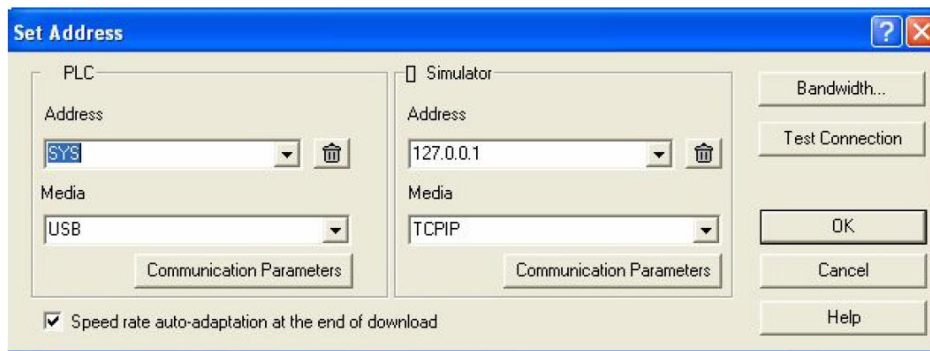


Рис.Д2.1. Вікно налаштування адреси ПЛК та імітатора

При необхідності роботи з імітатором ПЛК, перед з'єднанням необхідно вибрати пункт PLC->Simulation Mode, а для роботи з фізичним ПЛК – PLC->Standard Mode. Якщо з'єднання відбулося, для можливості зміни програми в ПЛК (режим роботи Programming Mode) необхідно щоб проекти в ПЛК і в середовищі UNITYPRO були ідентичними. Ідентичність проекту сигналізується в панелі статусу зеленим індикатором **EQUAL**, а якщо проекти відрізняються – червоним **DIFFERENT** (рис.Д2.2). Для переходу з режиму DIFFERENT в EQUAL необхідно зробити одну з двох операцій: завантажити проект з UNITY PRO в ПЛК (PLC->Transfer Project to PLC), або вивантажити проект з ПЛК в середовище UNITY PRO (PLC->Transfer Proect from PLC).

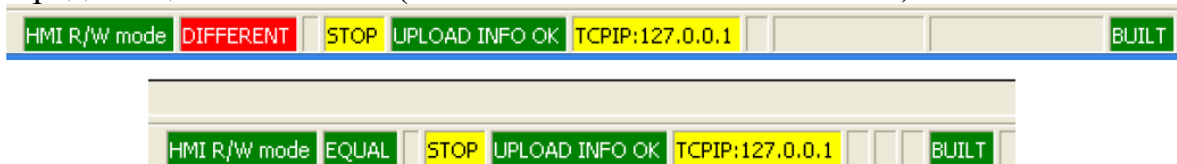


Рис.Д2.2. Вигляд панелі статусу в режимі DIFFERENT (зверху) та EQUAL (знизу).

Завантаження проекту в ПЛК можливе тільки у випадку, якщо він копільований і конфігурація апаратного забезпечення процесорного модуля відповідає дійсній. Можливі два варіанти компіляції:

- Build->**Build Changes**: ця команда приводить до компіляції тільки змінених частин проекту;
- Build->**Rebuild All Project**: ця команда приводить до повної перекомпіляції проекту.

Завантаження нового проекту в ПЛК приводить до його зупинки та ініціалізації. По цій причині зміни в проекті намагаються проводити в режимі EQUAL, шляхом часткової компіляції. Часткова компіляція Build Changes дає можливість в режимі on-line EQUAL змінювати програму в ПЛК не зупиняючи його. Для цього, не виходячи з режиму on-line, користувач робить всі зміни в середовищі UNITY PRO, після чого викликає

команду Build Changes, що приводить до часткової компіляції і зміни проекту в ПЛК.

Результат компіляції висвічується у вікні "Output Window", яке знаходиться в нижній частині екрану. Якщо при компіляції є помилки, в даному вікні з'явиться їх перелік. Зробивши подвійний клік по повідомленню помилки, можна перейти на її джерело. Перевірити синтаксичні помилки в проекті можна також командою Build->Analyze.

Для управління роботою ПЛК використовуються команди: PLC->RUN (запуск на виконання), PLC->STOP (зупинка), PLC->Init (ініціалізація ПЛК).

Д2.2. Конфігурування задач та створення секцій. Режими, періодичність та сторожові таймери задач ПЛК конфігуруються через їх контекстне меню в пункті Properties. Секції програми користувача створюються через команду контекстного меню Sections->New Section. Підрозділ Sections вибирається у відповідному розділі задач MAST, FAST або EVENTS. Для секції вказується ім'я, мова програмування а також, при необхідності, додаткові властивості захисту (Protection) та змінна яка буде управляти виконанням секції (Condition).

Д2.3. Робота з редакторами LD та ST. В загальному, мова LD в UNITY PRO дуже схожа на мову LD в PL7 PRO. Всі доступні компоненти знаходяться на панелі інструментів, яка показана на рис. Д2.3.

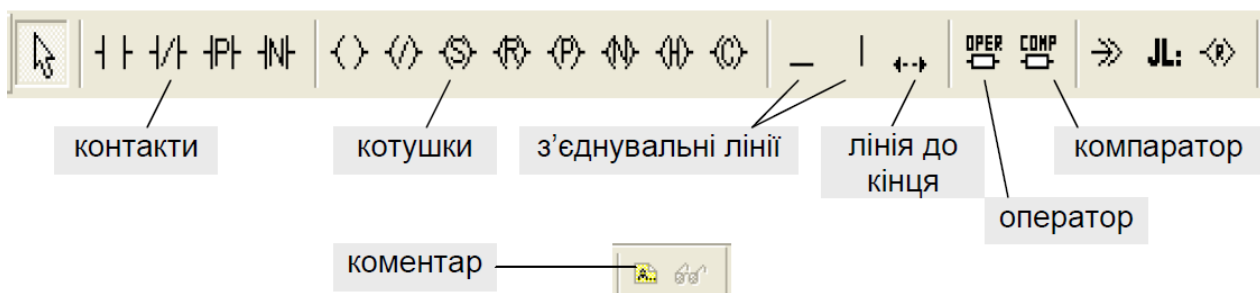


Рис.Д2.3. Палітра інструментів LD.

Крім стандартних елементів в LD можна використовувати будь яку функцію або функціональний блок з бібліотеки UNITY PRO. Серед особливостей LD в UNITY PRO порівняно з PL7 можна виділити (рис.Д2.4):

- більша кількість комірок;
- можливість послідовного з'єднання котушок;
- можливість розміщення контактів з права від котушок;
- можливість використання в аргументах контактів ST вирази;
- змішувати LD з мовою FBD в одній секції;

Мова ST в UNITY практично не відрізняється від реалізації в PL7. Серед особливостей можна назвати ряд інструкцій ST в PL7, які в UNITY доступні як функції (SET(), RESET(), RE(), FE()). Тому запис в змінну або по адресі логічної "1" в ST для UNITY буде виглядати так: %M20:=TRUE;

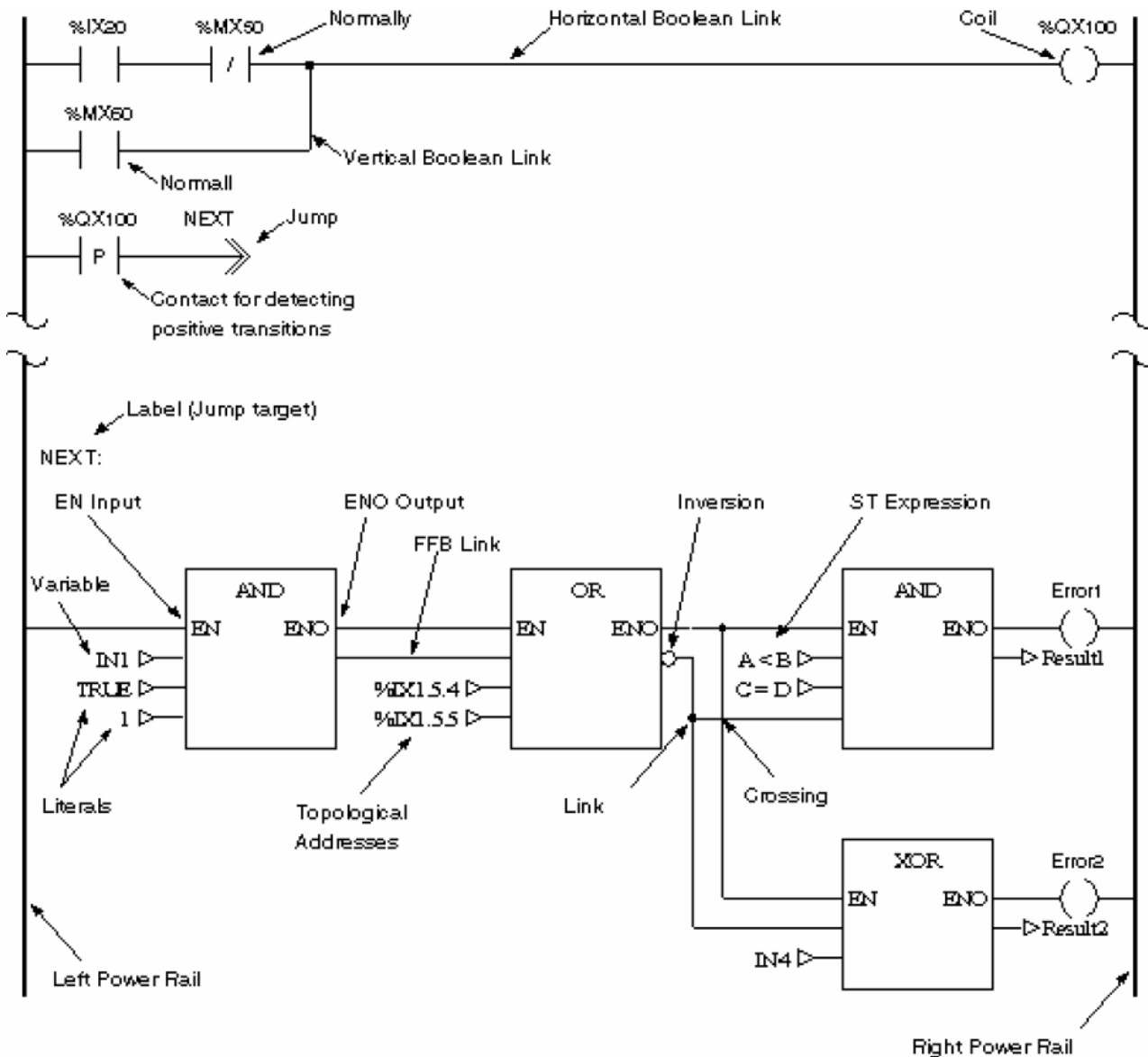


Рис.Д2.4. Загальне представлення LD.

Д2.4. Створення та редагування змінних.

Для створення та редагування змінних, екземплярів функціональних блоків, нових типів змінних та функціональних блоків в UNITY PRO використовується *редактор даних {Data Editor}*. Редактор даних викликається з Project Browser -> Variables @ FB Instances.

Д2.5. Відладка програм.

Для перевірки роботи програми в середовищі UNITY PRO доступні засоби відображення та вводу даних, а також засоби відладки (Debug). Засоби відображення та вводу даних можна умовно поділити на:

- анімаційні засоби програмних секцій *{ProgramAnimation}*,
- *анімаційні таблиці {AnimationTables}*,
- *операторські екрани {OperatorScreen}*,

Компільовані секції програм в режимі on-line по замовченню відображаються в анімаційному вигляді. Включення та виключення анімації

проводиться через команди головного меню Services->Animation.

В анімаційному режимі всі дискретні змінні підсвічуються зеленим в стані лог. "1" та червоним в стані лог. "0". Всі інші типи змінних підсвічуються жовтим кольором. В графічних мовах (LD, FBD, SFC) контакти та з'єднувальні лінії, по яким "проходить сигнал" підсвічуються зеленим, а по яким "не проходить" — червоним. Таким чином, наприклад в LD можна визначити де "обірвався" логічний ланцюжок. В секціях графічних мов програмування, показуються числові значення аргументів функцій та функціональних блоків. В текстових мовах (ST, IL) числові значення змінних підсвічуються при наведені мишкою. В анімаційному режимі, в контекстному меню блоків доступні додаткові команди, зокрема на зміну значення змінних.

Анімаційні таблиці в UNITY PRO реалізовані аналогічно як в PL7 PRO. Їх можна створити в контекстному меню розділу Animation Tables в Project Browser, або в контекстному меню виділеної частини програми через команду Initialize Animation Tables. Для можливості зміни змінних в анімаційній таблиці, необхідно активувати режим Modification (рис.Д2.5).

| Name | Value | Type | Comment |
|-----------|-------|------|-------------------------------|
| VA_nabor1 | 0 | BOOL | Клапан набору 1-го продукта |
| LE_Level | 100.1 | REAL | Імітований рівень |
| VA_nabor2 | 0 | BOOL | Клапан набору 2-го продукта |
| VA_sliv | 0 | BOOL | Клапан зливу |
| LS_nyz | 1 | BOOL | Сигналізатор нижнього рівня |
| LS_ser | 1 | BOOL | Сигналізатор середнього рівня |
| LS_verh | 1 | BOOL | Сигналізатор верхнього рівня |

Рис.Д2.5. Анімаційна таблиця.

Операторські екрани в UNITY PRO реалізовані аналогічно як Runtime Screens в PL7 PRO. Створюються екрани в розділі проекту Operator Screens. Для можливості зміни значень в операторських екранах необхідно викликати команду Services ->Enable Variable Modification.

ЛАБОРАТОРНА РОБОТА № 3

Регулювання технологічного параметра за заданим алгоритмом

Тривалість: 2 акад. години .

Мета: ознайомлення з принципами програмування системи автоматизації FBD-мовою та використання функцій і функціональних блоків стандартної бібліотеки типів.

Лабораторна установка.

Програмне забезпечення. UNITY PRO V \geq 4.0.

Загальна постановка задачі. Необхідно створити змінні та програму користувача для ПЛК M340 відповідно до наступної задачі (рис.3.1). Після натискання кнопки ПУСК відкривається клапан набору першого продукту. Після

досягнення середнього рівня клапан 1-го продукту закривається, відкривається клапан набору 2-го продукту. Після спрацювання сигналізатору верхнього рівня закривається клапан набору 2-го продукту, відкривається клапан пари на 100% (діапазон виходу 0-100%). Після досягнення температури 95°C (діапазон датчика 0-150°C), клапан пари залишається відкритим на 20% ще протягом 10 с. Після закінчення витримки, рідина зливається з

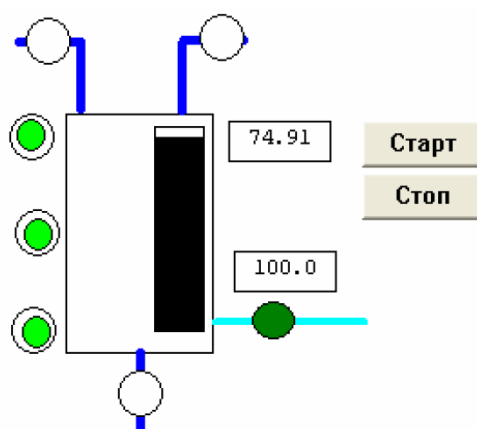


Рис.3.1. Приклад операторського екрану до поставленої задачі.

апарату. Після відключення сигналізатору нижнього рівня, цикл повторюється у випадку якщо кнопка СТОП не натиснута. Якщо СТОП натиснута — клапан зливу закривається. В ПЛК поступає сигнал від датчика рівня з діапазоном вимірювання 0-5 м.

Написати секцію з програмою імітатору роботи датчика та сигналізаторів рівня і температури. В окремих секціях написати програму масштабування входів та виходу. Логіку виконання програми теж реалізувати в окремій секції. Програму перевірити та відлагодити з використанням таблиць анімацій або операторських екранів. Програми в усіх секціях створити FBD-мовою.

Послідовність виконання роботи

Перед виконанням лабораторної роботи ознайомтесь з додатком 3.

1) Запустити на виконання UNITY PRO. Створити новий проект з ПЛК M340. В конфігурації замовити модулі відповідно дорис.3.2.

2) В редакторі даних проекту створити змінні, відповідно до задачі, як показано рис.3.3 (коментарі для змінних можна не вказувати).

3) В редакторі даних проекту створити екземпляри функціональних блоків, як показано на рис.3.4 (коментарі для екземплярів можна не вказувати).

4) Створити секцію імітації рівнів та температури (див. рис.3.5 тарис.3.6).

5) За допомогою анімаційної таблиці або операторського екрану перевірити роботу програми імітації сигналів.

6) Створити всі інші секції (рис. 3.7, рис. 3.8, рис. 3.9). Секції в проекті повинні розміщуватись відповідно до рис.3.5. Вставку екземплярів функціональних блоків проводити через команду Data Selection. Коментарі в секціях можна не вказувати.

7) Перевірити роботу програми за допомогою анімаційної таблиці або операторського екрану. При необхідності – відлагодити її.

Проект зберегти на диску (буде використаний в лаб.роб.№4).

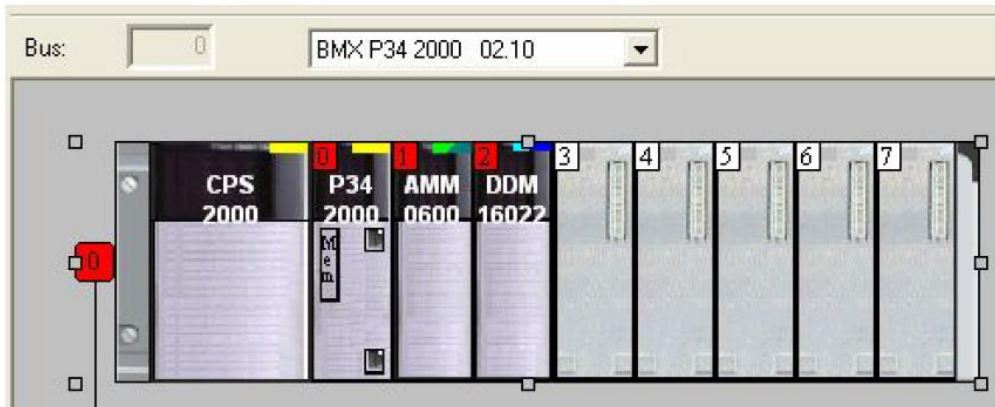


Рис.3.2 Конфігурація ПЛК М340 до поставленої задачі.

| Name | Type | Addr... | Value | Comment |
|--------------|--------------|----------|---------|---|
| VR_Par_R | REAL | | | Відмасштабоване значення ВМ 0-100% |
| TE_Tank_R | REAL | | | Відмасштабована Температура 0-150 гр.С |
| LE_Tank_R | REAL | | | Відмасштабований Рівень 0-5 м |
| SCALE_PARA_T | Para_SCALING | | | |
| in_min | REAL | | 0.0 | мінімальне значення вхідного сигналу |
| in_max | REAL | | 10000.0 | максимальне значення вхідного сигналу |
| out_min | REAL | | 0.0 | мінімальне значення вихідного сигналу 0 град С |
| out_max | REAL | | 150.0 | максимальне значення вихідного сигналу 150 град С |
| clip | BOOL | | 1 | активувати обмеження виходу |
| SCALE_PARA_L | Para_SCALING | | | |
| in_min | REAL | | 0.0 | мінімальне значення вхідного сигналу |
| in_max | REAL | | 10000.0 | максимальне значення вхідного сигналу |
| out_min | REAL | | 0.0 | мінімальне значення вихідного сигналу 0 м |
| out_max | REAL | | 5.0 | максимальне значення вихідного сигналу 5 м |
| clip | BOOL | | 1 | активувати обмеження виходу |
| VR_par | INT | %QW0.1.4 | | Клапан пари |
| TE_Tank | INT | %IW0.1.1 | | Датчик температури |
| StepProg | INT | | | |
| LE_Tank | INT | %IW0.1.0 | | Датчик рівня |
| VA_sliv | EBOOL | %QQ.2.18 | | Клапан зливу |
| VA_nabor2 | EBOOL | %QQ.2.17 | | Клапан набору 2-го продукта |
| VA_nabor1 | EBOOL | %QQ.2.16 | | Клапан набору 1-го продукта |
| LS_verh | EBOOL | %IO.2.2 | | Сигналізатор верхнього рівня |
| LS_ser | EBOOL | %IO.2.1 | | Сигналізатор середнього рівня |
| LS_nyz | EBOOL | %IO.2.0 | | Сигналізатор нижнього рівня |
| SB_stop | BOOL | | | Кнопка зупинки процесу |
| SB_Start | BOOL | | | Кнопка запуску процесу |

Рис.3.3. Перелік змінних до проекту.

| Name | r | Type | v | Comment |
|-----------|---|---------|---|---|
| RS_Nabor1 | | RS | | Тригер для управління VA_Nabor1 |
| RS_Nabor2 | | RS | | Тригер для управління VA_Nabor2 |
| RS_Sliv | | RS | | Тригер для управління VA_Sliv |
| SCALE_L | | SCALING | | екземпляр функц блока масштабування для LE_Tank |
| SCALE_T | | SCALING | | екземпляр функц блока масштабування для TE_Tank |
| T_DELAY | | TON | | екземпляр таймера на витримку |

Рис.3.4. Перелік екземплярів елементарних функціональних блоків.

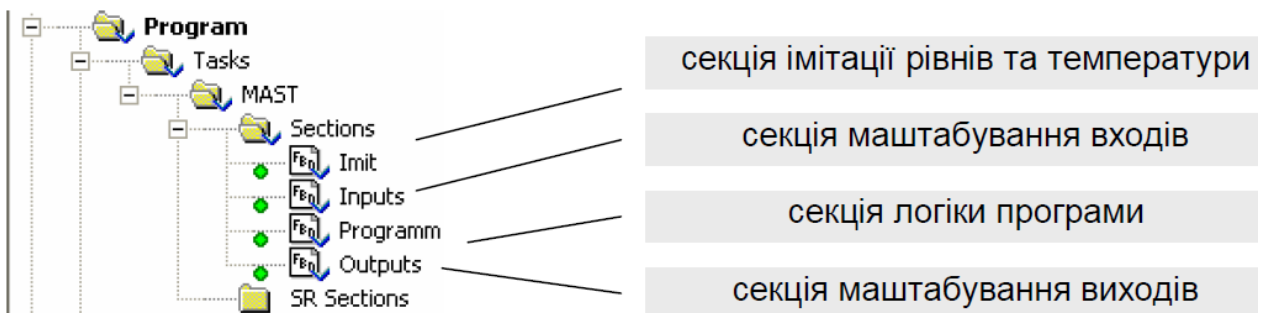


Рис.3.5. Структура програми проекту

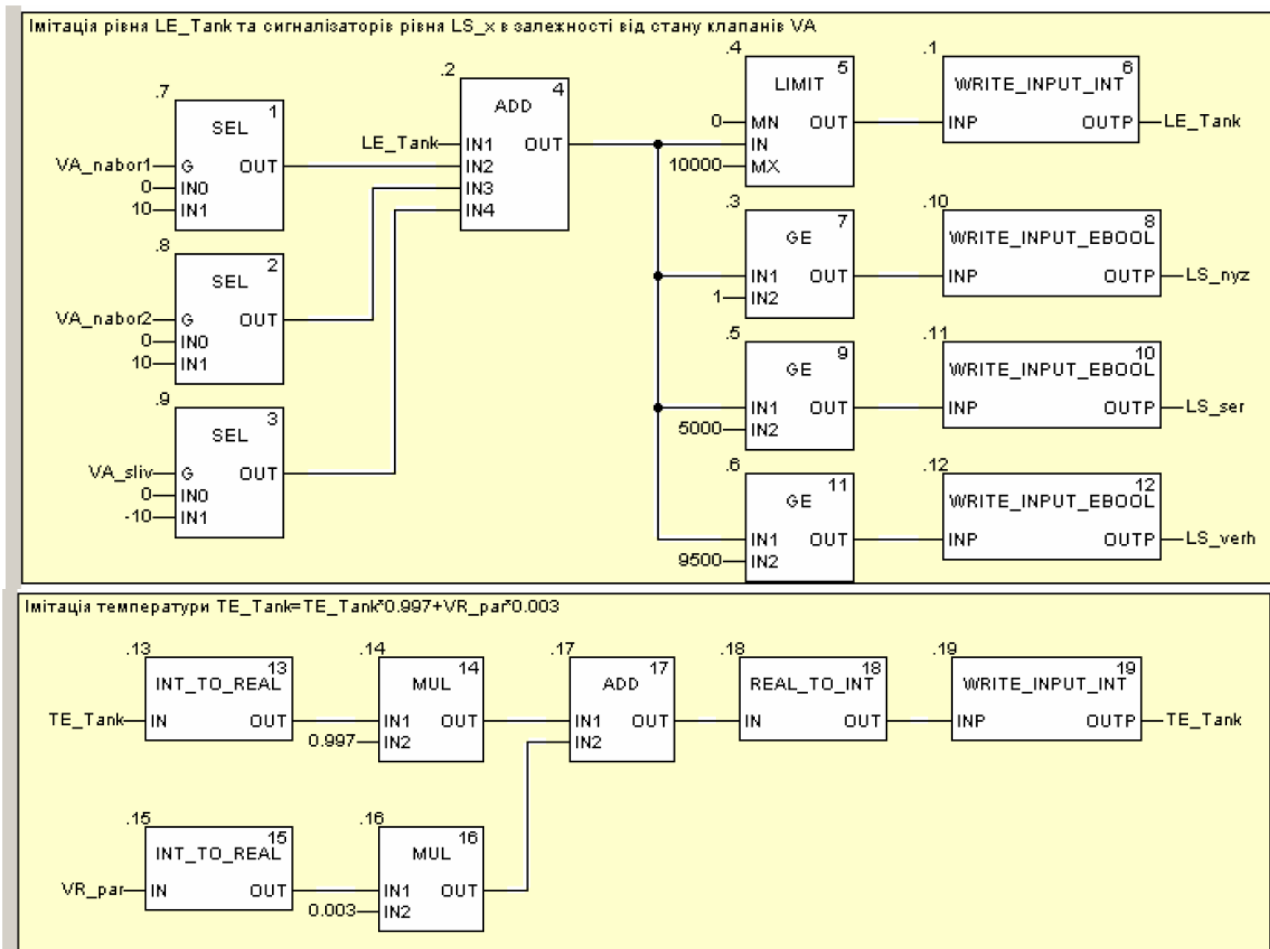


Рис.3.6. Секція імітації датчиків та сигналізаторів рівня і температури

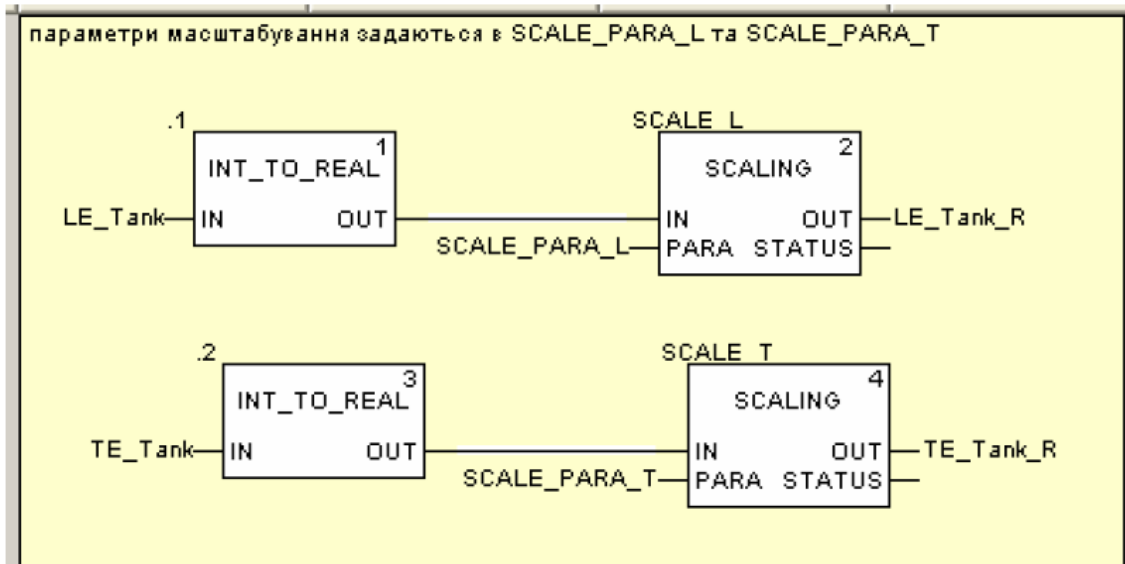


Рис.3.7. Секція Inputs.

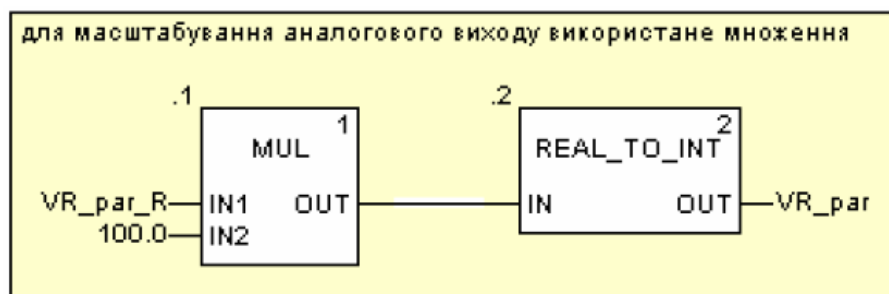


Рис.3.8. Секція Outputs.

Перевірка виконання роботи та питання до захисту.

Викладачем перевіряється виконання поставленого завдання. Студент повинен пояснити програму та призначення кожного пункту виконаного завдання.

1. Поясніть в якій послідовності виконуються FFB в одній секції програми. Де це видно і чим це визначається? Яким чином можна змінити цю послідовність?
2. Які правила використання елементів для зв'язку "Link" Ви можете назвати? Яке призначення інверсії?
3. Розкажіть про призначення параметрів EN/ENO. Яким чином вони активуються для FFB в секції FBD? Де в лабораторній роботі використані EN/ENO, прокоментуйте роботу цих блоків.
4. Поясніть чим відрізняються функції від функціональних блоків? Навіщо створювати екземпляри функціональних блоків, і яким чином це зробити в UNITY PRO?
5. Прокоментуйте роботу бібліотечних елементів FFB, які використані в програмі.
6. Яким чином можна змінювати значення вхідних сигналів в імітаторі ПЛК?
7. Поясніть роботу секції імітатора об'єкта.
8. Поясніть роботу секцій масштабування.
9. Поясніть роботу секції основної програми.

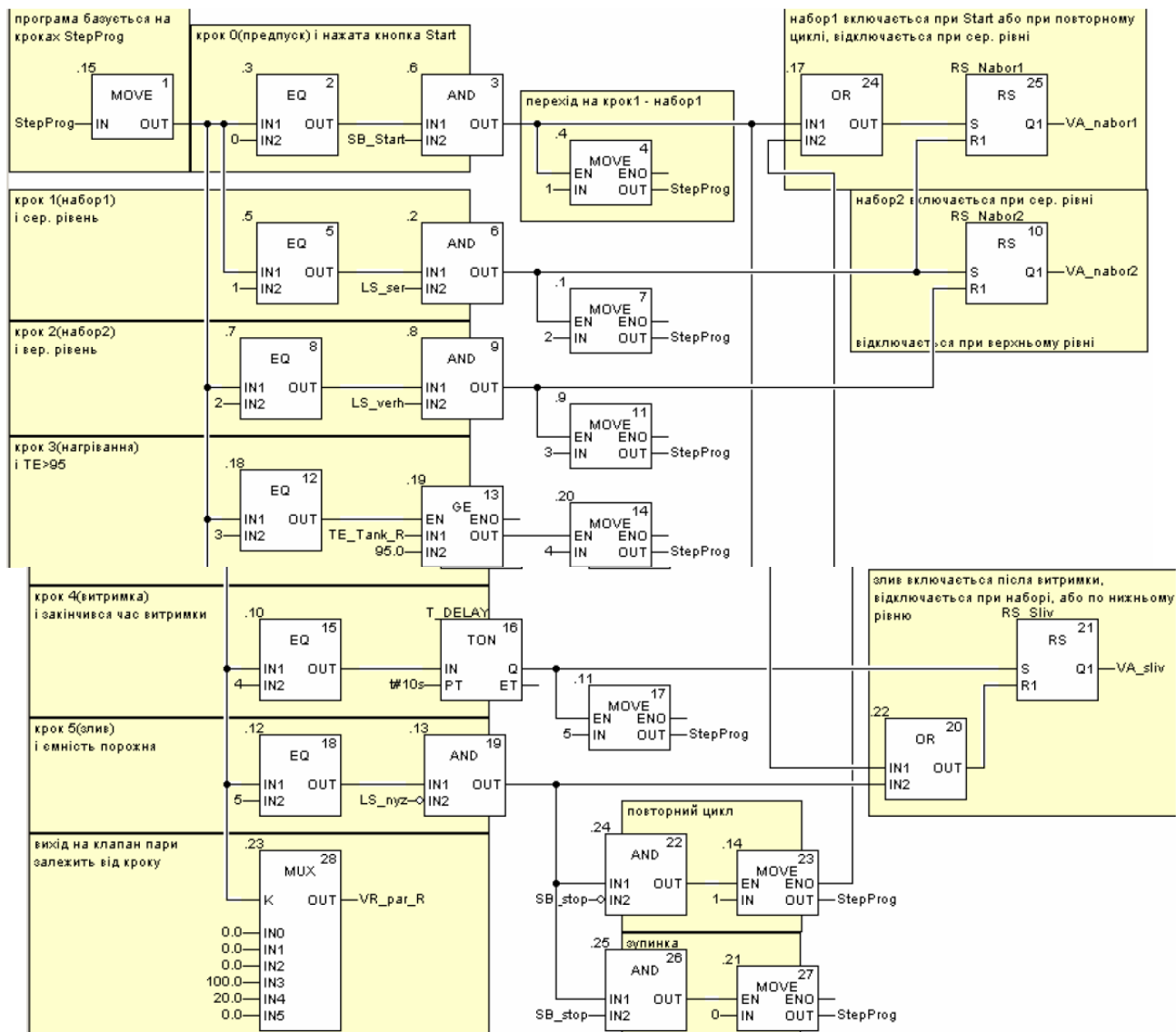


Рис.3.9. Секція Program.

ДОДАТОК 3. Створення програм FBD-мовою. Створення та використання функціональних блоків.

Д3.1. Редактор FBD. Вставка і з'єднання функцій та функціональних блоків (FFB) в UNITY PRO проводиться через панель палітри інструментів (рис.Д3.1).

Вставка функцій проводиться через виклик помічника вставки нового FFB (FFB Input Assistant), шляхом вибору елемента з бібліотеки або ручного вводу назви.

Кількість входів для деяких функцій (наприклад ADD) змінюється шляхом їх "розтягування" мишею, "заціпивши" мишкою за нижній край функції в редакторі FBD.

Вставка екземплярів функціональних блоків може проводитись через вибір існуючого по команді Data Selection, або шляхом створення нового FFB Input Assistant. Ці операції можна також проводити за допомогою провідника бібліотеки типів.

З'єднання FFB проводиться з використанням команди Link, при цьому з'єднуються входи з виходами. **При необхідності відгалуження**, починати

його необхідно з виходу FFB. *Інверсію* можливо проводити як для входів так і для виходів FFB. З'єднання можна показувати через *розірване посилання*, для чого в контекстному меню з'єднання вибирається команда "Show As Connector".

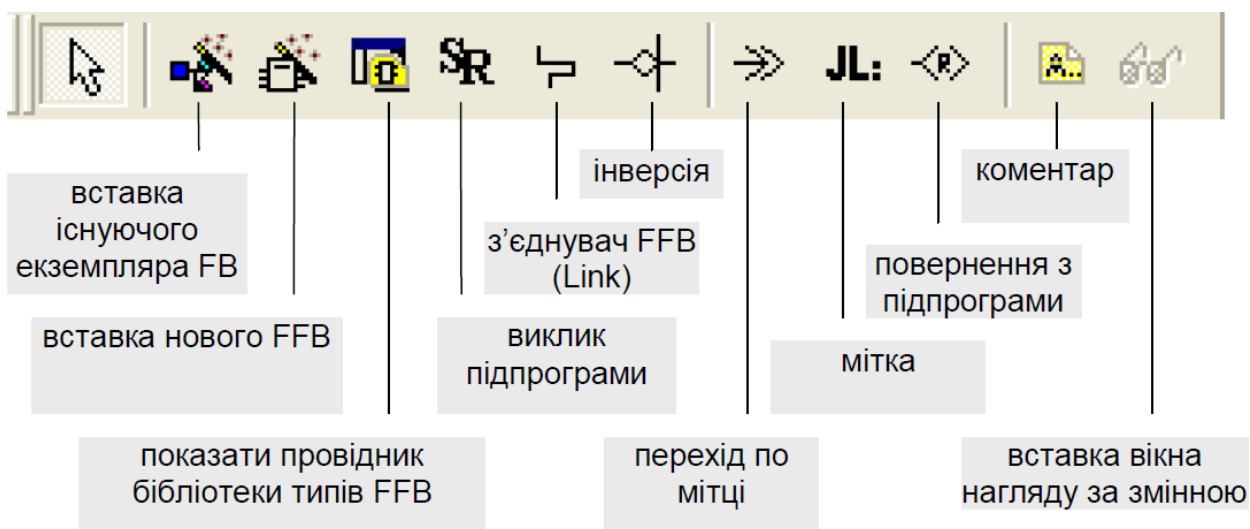


Рис.Д3.1. Палітра інструментів FBD.

Коментар розміщується на задньому фоні елементів FFB. При необхідності багаторядкового вводу, в кінці рядку нажимають CTRL+ENTER.

Для *активації/деактивації управляючих входів EN* та *контролюючого ENO*, через контекстне меню FFB викликають вікно властивостей (Properties), в якому виставляють опцію "Show EN/ENO".

Д3.2. Імітація вхідних каналів в імітаторі ПЛК. Для зміни в імітаторі ПЛК значень дискретних та аналогових вхідних сигналів (%I, %IW) програмним шляхом використовуються функції відповідно WRITE_INPUT_EBOOL та WRITE_INPUT_INT. На входи цих функцій подаються необхідні значення, а на виходах вказуються адреси %I (для WRITE_INPUT_EBOOL) та %IW (для WRITE_INPUT_INT), або локалізовані змінні, які до цих адрес прив'язані.

Д3.3. Створення екземплярів функціональних блоків. Оскільки функціональні блоки потребують виділення області пам'яті, перед їх використанням створюють екземпляри FB. Створення екземплярів можливе двома шляхами:

- через Data Editor (в провіднику проекту підрозділ Elementary FB Instances);

- "на льоту", при вставці його в секцію програми з бібліотеки типів.

В будь-якому випадку всі існуючі функціональні блоки (навіть якщо вони не використовуються ніде в програмі), доступні в розділі Variables & FB Instances.

ЛАБОРАТОРНА РОБОТА № 4

Програмування системи автоматизації з використанням блоків бібліотеки управління UNITY PRO

Тривалість: 2 акад. години.

Мета: ознайомлення з елементами бібліотеки управління Control Library, реалізація алгоритмів регулювання на базі ПІ-регулятора.

Лабораторна установка.

Програмне забезпечення. UNITY PRO V \geq 4.0.

Загальна постановка задачі. Необхідно створити змінні та програму користувача для ПЛК М340 відповідно до наступної задачі (рис.4.1). Робота програми описується алгоритмом наведеним в лабораторній роботі №4 за винятком етапу витримки. Витримка повинна тривати 3 хвилини, в цей час регулятор повинен підтримувати температуру на заданому рівні. Задане значення температури визначає оператор. Програму перевірити та відлагодити з використанням операторських екранів.

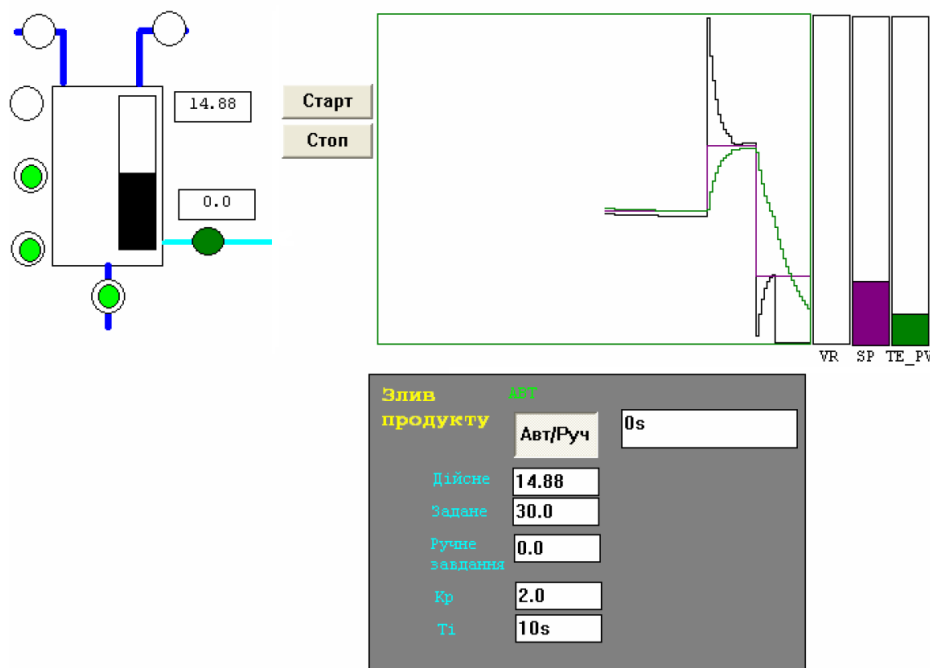


Рис.4.1. Приклад операторського екрану до поставленої задачі.

Послідовність виконання роботи.

- 1) Запустити на виконання UNITY PRO. Завантажити попередньо збережений проект для лабораторної роботи №3.
- 2) Додати змінні TIC1_Man_Auto, TIC1_PARA, TIC1_SP, VR_par_MAN до існуючих змінних в проекті (рис.4.3).
- 3) Додати екземпляри функціональних блоків TIC1, SMPL1 до існуючих екземплярів в проекті (рис.4.4).
- 4) Модифікувати секцію Programm: змінити час в блоці T_DELAY на 3 хвилини; доповнити програму починаючи з блоку ".23" (MUX) блоками, які вказані на рис.4.4.

| Name | Type | Addr... | Value | Comment |
|---------------|--------------|----------|-------|--|
| LE_Tank | INT | %IW0.1.0 | | Датчик рівня |
| LE_Tank_R | REAL | | | Відмасштабований Рівень 0-5 м |
| LS_nyz | EBOOL | %IO.2.0 | | Сигналізатор нижнього рівня |
| LS_ser | EBOOL | %IO.2.1 | | Сигналізатор середнього рівня |
| LS_verh | EBOOL | %IO.2.2 | | Сигналізатор верхнього рівня |
| SB_Start | BOOL | | | Кнопка запуску процесу |
| SB_stop | BOOL | | | Кнопка зупинки процесу |
| SCALE_PARA_L | Para_SCALING | | | |
| SCALE_PARA_T | Para_SCALING | | | |
| StepProg | INT | | 0 | Крок програми |
| TE_Tank | INT | %IW0.1.1 | | Датчик температури |
| TE_Tank_R | REAL | | | Відмасштабована Температура 0-150 гр.С |
| TIC1_Man_Auto | EBOOL | | 1 | Режим руч/авт для регулятора температури |
| TIC1_PARA | Para_PI_B | | | |
| id | UINT | | | |
| pv_inf | REAL | | 0.0 | обмеження по мінімуму вхідної величини |
| pv_sup | REAL | | 150.0 | обмеження по максимуму вхідної величини |
| out_inf | REAL | | 0.0 | обмеження по мінімуму вихідної величини |
| out_sup | REAL | | 100.0 | обмеження по максимуму вихідної величини |
| rev_dir | BOOL | | false | режим прямої роботи ПІ-регулятора |
| en_rcpy | BOOL | | false | не використовувати RCPY |
| kp | REAL | | 1.0 | коефіцієнт пропорційності |
| ti | TIME | | t#0s | час інтегрування |
| dband | REAL | | 0.2 | зона нечутливості |
| outbias | REAL | | 50.0 | зміщення виходу регулятора в П-режимі |
| TIC1_SP | REAL | | 80.0 | Задане значення на регулятор температури |
| VA_nabor1 | EBOOL | %Q0.2.16 | | Клапан набору 1-го продукту |
| VA_nabor2 | EBOOL | %Q0.2.17 | | Клапан набору 2-го продукту |
| VA_sliv | EBOOL | %Q0.2.18 | | Клапан зливу |
| VR_par | INT | %QW0.1.4 | | Клапан пари |
| VR_par_MAN | REAL | | | Вихід на ВМ в ручному режимі |
| VR_Par_R | REAL | | | Відмасштабоване значення ВМ 0-100% |

Рис.4.2. Перелік змінних.

| Name | r | Type | v | Comment |
|-----------|---|----------|---|---|
| RS_Nabor1 | | RS | | Тригер для управління VA_Nabor1 |
| RS_Nabor2 | | RS | | Тригер для управління VA_Nabor2 |
| RS_Sliv | | RS | | Тригер для управління VA_Sliv |
| SCALE_L | | SCALING | | екземпляр функц блока масштабування для LE_Tank |
| SCALE_T | | SCALING | | екземпляр функц блока масштабування для TE_Tank |
| SMPL1 | | SAMPLETM | | |
| T_DELAY | | TON | | екземпляр таймера на витримку |
| TIC1 | | PI_B | | екземпляр функц болка ПІ-регулятора |

Рис.4.3. Перелік екземплярів функціональних блоків.

5) Перейти в режим off-line, імпортувати операторський екран в проект: в Project Browser контекстне меню Operator Screens->Import->Робочий стіл\тека Лаб_ППР\PID.XCR.

6) Скопіювати проект, завантажити в імітатор контролера, запустити на виконання. На операторському екрані натиснути кнопку "Старт" для запуску програми на виконання.

7) На кроці "витримка" переключити контур в ручний режим і намагатися втримати завдання 85 °С в ручному режимі протягом 3 хв. витримки. Після закінчення часу витримки переключити контур в режим "АВТ".

8) На наступному циклі роботи, на кроці "витримка" в автоматичному режимі виставити завдання (уставку)=75 °С, $K_p=1.0$, $T_i=0s$ (П-режим). Занотувати в чернетці значення яке видає ВМ, значення завдання (уставки), плинного значення та розузгодження.

9) На наступному циклі роботи, на кроці "витримка" в автоматичному режимі виставити завдання (уставку)=75 °С, $K_p=1.0$, $T_i=30s$ (ПІ-режим). Змінюючи уставку з ряду (20,50,80,100) а також K_p , T_i , дочекавшись закінчення перехідного процесу досягнути найкращих результатів по динамічним характеристикам. Занотувати в чернетці значення K_p та T_i .

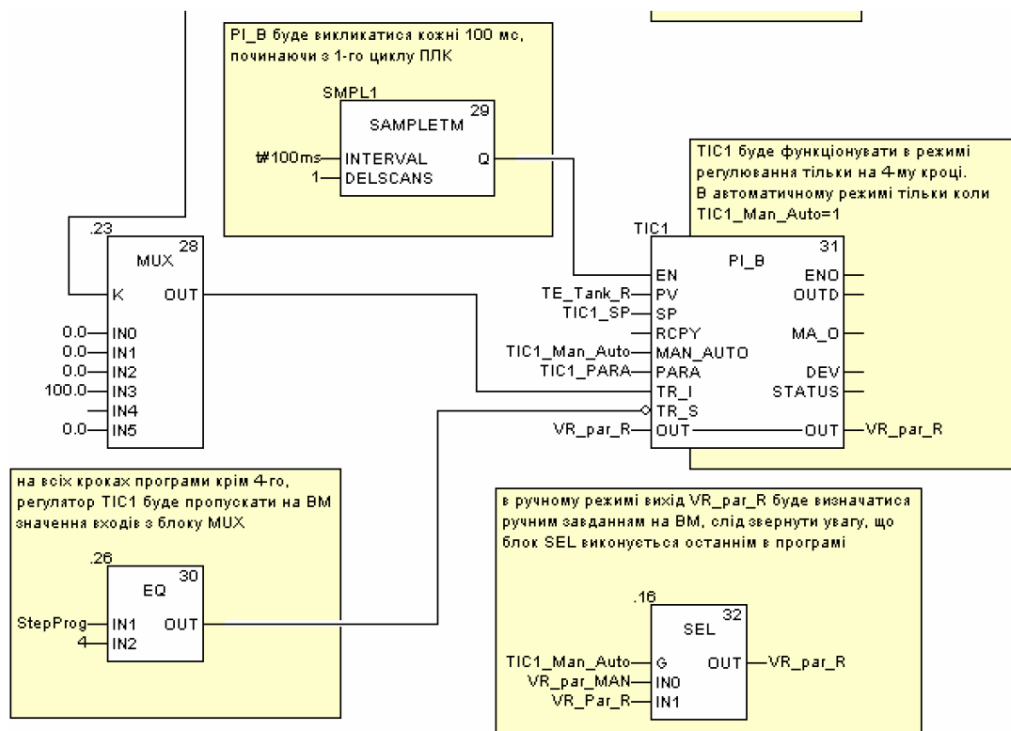


Рис.4.4. Модифікована частина секції програми з рис.3.9.

Перевірка виконання роботи та питання до захисту.

Викладачем перевіряється виконання поставленого завдання. Студент повинен пояснити програму та призначення кожного пункту виконаного завдання.

1. Перерахуйте які Ви знаєте бібліотечні елементи UNITY, які реалізують стандартні закони регулювання?

2. Прокоментуйте призначення блоків сімейства Controller бібліотеки UNITY.

3. Розкажіть про призначення та роботу функціонального блока SAMPLETM.

4. Прокоментуйте функціональну структуру функціонального блока PI_V.

5. Яким чином вказуються настройки регулятора для функціонального блока PI_V? Прокоментуйте структуру Para_PI_V.

6. Як працює ПІ-регулятор PI_V в ручному режимі? Яким чином задається цей режим?

7. Як працює ПІ-регулятор PI_V в режимі П-регулювання? Прокоментуйте результати п.8. лабораторної роботи.

8. Поясніть призначення параметрів TR_I та TR_S. Поясніть як ці входи використані в програмі користувача для даної лабораторної роботи.

ЛАБОРАТОРНА РОБОТА № 5

Створення проекту автоматизації технологічного процесу SFC-мовою

Тривалість: 4 акад. години.

Мета: придбати досвід створення проекту автоматизації SFC-мовою програмування в середовищі UNITY PRO.

Завдання для виконання роботи

Загальна постановка завдання Необхідно створити проект в UNITY PRO для реалізації поставленої задачі з використанням мови SFC. Відлагодження проекту зробити з використанням готового програмного імітатора об'єкта та операторського екрану.

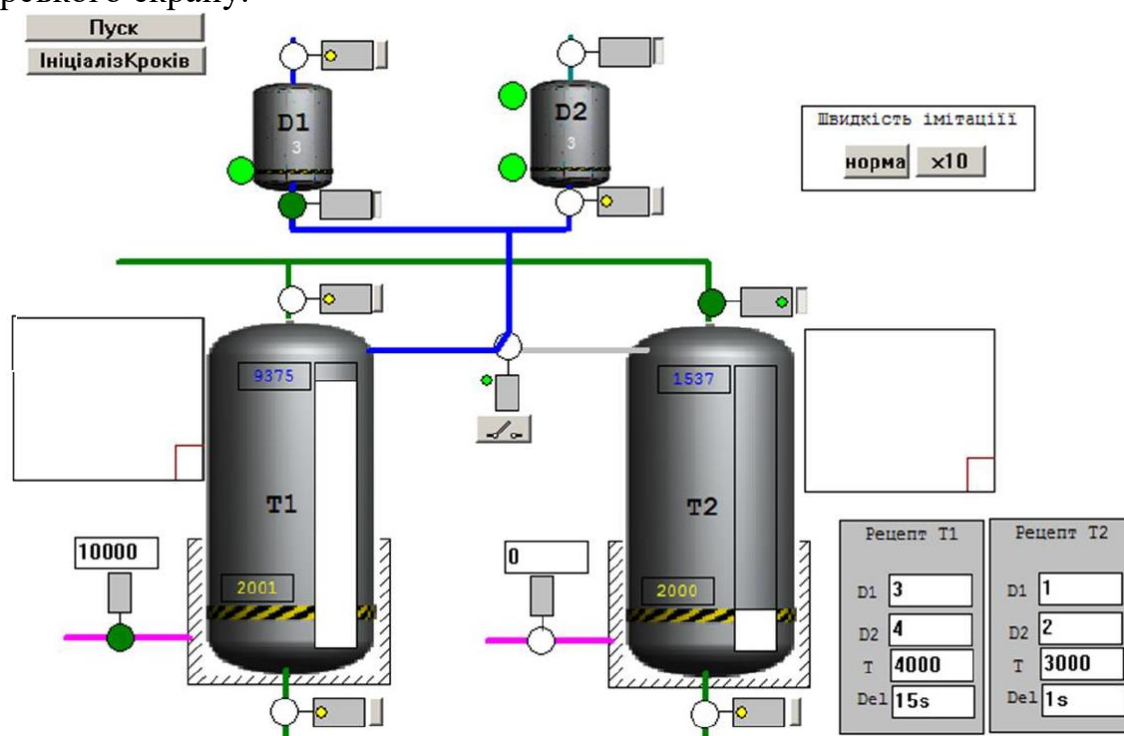


Рис.5.1. Операторський екран для контролю та управління за процесом приготування продукту

Опис об'єкта. Технологічна установка складається з наступних елементів (рис.5.1):

- танки T1 та T2, в яких готуються продукти за різними рецептами; танки об'язані наступними засобами КВПіА:

* запірні клапани набору та зливу, кожний з датчиками кінцевого положення "закритий" та "відкритий";

* регулюючий клапан (0-100%) подачі теплоагента у теплообмінний кожух танку (далі по тексту клапан нагрівання);

* датчик рівня (0-100%) в танку;

* датчик температури в танку (0-100°C);

- дозатори (мірні ємності) D1 та D2, які забезпечують подачу дози компоненту;

дозатори об'язані наступними засобами КВПіА:

* сигналізатор нижнього і верхнього рівнів;

* запірні клапани набору та зливу, кожний з датчиками кінцевого положення "закритий";

- 3-ходовий клапан перемикання трубопроводу подачі з дозаторів на танки T1 та T2; в нормальному стані положення "на T1"; має датчики кінцевого положення "T1" та "T2".

Опис алгоритму задач управління установкою. Управління дозаторами та танками повинно бути розв'язане одне від одного (але координоване), оскільки дозатори можуть бути використані в інших процесах. Дозатори в стані очікування завжди повинні бути наповнені.

Управління процесом приготування повинно відбуватися за таким алгоритмом:

1) У початковому стані (старті ПЛК) клапани набору та зливу танків T1 та T2 повинні бути закритими. Закритість клапанів контролюється кінцевими датчиками положення. Після цього система управління приготуванням продукту переходить в стан готовності.

2) Оператор повинен задати рецепт продукту для приготування в T1 та T2. Рецепт включає наступні поля:

- a. кількість доз компоненту з D1;
- b. кількість доз компоненту з D2;
- c. температуру попереднього нагрівання;
- d. час витримки;

3) Після натискання оператором кнопки "Пуск" відкривається клапан набору танку T1.

4) Після досягнення рівня 50% паралельно з набором включається дозування компонентів D1 та D2 відповідно до рецепту.

5) При досягненні рівня 80%, відкривається клапан набору танку T2.

6) Коли клапан набору T2 повністю відкрився (по датчику положення "відкритий"), клапан набору T1 закривається, і паралельно з приготуванням продукту в T1 йде наповнення і приготування продукту в танку T2.

7) При досягненні рівня 50% в T2 паралельно з набором включається дозування компонентів D1 та D2 відповідно до рецепту. Якщо дозатор в цей час використовується при дозуванні T1, необхідно дочекатися закінчення роботи дозаторів.

8) При досягненні рівня 80%, закривається клапан набору танку T2.

9) Після закриття клапану набору в танку T1 (в наступних пунктах для T2 аналогічно) і закінченні дозування, відкривається повністю клапан подачі теплоагента.

10) Рідина в танках повинна нагрітися до вказаного в рецепті значення, після чого клапан залишається відкритий на 10% протягом вказаного в рецепті часу.

11) Після витримки відкривається клапан зливу і рідина зливається з танку.

12) Через 5с після досягнення рівня менше 1% клапан зливу закривається.

13) Коли обидва танки T1 та T2 порожні, система переходить в початковий стан.

14) У будь який момент часу система повинна мати можливість переходу в початковий стан.

Лабораторна установка

Апаратне забезпечення: ПК.

Програмне забезпечення: UNITY PRO V>=4.0.

Порядок виконання роботи. Заходи безпеки

Необхідно дотримуватись стандартних заходів безпеки при роботі з ПК.

Перед виконанням лабораторної роботи ознайомтесь з додатками 4 та 5.

1) Запустити на виконання UNITY PRO. Створити новий проект з ПЛК М340.

2) Імпортувати файл апаратної конфігурації:

Project Browser -> контекстне меню Configuration -> Import "Робочий стіл\тека Лаб_ППК\lab5_SFC.XHW".

3) Подивитися конфігурацію обладнання для даної задачі.

4) Імпортувати файл з секцією імітатора об'єкту управління:

Project Browser -> Program-> Tasks -> Mast -> контекстне меню Sections -> Import "Робочий стіл\тека Лаб_ППК\LAB5_SFC_simul.XBD";

5) Імпортувати змінні:

Project Browser -> Variables & FB Instances -> Import "Робочий стіл\тека Лаб_ППК\lab5_SFC_vars.XSY"

| | | | | |
|------|---------------|----------|-------|---|
| | InitSFC | | BOOL | ініціалізація програми SFC |
| | Pusk | | BOOL | запуск процесу |
| | VNabor_T2_OPN | %I0.1.1 | EBOOL | клапан набору T2 відкритий |
| | VNabor_T1_CLS | %I0.1.4 | EBOOL | клапан набору T1 закритий |
| | VNabor_T2_CLS | %I0.1.5 | EBOOL | клапан набору T2 закритий |
| | VSliv_T1_CLS | %I0.1.6 | EBOOL | клапан зливу T1 закритий |
| | VSliv_T2_CLS | %I0.1.7 | EBOOL | клапан зливу T2 закритий |
| | VNabor_D1_CLS | %I0.1.8 | EBOOL | клапан набору D1 закритий |
| | VNabor_D2_CLS | %I0.1.9 | EBOOL | клапан набору D2 закритий |
| | VSliv_D1_CLS | %I0.1.10 | EBOOL | клапан зливу D1 закритий |
| | VSliv_D2_CLS | %I0.1.11 | EBOOL | клапан зливу D2 закритий |
| | LSH_D1 | %I0.1.12 | EBOOL | сигналізатор верхнього рівня D1 |
| | LSL_D1 | %I0.1.13 | EBOOL | сигналізатор нижнього рівня D1 |
| | LSH_D2 | %I0.1.14 | EBOOL | сигналізатор верхнього рівня D2 |
| | LSL_D2 | %I0.1.15 | EBOOL | сигналізатор нижнього рівня D2 |
| | VDoz_T1_OPN | %I0.3.0 | EBOOL | З-ходовий клапан в позиції T1 |
| | VDoz_T2_OPN | %I0.3.1 | EBOOL | З-ходовий клапан в позиції T2 |
| | LE_T1 | %IW0.2.0 | INT | рівень T1 (0-10000) |
| | LE_T2 | %IW0.2.1 | INT | рівень T2 (0-10000) |
| | TE_T1 | %IW0.2.2 | INT | температура в T1 (0-10000) |
| | TE_T2 | %IW0.2.3 | INT | температура в T2 (0-10000) |
| | VNabor_T1 | %Q0.1.17 | EBOOL | клапан набору T1 |
| | VSliv_T1 | %Q0.1.18 | EBOOL | клапан зливу T1 |
| | VNabor_T2 | %Q0.1.19 | EBOOL | клапан набору T2 |
| | VSliv_T2 | %Q0.1.20 | EBOOL | клапан зливу T2 |
| | VNabor_D1 | %Q0.1.21 | EBOOL | клапан набору D1 |
| | VNabor_D2 | %Q0.1.22 | EBOOL | клапан набору D2 |
| | VSliv_D1 | %Q0.1.23 | EBOOL | клапан зливу D1 |
| | VSliv_D2 | %Q0.1.24 | EBOOL | клапан зливу D2 |
| | VDoz_T1toT2 | %Q0.1.25 | EBOOL | З-ходовий клапан (0 - на T1, 1 - на T2) |
| | VNagrev_T1 | %Qw0.2.4 | INT | клапан нагрівання T1 (0-10000) |
| | VNagrev_T2 | %Qw0.2.5 | INT | клапан нагрівання T2 (0-10000) |

Рис.5.2. Змінні проекту

6) Після імпорту змінних, ознайомтесь з їх призначенням (рис.5.2 - 5.3). Змінні типу Recipe призначені для збереження рецептів для продуктів в танках T1 та T2. Змінні типу D1 та D2 використовуються для управління дозаторами.

7) Імпортувати операторський екран:

Project Browser -> контекстне меню Operator Screens-> Import-> Робочий стіл\тека Лаб_ППК\lab5_SFC_Screen.XCR.

8) Після імпорту скомпілюйте проект, завантажте його в симулятор ПЛК і запустіть на виконання. Відкрийте операторський екран і поуправляйте процесом вручну (опосередковано), шляхом відкриття і закриття клапанів за допомогою кнопок біля них (рис.5.1): завантажте і вивантажте дозатори; відкрийте клапани наборів танків; відкрийте клапан нагрівання на 50% (5000). Спробуйте роботу кнопок "швидкість імітації" при наборі рівнів та нагріванні.

| Name | Type | Comment |
|---------|----------|---------------------------------|
| Dozator | <Struct> | управління дозуванням |
| CV | INT | плинна кількість доз |
| PV | INT | задана кількість доз |
| START | BOOL | запуск дозування |
| Recipe | <Struct> | рецепт |
| D1Count | INT | кількість доз з дозатору D1 |
| D2Count | INT | кількість доз з дозатору D2 |
| T_SP | INT | значення температури нагрівання |
| Delay | TIME | час витримки |

| Name | A. | Type | Value | Comment |
|----------|----|---------|-------|---------------------------------|
| RecipeT1 | | Recipe | | рецепт для T1 |
| D1Count | | INT | 3 | кількість доз з дозатору D1 |
| D2Count | | INT | 4 | кількість доз з дозатору D2 |
| T_SP | | INT | 4000 | значення температури нагрівання |
| Delay | | TIME | !#15s | час витримки |
| RecipeT2 | | Recipe | | рецепт для T2 |
| D1Count | | INT | 1 | кількість доз з дозатору D1 |
| D2Count | | INT | 2 | кількість доз з дозатору D2 |
| T_SP | | INT | 3000 | значення температури нагрівання |
| Delay | | TIME | !#1s | час витримки |
| Dozator1 | | Dozator | | управління дозуванням D1 |
| Dozator2 | | Dozator | | управління дозуванням D2 |

Рис.5.3. Структурні типи та змінні проекту.

9) Відключіться від симулятора ПЛК. Створіть програми для управління процесом з використанням мови SFC. Можна використати варіант запропонований в додатку 4 (рис. Д4.1-Д4.6). Зверніть увагу, що секція "Simulation" повинна бути першою в розділі Sections. Секція D1 створюється аналогічно як D2. Секцію CTRL_SFC рекомендується робити в останню чергу.

10) Скомпілюйте і завантажте проект в ПЛК, перевірте його роботу. Подивіться на роботу анімації редактору SFC в режимі он-лайн.

Аналіз одержаних результатів

Викладачем перевіряється виконання поставленого завдання. Студент повинен пояснити виконання програми та призначення кожного пункту виконаного завдання. *Додатково оцінюється створення власного варіанту рішення задачі або модифікація існуючого.*

Запитання для самоперевірки

1. Які обмеження використання SFC в структурі проекту UNITY PRO.
2. Перерахуйте основні елементи які організовують мережу SFC та поясніть їх призначення.
3. Поясніть як ви розумієте що таке маркер (Token) і як він циркулює в мережі SFC. Що таке множинний і одиночний маркер?
4. Які бувають кроки в SFC? Як в UNITY PRO створюються різні типи кроків?
5. Що таке дії (Action)? Чим визначається виконання дій?
6. Розкажіть які типи кваліфікаторів (специфікаторів) Ви знаєте і їх призначення.
7. Що таке змінна дії (Action Variable) і секція дії (Action Section)? Які обмеження є в змінних дії?
8. Розкажіть про правила організації переходів. Що таке змінна переходу (Transition Variable) і секція переходу (Transition Section)? Які обмеження є в змінних переходу?
9. Розкажіть про типи і правила формування відгалужень.
10. Розкажіть про макрокроки і макросекції.
11. Розкажіть про часові налаштування та контроль за помилками виконанням кроків.
12. Розкажіть про призначення секції CTRL_SFC в запропонованій програмі.

ДОДАТОК 4. Опис роботи варіанту програми SFC для вирішення задачі в лабораторній роботі.

Д4.1. Загальні принципи роботи програми. Для реалізації даної задачі використовуються 4-ри секції (рис.Д4.1): "D1" і "D2" (SFC-мовою) для управління дозаторами, "Production" (SFC-мовою) для управління приготуванням продукту, та "CTRL_SFC" (LD-мовою) для ініціалізації мереж SFC в цих секціях (скидання кроків і перехід на початковий крок). Секція "Simulation" призначена тільки для імітації об'єкта.

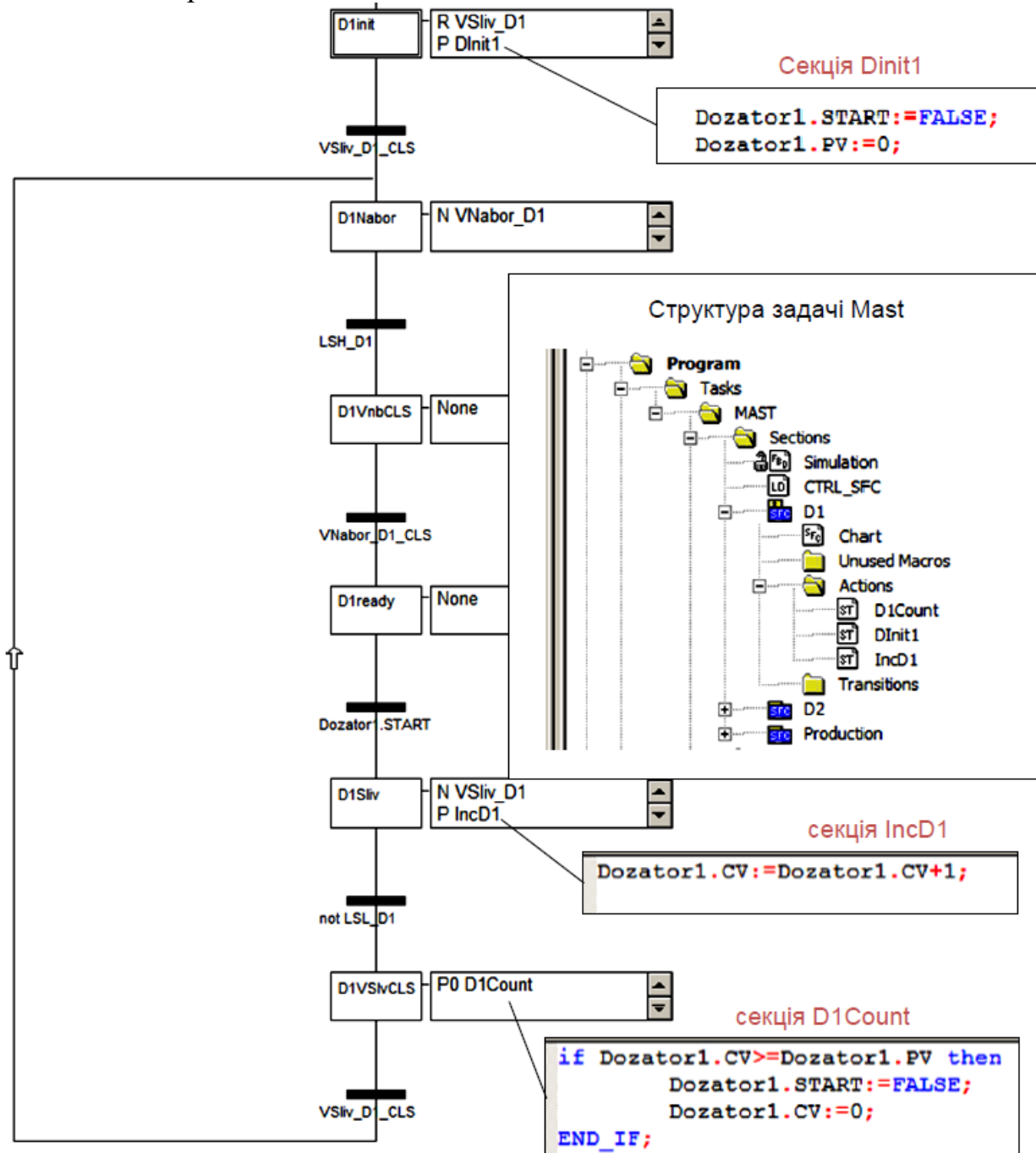


Рис.Д4.1. Структура задачі MAST та Секція "D1" для управління дозатором D1

Д4.2.Управління дозаторами. Дозатори управляються незалежними автоматами станів, які реалізовані через секції D1 та D2. Вони ідентичні по своїй структурі і відрізняються тільки використовуваними змінними. Для формування завдання (кількість доз), його запуску та виконання використовуються структурні змінні *Dozator1* та *Dozator2* заздалегідь створеного типу *Dozator* (рис.5.3).

При ініціалізації мережі SFC (рис.Д4.1) закривається клапан *Vsliv_D1*, та на один цикл (специфікатор P) запускається секція *Dinit1*, де обнулюються поля START та PV структури управління дозуванням. Коли клапан зливу закривається, що сигналізується його датчиком кінцевого положення, - відкривається клапан *VNabor_D1*. Клапан набору буде до тих пір відкритий, поки крок *D1_Nabor* активний (специфікатор N), тобто поки не спрацює сигналізатор верхнього рівня *LSH_D1*. Після спрацювання датчика кінцевого положення закриття клапану набору, програма переходить до кроку *D1Ready*. На цьому кроці програма очікує команду дозування *Dozator1.START*, яка повинна надійти (змінитися в TRUE) з іншої частини програми.

Процес роботи дозування заключається у відпрацюванні заданої кількості вивантажень дози, що підраховується змінною-лічильником *Dozator1.CV*. Тому при *Dozator1.START=TRUE*, протягом одного циклу активності *D1Sliv* відбувається збільшення значення *Dozator1.CV* на 1. Крок буде активний а клапан *Vsliv_D1* буде відкритий до тих пір, поки не відключиться сигналізатор нижнього рівня, що сигналізує про порожність дозатору.

Крок *D1VslvCLS* буде активний до тих пір, поки не спрацює датчик кінцевого положення закриття клапану зливу (*Vsliv_D1_Cls*). На останньому циклі активності кроку запускається секція *D1Count*, в якій при досягненні кількості доз рівній уставці, скидається команда *Dozator1.START* і обнулюється плинне значення лічильника.

Слід зазначити, що після останньої вивантаженої дози, дозатор все одно набирається. Це задовольняє умовам задачі.

Д4.3.Управління приготуванням продукту. Секція управління приготуванням продукту "Production" побудована з використанням макрокроків (рис.Д4.2). При ініціалізації мережі SFC проходить закривання всіх клапанів танків T1 та T2. Ініціалізація закінчується при спрацюванні датчиків положення "закрито" клапанів набору та зливу.

Для одночасності процесів приготування продуктів в танках T1 та T2 використане паралельне відгалуження. Кожна з гілок після ініціалізації переходить в стан готовності (кроки *T1Ready* та *T2Ready*):

- для танку T1 це очікування кнопки *Pusk*;
- для танку T2 це закінчення набору *T1*, що сигналізується активністю кроку *T1WaitT2* (*T1WaitT2.x=TRUE*).

Кожна з паралельних гілок ідентичні одна одній, за винятком певних особливостей, що стосуються координації їх роботи. Тому розглянемо тільки гілку для управління танком T1 і особливості гілки T2. Гілка управління T1 включає 3 макрокроки, які відповідають за певний етап роботи: *T1Nabor* – за набір і дозування, *T1Nagrev* – за нагрівання і витримку, *T1Sliv* – за вивантаження продукту.

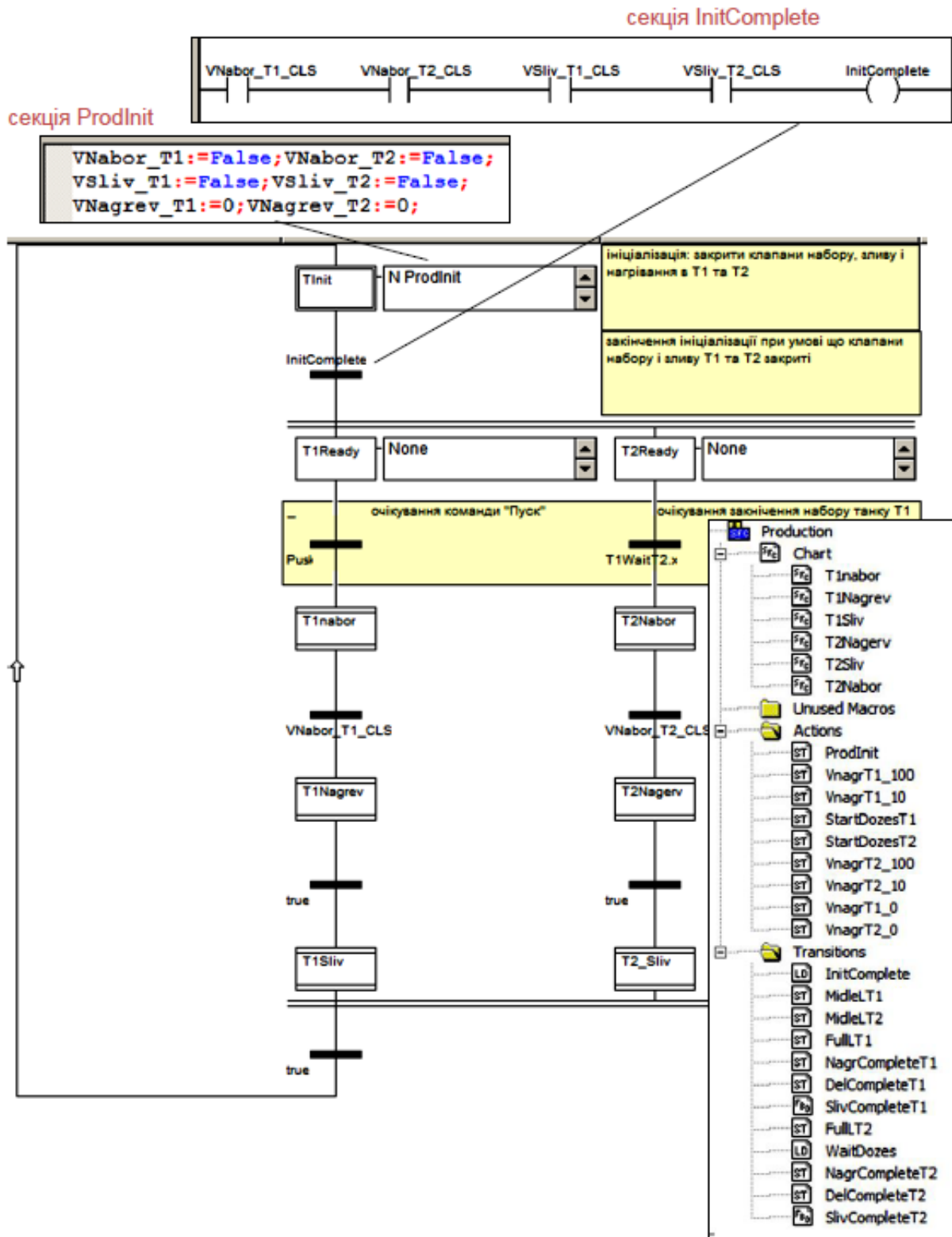


Рис.Д4.2. Секція "Production" для управління виробництвом продукції

Макросекція кроку *T1Nabor* показана на рис. Д4.3. Вхідний крок макросекції активує відкриття клапану *VNabor_T1*, специфікатор *S* вказує на те, що після деактивації кроку, клапан повинен залишитись відкритим. Після наповнення 50% (умова в секції *MidleLT1*: $LE_{T1} \geq 5000$) мережа SFC знову ділиться на 2 паралельні гілки: ліва гілка для повного наповнення танку, права – для дозування. Таке розгалуження дає можливість вчасно закрити клапан набору, навіть якщо дозування ще не закінчилось.

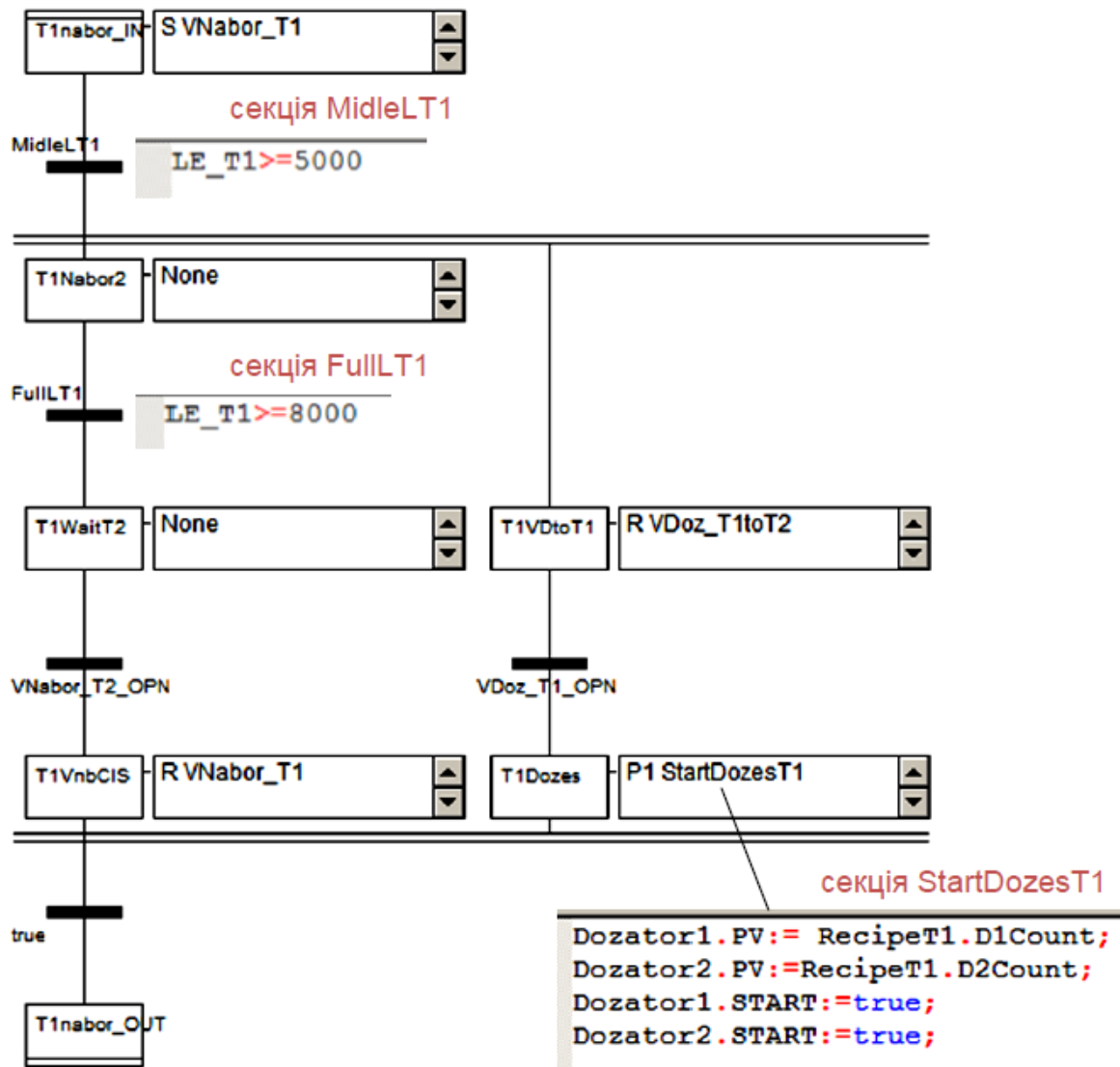


Рис.Д4.3. Макросекція "T1Nabor"

Після наповнення рівня до 80%, ліва гілка переходить до кроку *T1WaitT2*, активність якого (*T1WaitT2.x=TRUE*) сигналізує гілці управління танком T2, що необхідно відкривати клапан набору танку T2 (рис.Д4.2, перехід *T1WaitT2.x*). Після спрацювання датчика положення "відкрито" клапану набору танку T2 *VNabor_T2_OPN*, закривається клапан набору T1 (*R VNabor_T1*).

Перший крок правої гілки *T1VDtoT1* перемикає 3-х ходовий клапан в положення дозування в танк T1 (*VDoz_T1toT2=FALSE*). Після спрацювання датчика положення запускається процес дозування шляхом задання кількості доз для D1 і D2 згідно рецепту, і запуску дозаторів шляхом подання команди START (див.Д.4.2).

Після початку дозування, якщо команда закриття на клапан набору відправлена, тобто коли активні кроки *T1VnbCIs* і *T1Dozes*, йде вихід з макросекції.

Умовою переходу до макрокroku *T1Nagrev* (Рис.Д4.2) є спрацювання датчика кінцевого положення "закрито" клапану набору T1. Для T2, макросекція набору має аналогічну структуру, за винятком відсутності кроку очікування відкриття сусіднього танку і присутності кроку очікування закінчення дозування (рис.Д4.4). Таким чином програма набору танку T2 перед

дозуванням буде чекати поки дозатор не відпрацює всі замовлені для нього дози, тобто поки *Dozator1.START* та *Dozator2.START* не стануть рівними *FALSE*.

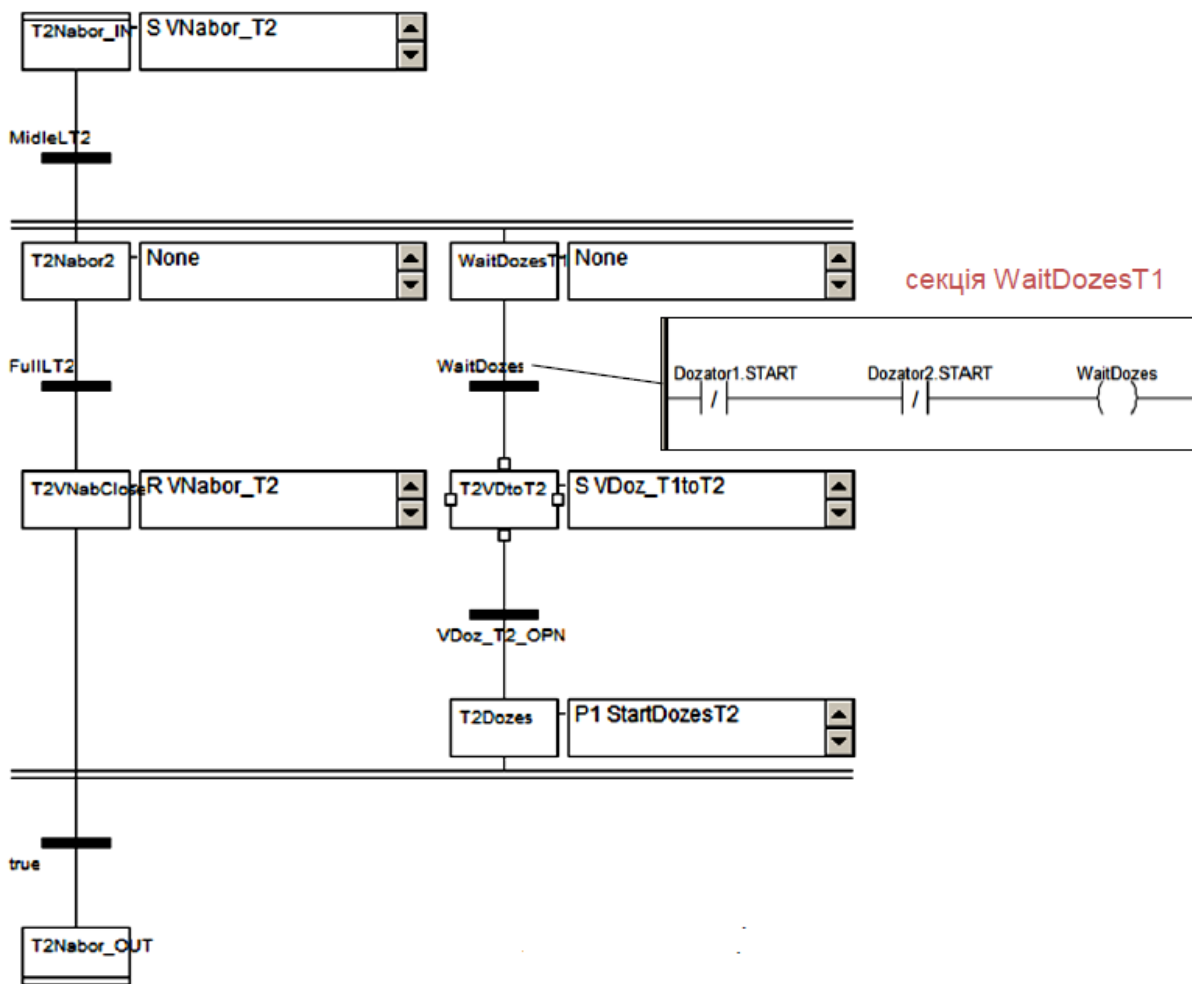


Рис.Д4.4. Макросекція "T2Nabor"

Макросекція нагрівання починається з відкриття клапану нагрівання на 100%. Специфікатор P1 вказує на те, що дія виконається один раз, при активації кроку. При досягненні заданої в рецепті температури, маркер отримує крок *T1Vytrym*. При активації кроку клапан нагрівання прикривається до 10%, при деактивації (специфікатор P0), клапан закривається повністю. Сам крок залишається активним протягом заданого в рецепті часу, тобто поки не спрацює умова переходу задана в секції *DelCompleteT1: T1Vytrym.t >= RecipeT1.Delay*, де *t* – час активності кроку. Після виходу з макросекції нагрівання, маркер автоматично переходить до макрокroku вигрузки T1Sliv (див. рис.Д4.2), так як умова переходу завжди TRUE.

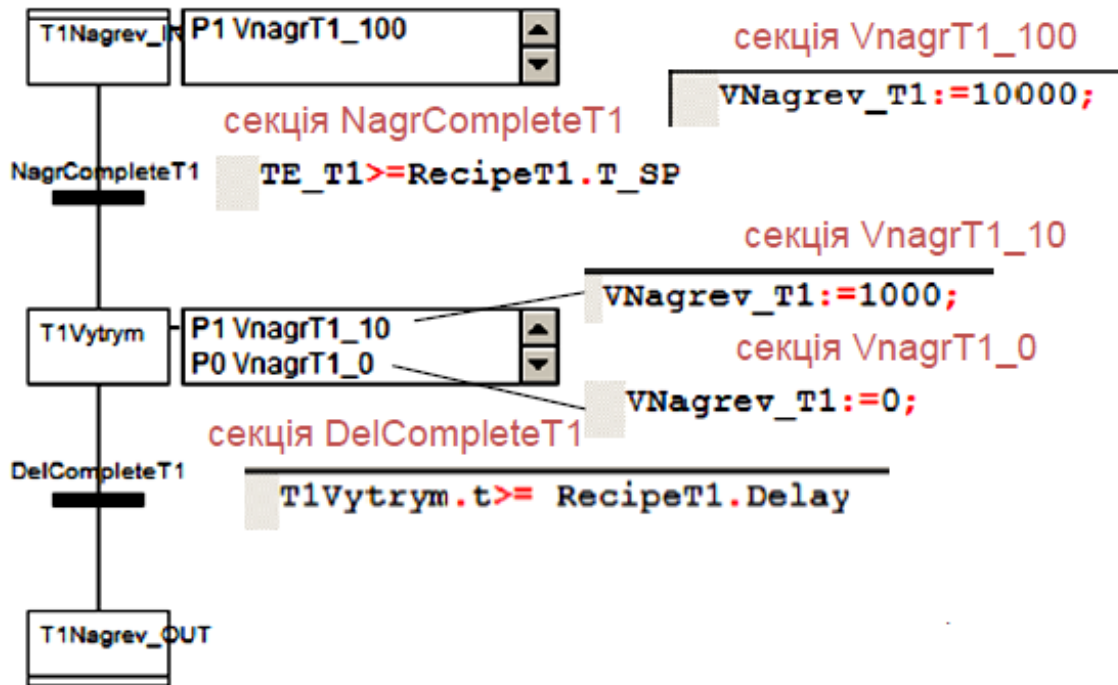


Рис.Д4.5. Макросекція "T1Nagrev"

Макросекція *T1Sliv*, починається з відкриття клапану зливу (рис.Д.4.6). При досягненні рівня в танку T1 менше ніж 1% (100 в діапазоні 0-10000), активується крок витримки *PauseT1*, яка потрібна для повного випорожнення танку. Час затримки досягається за рахунок налаштування мінімального часу активності кроку *Delay=T#5s*. Умова переходу TRUE, відразу пропускає маркер на крок *T1VslivClose*, в якому відбувається закриття клапану зливу. Після спрацювання датчика "закритий" клапану зливу T1, і активності останнього кроку в гілці управління T2 маркер покидає макросекцію, з'єднується з маркером з паралельної гілки і переходить на крок ініціалізації (рис.Д4.2) .

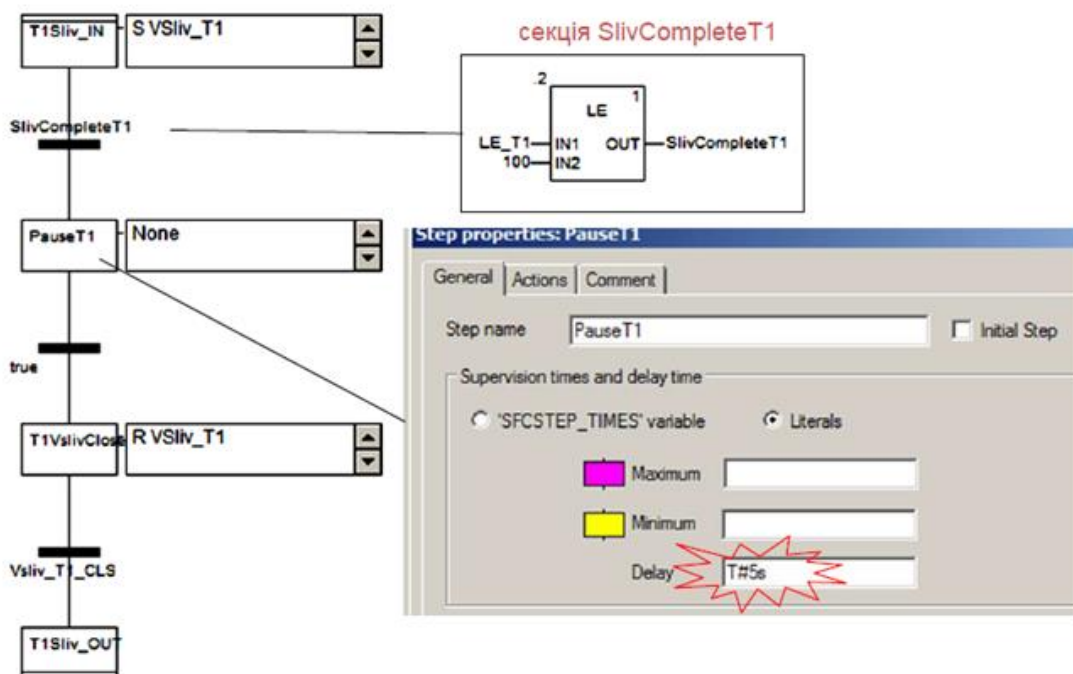


Рис.Д4.6 Макросекція *T1Sliv*

Д4.4. Секція ініціалізації мереж SFC. Для запобігання "зависання" маркера на певному кроці при нештатній ситуації, в програмі передбачена секція ініціалізації мереж SFC під назвою "CTRL_SFC". Секція написана на мові LD (рис. Д4.7) і виконується на кожному циклі задачі MAST, незалежно від активності кроків SFC. При спрацюванні змінної-команди *InitSFC=TRUE* (рис.5.1, кнопка "ІніціалізКроків") через виклики системної функції INITCHART ініціалізуються всі мережі SFC секцій Production, D1 та D2:

- при переході *InitSFC* з FALSE->TRUE скидає (деактивує) всі кроки;

- при переході *InitSFC* з TRUE -> FALSE ініціалізує мережі SFC, тобто активує початковий крок.

Слід зазначити, що UNITY PRO має ряд функцій, які дають можливість ручного управління переходами та активації кроків.

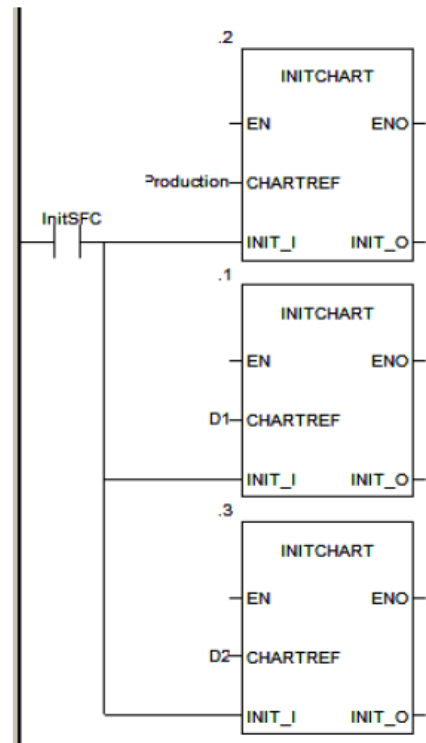


Рис.Д4.7. Секція CTRL_SFC для ініціалізації SFC

ДОДАТОК 5. Робота з SFC в середовищі UNITY PRO.

Д5.1. Редактор SFC. SFC секція окрім діаграми, що включає мережі SFC, включає в себе секції дій, секції переходів, всі включені та видалені макросекції (рис.Д.5.1).

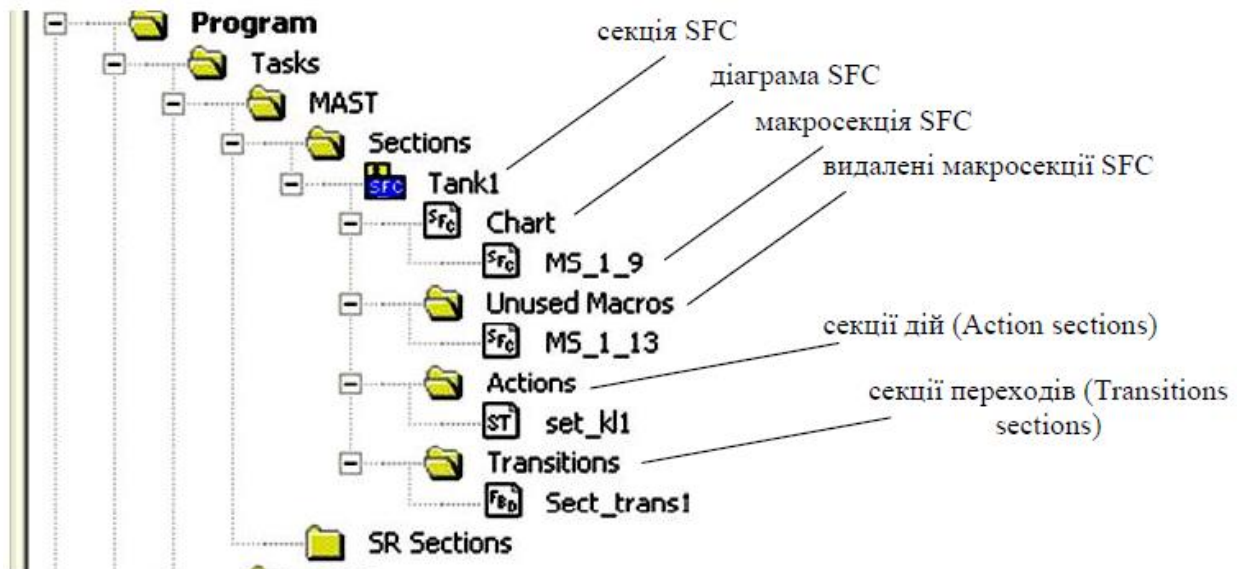


Рис.Д5.1. Структура секції SFC.

Редактор *діаграми SFC (Chart)* ділиться на комірки (200 рядків на 32 колонки), в яких розміщуються кроки, переходи та стрибки. Зв'язки, відгалуження та сходження не потребують окремих комірок. У кожній секції SFC може бути максимум 1024 кроки, включаючи всі кроки макросекцій. Для створення мереж SFC використовується палітра інструментів (рис.Д.5.2). Серед всіх типів кроків в палітрі відсутні: початковий крок, так як він робиться зі звичайного кроку через його основні властивості; вхідні та вихідні кроки макросекцій, так як вони створюються автоматично з макросекцією.

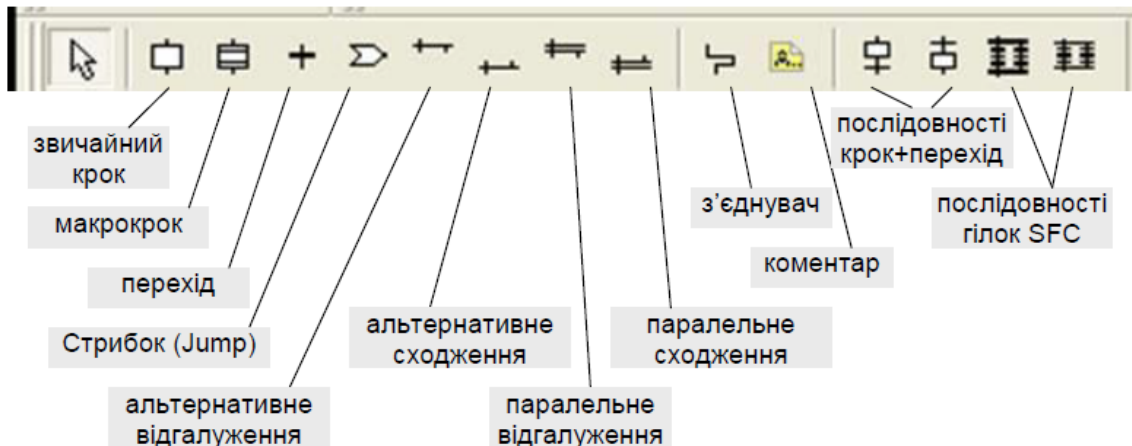


Рис.Д5.3.Палітра інструментів SFC

Доступ до властивостей кроку проводиться через його контекстне меню. У основних властивостях (вкладка General, рис.Д.5.4) вказується ім'я кроку, часові налаштування кроку та чи буде даний крок початковим (Initial Step). Часові налаштування вказуються часовими літералами (Literals) або через змінну ('SFCSTEP_TIMES' variable).

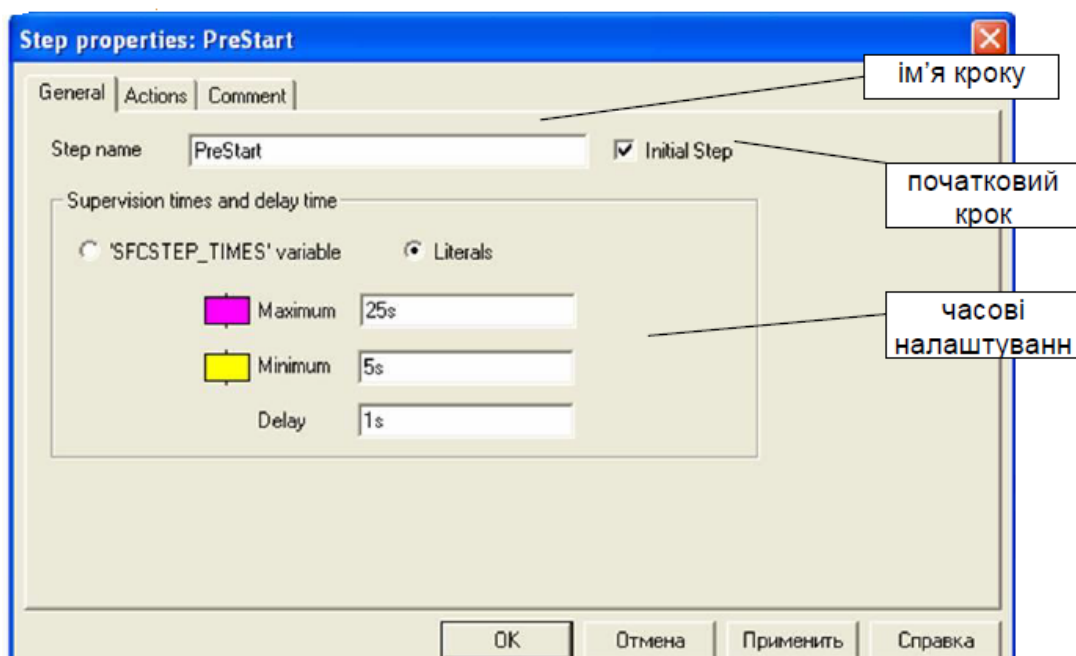


Рис.Д5.4. Вікно загальних властивостей кроку SFC.

Дії для кожного кроку налаштовуються у вкладці Actions, що доступне у вікні властивостей кроку. Тип дії (змінна чи секція) вибирається відповідним перемикачем Variable/Section, під ним вказується назва змінної або секції. Кожна дія супроводжується вибраним специфікатором (Qualifier). Для дій з специфікаторами L, D, DS вказуються часові літерали або змінні для визначення затримок.

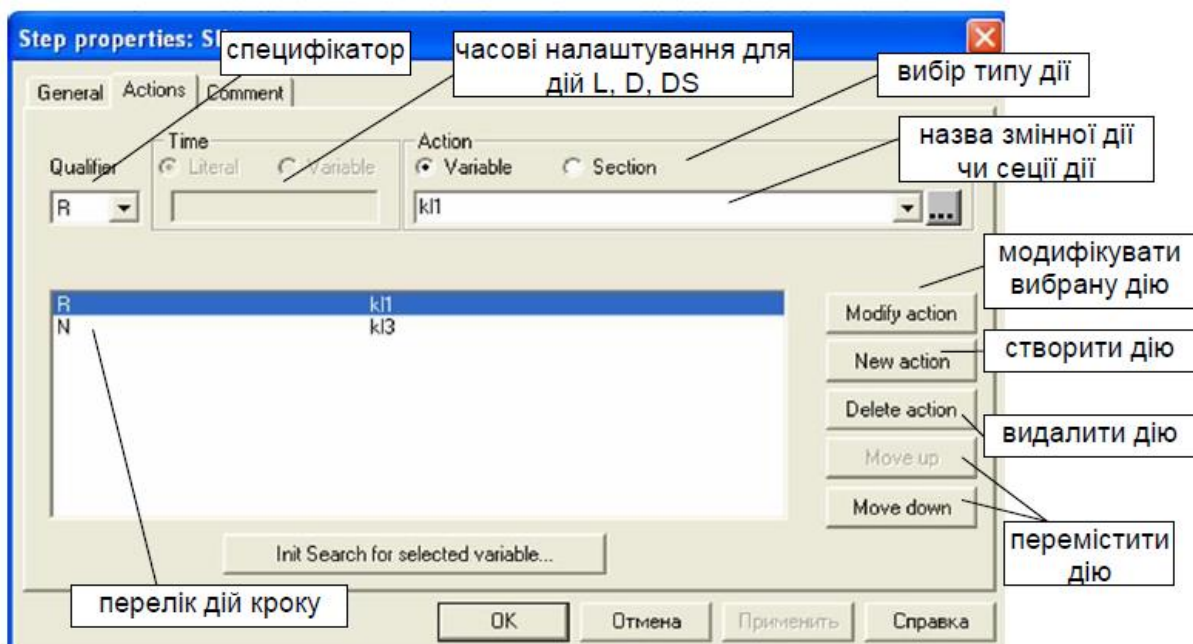


Рис.Д5.5. Вікно налаштування дій для кроку SFC.

Діаграму SFC можна відобразити в розширеному вигляді "View->Expanded Display", де з правого боку від кроків вказується список дій.

Для відгалужень і сходжень через вікно властивостей налаштовується кількість гілок і номер точки входу (для відгалужень) або виходу (для сходжень) основної гілки.

Д5.2. Дії та специфікатори (кваліфікатори).

Таблиця Д5.1 Типи специфікаторів

| Специфікатор | Призначення | Опис |
|--------------|------------------------------|--|
| N/None | без фіксації | дія активна тільки при активності кроку, крок активний – дія TRUE, крок не активний - FALSE |
| S | Встановити з фіксацією | дія встановлюється в TRUE і залишається такою, навіть при деактивації кроку; дія повинна скинутися в FALSE специфікатором R в іншому кроці тієї ж секції SFC |
| R | Скинути | дія встановлюється в FALSE і залишається такою, навіть при деактивації кроку; |
| P | Імпульс по передньому фронту | при активації кроку дія стає активною (встановлюється в TRUE) на один цикл задачі MAST |
| P1 | Імпульс по передньому фронту | аналогічно специфікатору P, однак дія виконується завжди першою незалежно від розміщення в списку дій |
| P0 | Імпульс по | при деактивації кроку |

| | | |
|----|-----------------------------------|---|
| | задньому фронті | у дія стає активною (встановлюється в TRUE) на один цикл задачі MAST |
| L | Обмеження по часу | дія активується (встановлюється в TRUE) одночасно з активацією кроку, деактивується (скидається в FALSE) при закінченні встановленого часу (тривалість задається) або при деактивації кроку |
| D | Затримка на включення | подібно до роботи таймеру TON; при активації кроку запускається внутрішній таймер (тривалість задається), після спрацювання якого дія активується (встановлюється в TRUE); дія деактивується разом з деактивацією кроку |
| DS | Затримка на включення з фіксацією | подібно до дії визначеного специфікатором D, однак при деактивації кроку дія залишається активною; дія повинна скинутися в FALSE специфікатором R в іншому кроці тієї ж секції SFC |

Д5.3. Часові налаштування та контроль за виконанням кроків.

Таблиця Д5.2 Поля змінної кроку типу SFCSTEP_STATE.

| поле | Тип | Опис |
|---------|------|--|
| t | TIME | час активності кроку; при деактивації кроку даний час запам'ятовується, а при повторній активації обнуляється |
| x | BOOL | активність кроку: TRUE - крок активний (має маркер), FALSE - крок неактивний |
| tminErr | BOOL | помилка мінімального контрольного часу; якщо час виконання кроку менше ніж заданий в мінімальному контрольному часі tminErr=TRUE, інакше FALSE; автоматично скидається в FALSE при повторній активації кроку |
| tmaxErr | BOOL | помилка максимального контрольного часу; якщо час виконання кроку більше ніж заданий в максимальному контрольному часі tmaxErr =TRUE, інакше FALSE; автоматично скидається в FALSE при повторній активації кроку |

ЛАБОРАТОРНА РОБОТА № 6

Реалізація каскадної САР з програмним задатчиком

Тривалість: 2 акад. години.

Мета: ознайомитися з основами побудови контурів аналогового регулювання в середовищі UNITY PRO з використанням бібліотечних блоків сімейства ControlLIB (регуляторів, блоків ручного управління, блоків управління уставками та інші).

Завдання для виконання роботи

Загальна постановка завдання. Необхідно створити проект в UNITY PRO для реалізації поставленої задачі (рис.6.1) з використанням FFB бібліотеки ControlLIB. Відлагодження проекту зробити з використанням готового програмного імітатора об'єкту та операторського екрану.

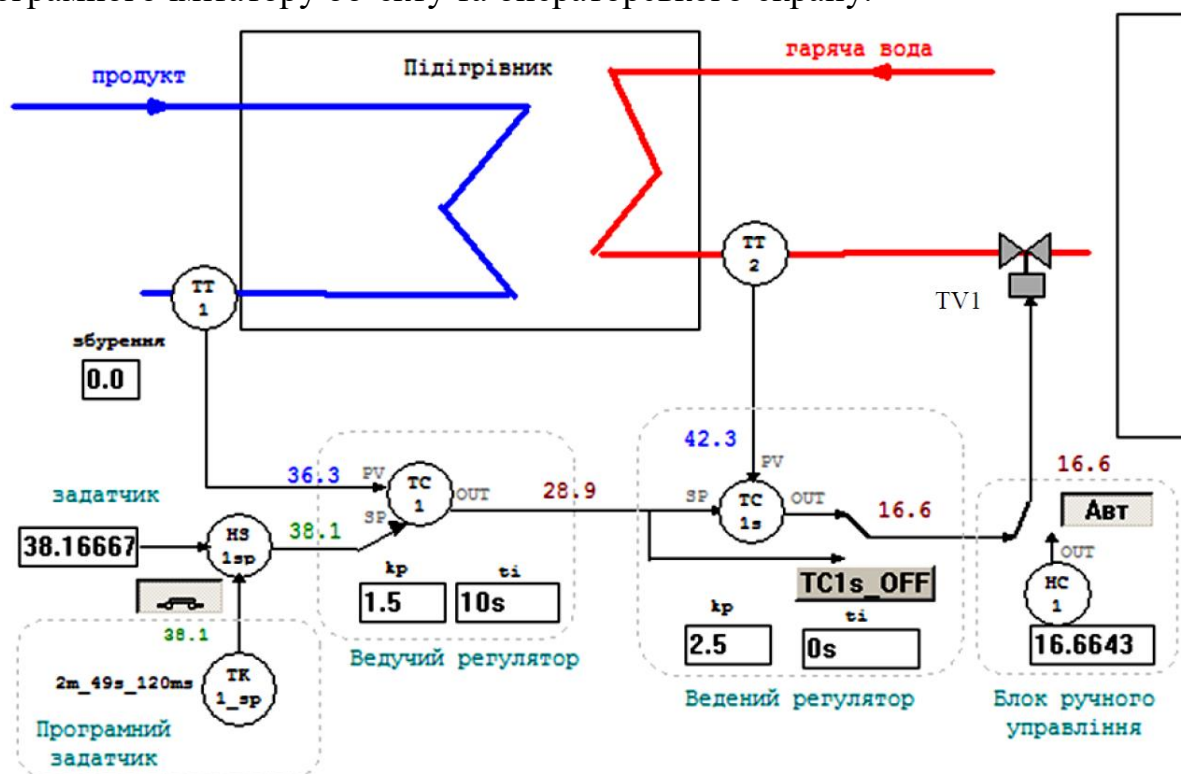


Рис.6.1. Операторський екран для контролю та управління процесом нагрівання

Опис об'єкту. Технологічна установка являє собою теплообмінник для підігріву рідкого продукту (рис.6.1) за допомогою гарячої води. Температура продукту на виході вимірюється датчиком температури ТТ1 (0-100°C, вхід ПЛК %IW0.1.2) який вмонтований на трубопроводі на відстані кількох метрів від підігрівника, а регулюється витратою гарячої води регулюючим клапаном TV1 на виході підігрівника (0-100% ХРО, вихід %QW0.2.0). Додатково вимірюється також температура води в трубопроводі безпосередньо на виході підігрівника за допомогою датчика ТТ2 (0-100°C, вхід ПЛК %IW0.1.1). Інерційність об'єкту по каналу ТТ2 менша ніж по ТТ1.

Опис функцій. Система управління повинна забезпечити регулювання температури ТТ1 з підтримкою наступних функцій:

1) стабілізація температури на виході підігрівника на заданому оператором або програмним задатчиком значенні, з використанням каскадного регулятора, де:

а- ведучий регулятор (ПІ режим) стабілізує температуру ТТ1;

б- ведений регулятор (ПІ режим) служить для швидкої реакції контуру регулювання на зміну малоінерційного ТТ2;

2) можливість настройки коефіцієнтів K_p та T_i з операторського екрану;

3) можливість ручного управління виконавчим механізмом TV1 з операторського екрану при переключенні в ручний режим з забезпеченням безударності переходу;

4) можливість формування завдання програмним задатчиком в залежності від часу по наступній залежності:

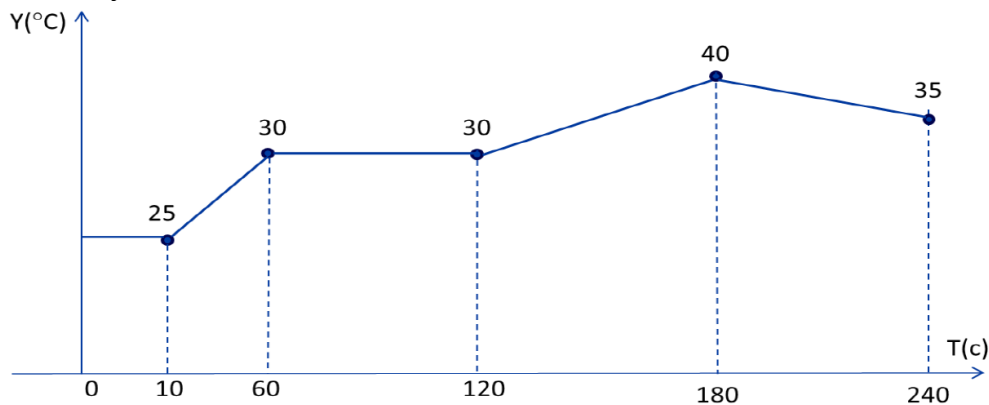


Рис.6.2. Залежність завдання температури від часу

5) можливість переключення з ручного завдання на програмне управління та навпаки в будь який момент часу з забезпеченням безударності переходу;

6) можливість виключення веденого регулятора з каскаду (з забезпеченням безударності переходу), тобто переводу контуру в режим прямого ПІ-регулювання по температурі ТТ1;

7) при ручному управлінні виконавчим механізмом ведучий регулятор повинен фіксувати своє вихідне значення в останньому положенні.

Лабораторна установка

Апаратне забезпечення: ПК.

Програмне забезпечення: UNITY PRO V \geq 4.0.

Порядок виконання роботи. Заходи безпеки

Необхідно дотримуватись стандартних заходів безпеки при роботі з ПК.

Перед виконанням лабораторної роботи ознайомтесь з додатком 6.

1) Запустити на виконання UNITY PRO. Створити новий проект з ПЛК M340.

2) Імпортувати файл апаратної конфігурації:

Project Browser -> контекстне меню **Configuration** -> **Import** "Робочий стіл\тека Лаб_ППК\lab6_Cascad_HW.XHW".

3) Подивитися конфігурацію обладнання для даної задачі.

4) Імпортувати файл з секцією імітатора об'єкту управління:

Project Browser -> Program-> Tasks -> Mast -> контекстне меню Sections -> Import "Робочий стіл\тека Лаб_ППК\LAB6_Cascad_simul.XBD";

5) Імпортувати змінні (див. рис.6.3):

Project Browser -> Variables & FB Instances -> Import ""Робочий стіл\тека Лаб_ППК\lab6_Cascad_vars.XSY"

The image displays three screenshots of the 'Variables & FB Instances' window in a software interface, showing different views of project variables.

Top Screenshot: Function Blocks

| Name | r | Type | Comment |
|-----------|---|------------|--|
| HC1 | | MS | Блок ручного управління |
| smHeater1 | | smHeatExch | Імітаційна модель підігрівача |
| HS1sp | | SP_SEL | Перемикач ручного/програмного завдання |
| smk1 | | smValve | Імітаційна модель клапану |
| smDEL1 | | smDELAY | Імітаційна модель блоку запізнювання |
| SMPL1 | | SAMPLETM | Диспетчер виклику |
| TC1 | | PI_B | Ведучий регулятор |
| TC1s | | PI_B | Ведений регулятор |
| TK1 | | TON | Таймер програмного задатчика |

Middle Screenshot: Global Variables

| Name | Type | Value | Comment |
|----------|------|-------|--|
| smInit | BOOL | | Ініціалізація імітаційної моделі |
| TC1_AUTO | BOOL | | 1 - включити автоматичний режим роботи контуру |
| TC1s_OFF | BOOL | | 1 - відключити ведений регулятор |
| TK1_ON | BOOL | | 1 - включити програмний задатчик |
| smZ1 | REAL | | імітація збурення |
| TC1_OUT | REAL | | Вихід ведучого регулятора |
| TC1_SP | REAL | 20.0 | Уставка для ведучого регулятора |
| TC1s_OUT | REAL | | Вихід веденого регулятора |
| TK1_SP | REAL | | Вихід завдання з програмного задатчика |
| TT1 | REAL | | T продукту на виході підігрівача |
| TT2 | REAL | | T гарячої води на виході підігрівача |
| TV1 | REAL | | Клапан подачі гарячої води |

Bottom Screenshot: Parameters

| Name | Type | Value | Comment |
|-------------|-------------|-------|--------------------------------|
| HC1_PARA | Para_MS | | Параметри ручного задатчика |
| out_min | REAL | 0.0 | |
| out_max | REAL | 100.0 | |
| inc_rate | REAL | | |
| dec_rate | REAL | | |
| outbias | REAL | | |
| use_bias | BOOL | FALSE | |
| bumpless | BOOL | FALSE | |
| TC1_PARA | Para_PI_B | | Параметри введучого регулятора |
| TC1s_PARA | Para_PI_B | | Параметри веденого регулятора |
| HS1sp_PA... | Para_SP_SEL | | Параметри перемикача завдань |
| sp_min | REAL | 0.0 | |
| sp_max | REAL | 100.0 | |
| bump | BOOL | | |
| track | BOOL | | |
| rate | REAL | | |

Рис.6.3. Змінні проекту

15) Виведіть ведений регулятор з каскаду (виключити TC1s_OFF). Змініть завдання на 50°C, після закінчення перехідного процесу знов на 40°C. Оцініть приблизний час перехідних процесів.

16) Включіть каскадне управління. Включіть програмний задатчик. При програмному управлінні змінюйте збурення на $\pm 2^\circ\text{C}$.

Аналіз одержаних результатів

Викладачем перевіряється виконання поставленого завдання. Студент повинен пояснити виконання програми та призначення кожного пункту виконаного завдання. *Додатково оцінюється створення власного варіанту рішення задачі або модифікація існуючого.*

Запитання для самоперевірки

1. Поясніть функціональну схему з рис.6.1. Що таке каскадне управління?
2. Наведіть перелік FFB та їх призначення, які використані для реалізації контуру регулювання в даній лабораторній роботі.
3. Поясніть принцип роботи блоків типу SP_SEL з використанням їх функціональної схеми.
4. Поясніть принцип роботи блоків типу PI_V з використанням їх функціональної схеми.
5. Поясніть принцип роботи блоків типу MS з використанням їх функціональної схеми.
6. Поясніть принцип роботи процедури LOOKUP_TABLE1.
7. Поясніть яким чином забезпечується переключення контуру регулювання в ручний режим в даній лабораторній роботі. Як при цьому ведуть себе всі регулятори і чому? Чи залежить їх поведінка від включення/виключення веденого регулятора з каскаду?
8. Поясніть яким чином формуються завдання для контуру регулювання в даній лабораторній роботі.
9. Поясніть яким чином забезпечується відключення веденого регулятора з каскаду в даній лабораторній роботі.
10. Поясніть яким чином забезпечується ручне управління в даній лабораторній роботі.

ДОДАТОК 6. Опис варіанту функціональної схеми контуру регулювання та програми для його реалізації в UNITY PRO.

Д6.1. Загальні принципи роботи контуру. Функціональна схема контуру регулювання температури продукту показана на операторському екрані (рис.6.1).

Контур включає:

- канали вимірювання (ТТ1, ТТ2);
- канал формування завдання, який складається з ручного задатчика, програмного задатчика ТК1_sp (формування програмного управління по часу), перемикача ручного/програмного завдання HS1sp (включена кнопка – запуск програмного задатчика) ;
- ведучого регулятора TC1 (ПІ закон), який на основі сформованого завдання (вхід SP) та вимірювального значення (ТТ1 на вхід PV), формує сигнал управління (вихід OUT), який подається на ведений регулятор TC1s; для регулятора налаштовуються K_p та T_i ;
- веденого регулятора TC1s (ПІ закон), який на основі сформованого завдання (TC1.OUT на вхід SP) та вимірювального значення (ТТ2 на вхід PV), формує сигнал управління (вихід OUT), який подається на виконавчий механізм TV1 через перемикач TC1s_OFF; перемикач TC1s_OFF дозволяє переключити TV1 на вихід ведучого регулятора TC1.OUT, тим самим відключивши ведений регулятор з контуру управління; для регулятора налаштовується K_p а також T_i , ненульове значення якого дозволяє перевести регулятор в ПІ-режим;
- блок ручного управління HC1, який дає можливість переключитися в режим ручного управління виконавчим механізмом TV1 з операторського екрану; кнопка АВТ для переключення режиму (нажата - автоматичний режим);
- канал управління (TV1);

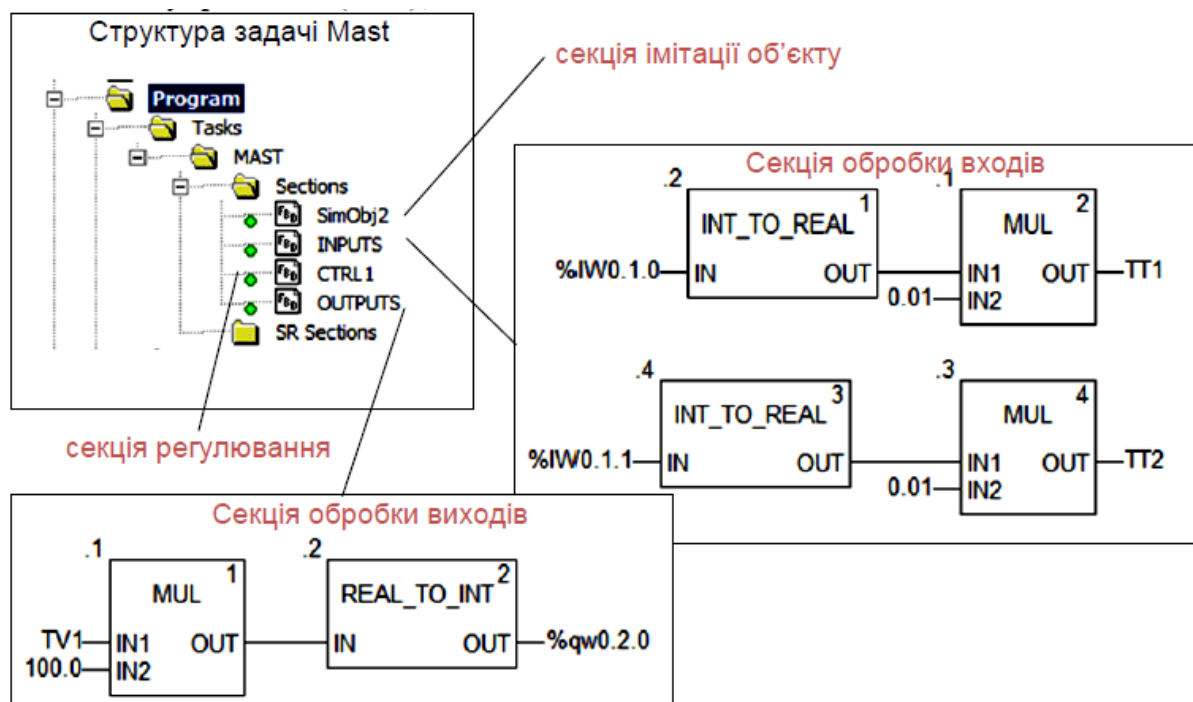


Рис.Д.6.1. Структура задачі MAST

Д6.2. Загальні принципи роботи програми. Для реалізації даної задачі використовуються 4-ри секції (рис.Д6.1): секція "INPUTS" – для обробки вхідних каналів вимірювання (датчики температури); секція "OUTPUTS" – для обробки вихідного каналу правління; секція "CTRL1" – для реалізації контуру регулювання.

Секція "Simulation" призначена тільки для імітації об'єкта.

У секції "INPUTS" оцифровані значення аналогових входів масштабуються шляхом множення на коефіцієнт (діапазон 0-10000 в 0-100°C).

У секції "OUTPUTS" аналогічним чином проводиться масштабування для значення виконавчого механізму TV1, тільки в зворотному напрямку (діапазон 0-100%ХРО в 0-10000).

Д6.3. Реалізація програмного задатчика. Для реалізації програмного задатчика у секції "CTRL1" перед викликом контуру регулювання використана процедура LOOKUP_TABLE1, яка реалізовує кусково-лінійну інтерполяцію за заданими вузловими точками. Вузлові точки задаються парами значень $X_i Y_i$, перше з яких – час в секундах, друга – задане значення температури в цій точці. Таким чином на вході X процедура буде отримувати значення часу роботи програмного задатчика в форматі REAL, в залежності від якого a на виході Y буде формувати задане значення (уставку для регулятора) $TK1_SP$.

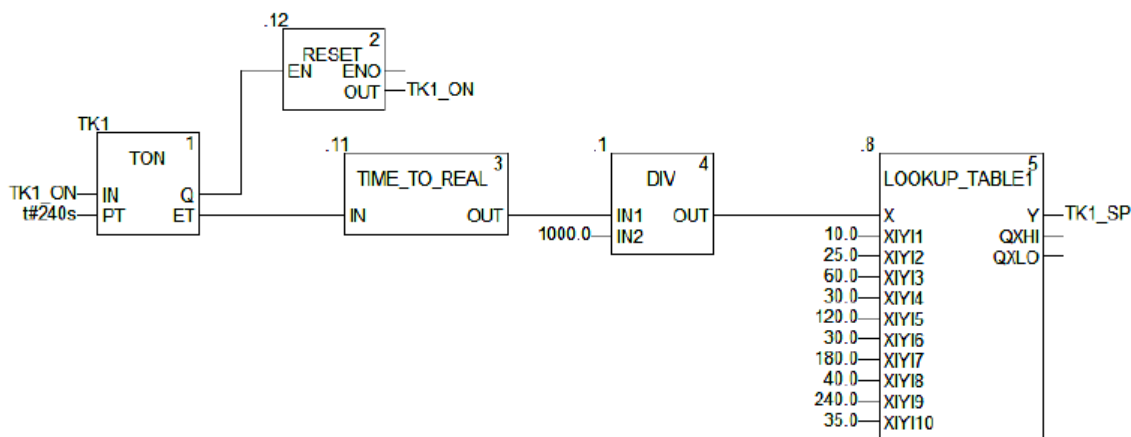


Рис.Д.6.2. Реалізація програмного задатчика

Для формування часу роботи задатчика використовується таймер $TK1$, який стартує командою запуску задатчика $TK1_ON$, та налаштований на повний час його роботи 240с. Спрацювання виходу $TK1.Q$, тобто по закінченню роботи програмного задатчика, приводить до скидання команди $TK1_ON$. У зв'язку з тим, що вихід таймеру $TK1.ET$ видає значення типу TIME, воно перетворюється в тип REAL, а отримані мілісекунди шляхом ділення на 1000 переводяться в секунди.

Д6.4. Загальні принципи реалізації контуру регулювання. Фрагмент програми реалізації контуру регулювання показана на рис.Д.6.3. Старші версії UNITY PRO (<6.0) не підтримують можливість безпосереднього з'єднання в FBD виходів типу INOUT та IN (або OUT). Тому для цих версій UNITY PRO наведений фрагмент треба виконати як на рис.Д.6.4.

Використовуючи SMPL1 та входи EN функціональних блоків, блоки HS1sp(перемикача завдань), TC1(ведучий регулятор) та TC1s(ведений

регулятор) викликаються періодично з часом 500 мс, послідовно один за одним. Це дає змогу зменшити навантаження на ресурси ПЛК та виконавчих механізмів в автоматичному режимі. Однак для ручного режиму необхідна швидка реакція системи на дії оператора, тому блок ручного завдання викликається з кожним циклом (вхід EN не задіяний).

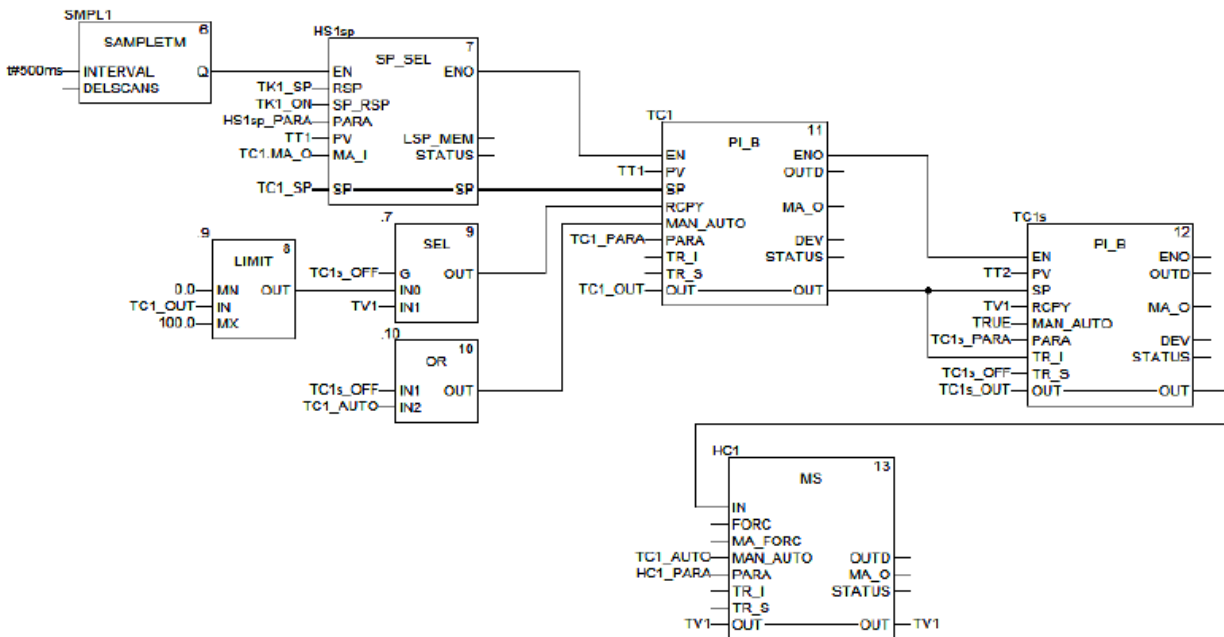


Рис.Д.6.3. Фрагмент секції з реалізацією контуру регулювання (для UNITY PRO ≥ 6.0)

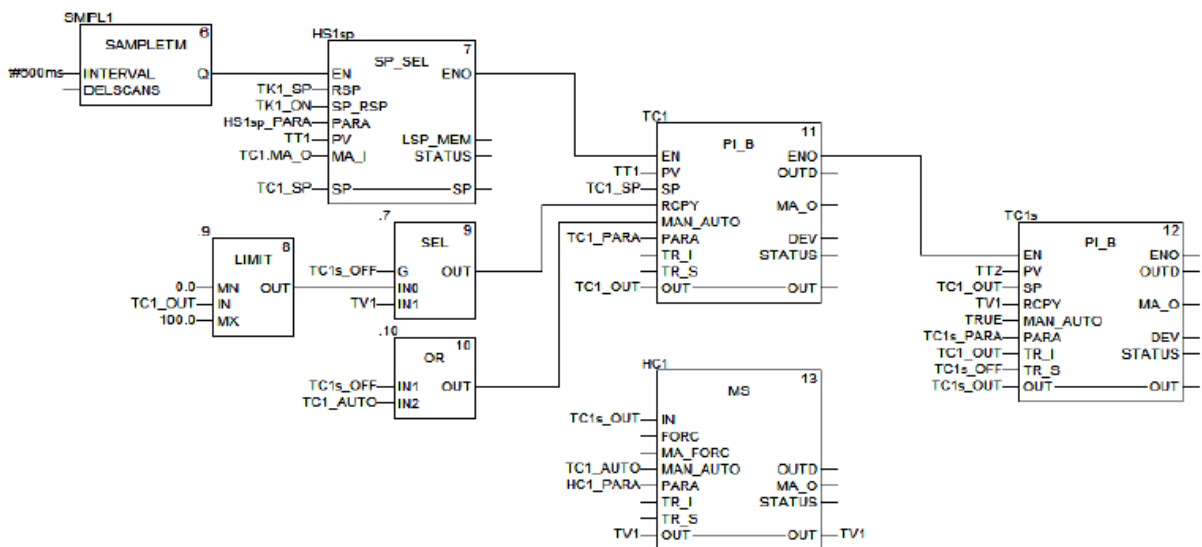


Рис.Д.6.4. Фрагмент секції з реалізацією контуру регулювання (для UNITY PRO ≥ 4.0)

Д6.5. Реалізація каналу формування завдання. Вихід блоку *HS1sp.SP* через змінну *TC1_SP* формує завдання для блоку *TC1*, вихід якого в свою чергу формує задання *TC1s*. Таким чином формується каскадне регулювання. Блок перемикавання завдання *HS1sp.SP* на вході *RSP* (дистанційне завдання) отримує значення в змінній *TK1_SP* з програмного задатчика (див.Д.6.3). При

активації оператором програмного управління ($TK1_ON=TRUE$), через вхід SP_RSP блок $HS1sp$ переключається в дистанційний режим, при якому $HS1sp.SP=HS1sp.TK1_SP$. Це ж значення копіюється в змінну $TC1_SP$, так як вона прив'язана до входу/виходу SP . У локальному режимі ($TK1_ON=FALSE$) оператор може змінювати уставку безпосередньо змінюючи $TC1$.

Дб.6. Реалізація каскадного регулятора. Ведучий регулятор $TC1$ реалізований через функціональний блок типу PI_B . Його задача забезпечити значення температури $TT1$ на рівні заданому $TC1_SP$.

Ведений регулятор $TC1s$ теж реалізований через функціональний блок типу PI_B , однак по замовченню він працює в П-режимі, так як інтегральна складова відключена ($Ti=0$). Його задача швидко зреагувати на зміну менш інерційної змінної процесу $TT2$. Таким чином велику частину збурень швидко компенсує ведений регулятор, а ведучий регулятор забезпечить точність регулювання.

Враховуючи вимоги до функцій контуру регулювання, ведучий регулятор повинен мати можливість працювати в каскаді з веденим регулятором ($TC1s$), або самостійно без нього. У каскадному режимі вихід регулятора підключається до $TC1s.SP$ через змінну $TC1_OUT$, тим самим формуючи завдання веденому регулятору. Виведення з каскаду веденого регулятора проводиться шляхом переключення $TC1s$ в режим слідування (Tracking), подавши на його вхід TR_s змінну $TC1s_OFF$ ($TC1s.TR_s=TC1s_OFF$). Тобто в режимі слідування на вихід $TC1s.OUT$ буде подаватися значення $TC1s.TR_I=TC1_OUT$.

Враховуючи вимоги до безударності переходів та різні режими роботи контуру регулювання, обидва регулятори $TC1$ та $TC1s$ працюють в режимі використання входу $RSPY$ ($en_rcpy=TRUE$). Це значить, що нове значення виходу OUT регулятори будуть розраховувати на базі значення входу $RSPY$.

Для веденого регулятора $TC1s$ на вхід $RSPY$ завжди подається значення яке йде на виконавчий механізм TVI . Тобто, якщо ведений регулятор виведений з каскаду, він все одно буде формувати нове значення виходу внутрішнього регулятора на базі плинного TVI , що забезпечить безударний перехід при повторному вводити його в каскад. Для ведучого регулятора значення $RSPY$ залежить від того, чи включений в контур управління ведений регулятор $TC1s$. Якщо ведений регулятор виведений із каскаду, то $RSPY=TVI$, так як $TC1$ безпосередньо управляє виконавчим механізмом, отже нове розраховане значення базується на значенні TVI . Якщо ведений регулятор в складі контуру, то $RSPY=LIMIT(TC1_OUT)$. Функція обмеження використовується тому, що в режимі ($en_rcpy=TRUE$) обмеження на вихід блоку PI_B не діють.

Дб.6. Реалізація ручного управління. Ручне управління контуру реалізоване через блок ручного управління $HC1$ та відповідних режимів роботи регуляторів $TC1$ та $TC1s$. Значення виходу $HC1.OUT$ зв'язане зі змінною TVI . У автоматичному режимі роботи контуру ($TC1_AUTO=TRUE$) вихід $HC1.OUT=TC1s_OUT$. У ручному режимі вихід $HC1.OUT$ може бути змінений оператором. При цьому, враховуючи що значення TVI повторюється на входах $RSPY$ блоків регулювання забезпечується безударний перехід.

Ведений регулятор завжди знаходиться в автоматичному режимі, так як для ручного режиму передбачений блок *НС1*. Ведучий регулятор, якщо він знаходиться в режимі каскаду з веденим, в ручному режимі повинен фіксувати своє вихідне значення в останньому положенні. Це забезпечується шляхом переключення його в ручний режим. Таким чином ведучий регулятор працює в автоматичному режимі, коли працює в автоматичному режимі весь контур, або коли ведений регулятор виведений з каскаду.

ЛАБОРАТОРНА РОБОТА № 7

Реалізація контурів регулювання з імпульсним вихідним сигналом

Тривалість: 2 акад. години

Мета: ознайомитися з основами побудови контурів регулювання з використанням виконавчих механізмів типу серводвигун, на базі бібліотечних блоків сімейства ControlLIB (регуляторів та блоку SERVO).

Завдання для виконання роботи

Загальна постановка завдання. Необхідно створити проект в UNITY PRO для реалізації поставленої задачі (рис.7.1 та рис.7.2) з використанням FFB бібліотеки ControlLIB. Відлагодження проекту зробити з використанням готового програмного імітатору об'єкту та операторського екрану.

Опис об'єкта 1. Технологічна установка являє собою теплообмінник для підігріву рідкого продукту (рис.7.1) за допомогою гарячої води. Температура продукту на виході вимірюється датчиком температури TT1 (0-100 °C, вхід ПЛК %IW0.1.2) який вмонтований на трубопроводі на відстані кількох метрів від підігрівника, а регулюється витратою гарячої води на виході підігрівника з використанням клапану TV1 і приводом типу МЕО (вихід ПЛК %Q0.3.16 – "більше", %Q0.3.17 – "менше"). Додатково вимірюється також температура води в трубопроводі безпосередньо на виході підігрівника за допомогою датчика TT1a (0-100°C, вхід ПЛК %IW0.1.3).

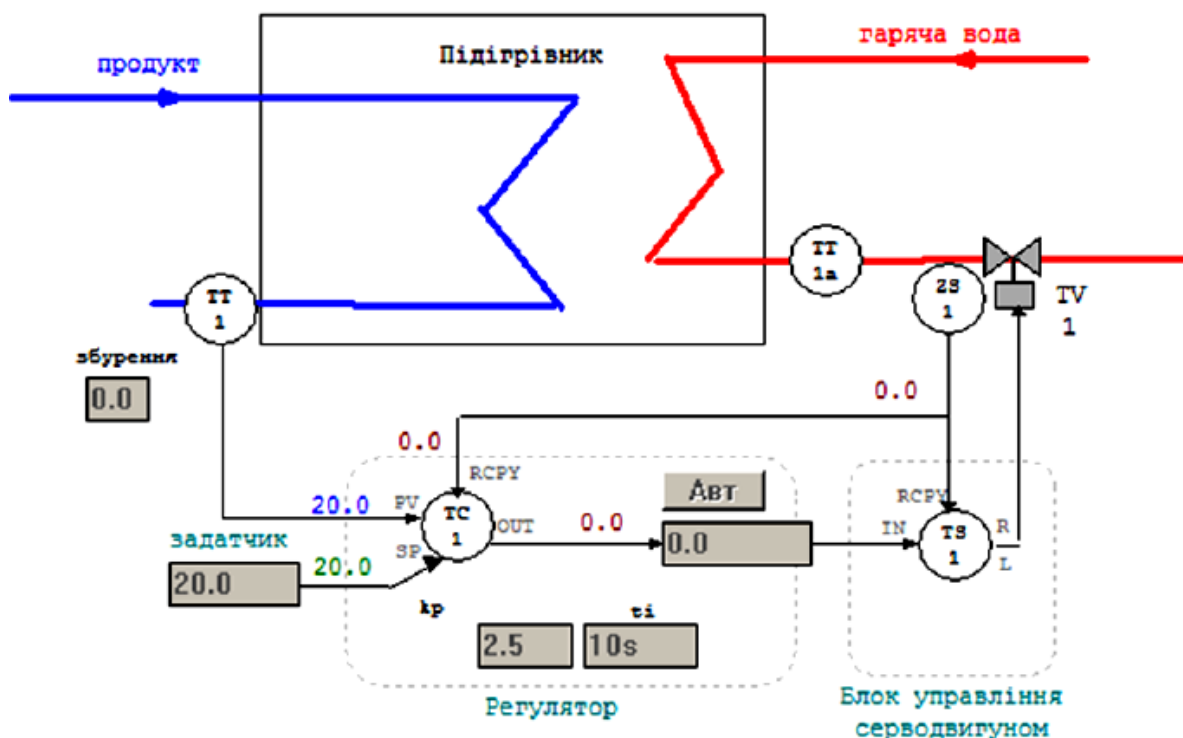


Рис.7.1. Операторський екран для контролю та управління процесом нагрівання в підігрівнику 1

Виконавчий механізм TV1 має показник положення регулюючого органу (0-100%, вхід ПЛК %IW0.1.4). Час повного відкриття клапану – 10с, мінімальний імпульс – 250 мс.

Опис об'єкта 2. Аналогічно як до першого об'єкту, технологічна установка являє собою теплообмінник для підігріву рідкого продукту (рис.7.2) за допомогою гарячої води. Температура продукту на виході вимірюється датчиком температури TT2 (0-100°C, вхід ПЛК %IW0.1.6), а регулюється витратою гарячої води на виході підігрівника з використанням клапану TV2 з приводом типу МЕО (вихід ПЛК %Q0.3.18 – "більше", %Q0.3.19 – "менше"). Додатково вимірюється також температура води в трубопроводі безпосередньо на виході підігрівника за допомогою датчика TT2a (0-100°C, вхід ПЛК %IW0.1.7).

Виконавчий механізм TV2 має датчики кінцевого положення регулюючого органу: "повністю відкритий" - вхід ПЛК %I0.3.0, "повністю закритий" - вхід ПЛК %I0.3.1. Час повного відкриття клапану – 10с, мінімальний імпульс – 250 мс.

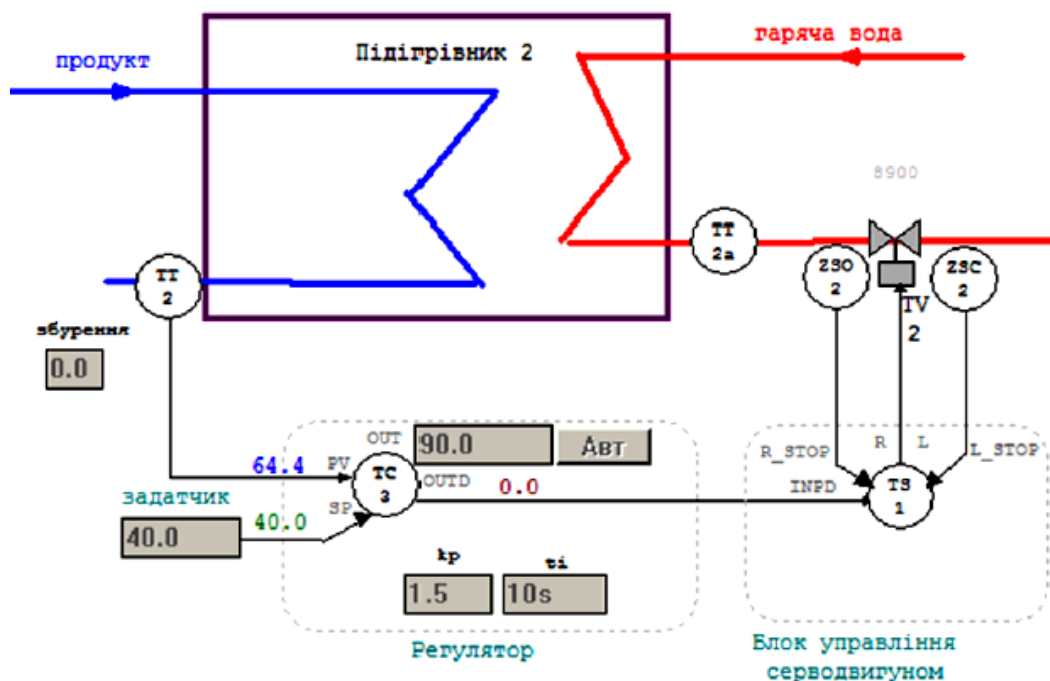


Рис.7.2. Операторський екран для контролю та управління процесом нагрівання в підігрівнику 2

Опис функцій. Система управління повинна забезпечити регулювання температури TT1 та TT2 з підтримкою наступних функцій:

- 1) стабілізація температур на виходах підігрівників на заданому оператором значенні, з використанням ПІ регуляторів та блоків управління серводвигунами;
- 2) можливість настройки коефіцієнтів K_p та T_i з операторського екрану;
- 3) можливість ручного управління виконавчими механізмами TV1 та TV2 з операторського екрану при переключенні в ручний режим з забезпеченням безударності переходу.

Лабораторна установка

Апаратне забезпечення: ПК.

Програмне забезпечення: UNITY PRO V \geq 4.0.

Порядок виконання роботи. Заходи безпеки.

Необхідно дотримуватись стандартних заходів безпеки при роботі з ПК.

Перед виконанням лабораторної роботи ознайомтесь з додатком 7.

1) Запустити на виконання UNITY PRO. Створити новий проект з ПЛК M340.

2) Імпортувати файл апаратної конфігурації:

Project Browser -> контекстне меню Configuration -> Import "Робочий стіл\тека Лаб_ППК\lab7_Servo_HW.XHW".

3) Подивитися конфігурацію обладнання для даної задачі.

4) Імпортувати файл з секцією імітатора об'єкту управління:

Project Browser -> Program-> Tasks -> Mast -> контекстне меню Sections -> Import "Робочий стіл\тека Лаб_ППК\LAB7_Servo_simul.XBD";

5) Імпортувати змінні:

Project Browser -> Variables & FB Instances -> Import "Робочий стіл\тека Лаб_ППК\lab7_Servo_vars.XSY"

Враховуючи що деякі змінні на момент імпорту в проекті вже існують, в діалоговому вікні (рис.6.4) виберіть "Replace All", після чого "Ok".

6) Після імпорту змінних, ознайомтесь з їх призначенням (Рис.7.3)

7) Імпортувати операторські екрани:

Project Browser -> контекстне меню OperatorScreens-> Import-> "Робочий стіл\тека Лаб_ППК\ lab7_Servo_Screen.XCR".

8) Активуйте можливість використання динамічних масивів

Меню Tools->Project Settings-> Variableless->виставити опцію "Allow Dynamic Arrays"

9) Перевірте проект на наявність помилок. Якщо помилки є повторіть пункти 1-9 знову.

10) Створіть DFB тип *aligRCPY* відповідно до рис.Д.7.5. (У редакторі даних Data Editor-> DFB Types).

Після створення інтерфейсу та програми перевірте правильність: меню Build->Analyze

11) Створіть екземпляр з іменем "TY1_RCPY" типу *aligRCPY*.

12) Створіть програму для реалізації обробки входів та виходів. Можна використати варіант секцій "INPUTS" та "OUTPUTS" запропонований в додатку 7.

Увага! Використовуйте вже існуючі (імпортовані) змінні та екземпляри функціональних блоків, інакше операторські екрани не будуть функціонувати!

13) Створіть програму для реалізації управління першим підігрівником. Можна використати варіант секції CTRL1 запропонований в додатку 7.

14) Скомпілюйте проект і завантажте його в симулятор ПЛК.

| Variables DDT Types Function Blocks DFB Types | | | | |
|---|------|--------|--|--|
| Filter <input type="text" value="Name ="/> | | | | |
| Name | Type | Val... | Comment | |
| smInit | BOOL | | Ініціалізація імітаційної моделі | |
| TC1_AUTO | BOOL | | 1 - включити автоматичний режим роботи контуру TC1 | |
| TC2_AUTO | BOOL | | 1 - включити автоматичний режим роботи контуру TC2 | |
| TV1_CLS | BOOL | | 1 - команда на закриття Кл1 | |
| TV1_OPN | BOOL | | 1 - команда на відкриття Кл1 | |
| TV2_CLS | BOOL | | 1 - команда на закриття Кл2 | |
| TV2_OPN | BOOL | | 1 - команда на відкриття Кл2 | |
| ZSC2 | BOOL | | 1 - спрацював датчик повного закриття | |
| ZSO2 | BOOL | | 1 - спрацював датчик повного відкриття | |
| smZ1 | REAL | | збурення для ТТ1 | |
| smZ2 | REAL | | збурення для ТТ2 | |
| TC1_OUT | REAL | | Вихід регулятора TC1 | |
| TC1_SP | REAL | 20.0 | Уставка регулятора TC1 | |
| TC2_OUT | REAL | | Вихід регулятора TC2 | |
| TC2_OUTD | REAL | | Інкрементальний вихід регулятора TC2 | |
| TC2_SP | REAL | 20.0 | Уставка регулятора TC1 | |
| TT1 | REAL | | Т продукту на виході підігрівача 1 | |
| TT1a | REAL | | Т гарячої води на виході підігрівача 1 | |
| TT2 | REAL | | Т продукту на виході підігрівача 2 | |
| TT2a | REAL | | Т гарячої води на виході підігрівача 2 | |
| ZS1 | REAL | | показник положення регулюючого органу Кл1 | |

| Variables DDT Types Function Blocks DFB Types | | | | |
|---|------------|---------|--|--|
| Filter <input type="text" value="Name ="/> | | | | |
| Name | Type | Value | Comment | |
| TC1_PARA | Para_PI_B | | Параметри регулятора | |
| id | UINT | | | |
| pv_inf | REAL | 0.0 | | |
| pv_sup | REAL | 100.0 | | |
| out_inf | REAL | 0.0 | | |
| out_sup | REAL | 100.0 | | |
| rev_dir | BOOL | | | |
| en_rcpy | BOOL | 1 | | |
| kp | REAL | 2.5 | | |
| ti | TIME | t#10s | | |
| dband | REAL | 0.2 | | |
| outbias | REAL | | | |
| TC2_PARA | Para_PI_B | | Параметри регулятора | |
| TS1_PARA | Para_SERVO | | Параметри блоку управління серводвигуном | |
| en_rcpy | BOOL | 1 | | |
| rcpy_rev | BOOL | | | |
| t_motor | TIME | t#10s | | |
| t_mini | TIME | t#250ms | | |
| TS2_PARA | Para_SERVO | | Параметри блоку управління серводвигуном | |
| en_rcpy | BOOL | 0 | | |
| rcpy_rev | BOOL | | | |
| t_motor | TIME | t#10s | | |
| t_mini | TIME | t#250ms | | |

| Variables DDT Types Function Blocks DFB Types | | | | |
|---|-----|-------|---|---|
| Filter <input type="text" value="Name <> sm*"/> | | | | |
| Name | n.. | Type | \ | Comment |
| TC1 | | PI_B | | Регулятор контуру 1 |
| TC2 | | PI_B | | Регулятор контуру 2 |
| TS1 | | SERVO | | Блок управління серводвигуном контуру 1 |
| TS2 | | SERVO | | Блок управління серводвигуном контуру 2 |

Рис.7.3. Змінні проекту

15) Змініть завдання до значення 40%, переключіть регулятор в автоматичний режим. Зверніть увагу на входи та виходи регулятора та блоку управління серводвигуном. Намагайтеся визначити причину такої поведінки. Зверніть увагу на тренди: нижні два показують активність виходів "більше" (зелений) та "менше"(жовтий). Дочекайтесь закінчення перехідного процесу.

16) Переведіть регулятор в ручний режим. Виставте значення на виконавчий механізм рівним 50%. Зверніть увагу на входи та виходи блоку управління серводвигуном. Дочекайтесь поки виконавчий механізм дійде до усталеного значення.

17) Повторіть пункт 16 для уставки 100%, після чого переведіть регулятор в автоматичний режим.

18) Створіть програму для реалізації управління другим підігрівником. Можна використати варіант секції CTRL2 запропонований в додатку 7.

19) Скомпілюйте проект і завантажте його в симулятор ПЛК.

20) Змініть завдання до значення 40%, переключіть регулятор в автоматичний режим. Зверніть увагу на входи та виходи регулятора та блоку управління серводвигуном. Намагайтеся визначити причину такої поведінки. Дочекайтесь закінчення перехідного процесу.

21) Переведіть регулятор в ручний режим. Виставте значення на виконавчий механізм рівним 50%. Зверніть увагу на входи та виходи блоку управління серводвигуном. Дочекайтесь поки виконавчий механізм дійде до усталеного значення.

22) Повторіть пункт 21 для уставки 100%, після чого переведіть регулятор в автоматичний режим.

Аналіз одержаних результатів

Викладачем перевіряється виконання поставленого завдання. Студент повинен пояснити виконання програми та призначення кожного пункту виконаного завдання. *Додатково оцінюється створення власного варіанту рішення задачі або модифікація існуючого.*

Запитання для самоперевірки

1. Розкажіть про призначення блоку управління серводвигунами SERVO.
2. Чим відрізняються принципи управління серводвигунами для першого та другого підігрівника?
3. Яким чином блоки управління серводвигунами можуть визначати положення регулюючого органу?
4. Як контролюється блоком управління серводвигунами досягнення регулюючим органом крайніх положень? Як при цьому веде себе блок?
5. Які особливості обробки серводвигунів в автоматичному та ручному режимах? Розкажіть про призначення входу SEN.
6. Розкажіть про роботу блоку управління серводвигуном в режимі активації RSPY? Як при цьому правильно з'єднувати SERVO та регулятор? Що необхідно передбачити в програмі в ручному режимі?
7. Розкажіть про роботу блоку управління серводвигуном в режимі без RSPY? Як при цьому правильно з'єднувати SERVO та регулятор? Що необхідно передбачити в програмі в ручному режимі?
8. Навіщо задаються в блоці управління серводвигунами параметри t_{motor} та t_{mini} ?

ДОДАТОК 7. Опис варіантів функціональних схем контурів регулювання та програми для їх реалізації в UNITY PRO.

Д7.1. Загальні принципи роботи контуру управління температурою в підігрівнику 1. Функціональна схема контуру регулювання температури продукту показана на операторському екрані (рис.7.1).

Контур включає:

- канали вимірювання ($TT1$, $TT1a$);
- регулятор $TC1$ (ПІ закон), який на основі сформованого завдання (вхід SP) та вимірювального значення ($TT1$ на вхід PV), формує сигнал управління (вихід OUT), що подається на блок управління серводвигуном $TS1$; на вхід $RCPY$ регулятора заводиться сигнал зворотного зв'язку по положенню регулюючого органу $ZS1$; для регулятора налаштовуються Kp та Ti ; регулятор може працювати в автоматичному або ручному режимі (вмикається/вимикається кнопка "АВТ"); в ручному режимі вихід OUT задається безпосередньо оператором;
- блок управління серводвигуном $TS1$, який перетворює числовий сигнал що поступає на вхід IN в діапазоні 0-100% у дискретні сигнали відповідної тривалості типу "більше" (вихід R) та "менше" (вихід L); на вхід $RCPY$ блоку заводиться сигнал зворотного зв'язку по положенню регулюючого органу $ZS1$;
- канал управління, який окрім клапану з виконавчим механізмом $TV1$ включає показник положення регулюючого органу ($ZS1$);

Д7.2. Структура програми. Для реалізації даної задачі використовуються 5-ть секцій (рис.Д7.1): секція "INPUTS" – для обробки вхідних каналів вимірювання; секція "OUTPUTS" – для обробки вихідних каналів управління; секція "CTRL1" та "CTRL2" – для реалізації контурів управління температурою відповідно в підігрівнику 1 та підігрівнику 2. Секція "Simulation" призначена тільки для імітації об'єкта.

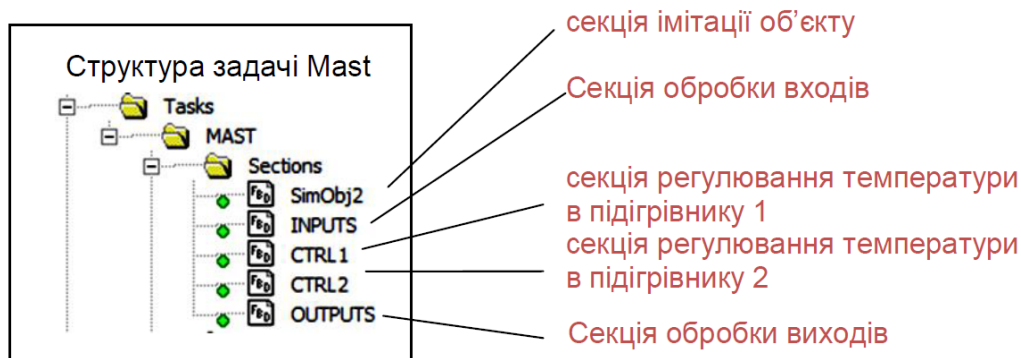


Рис.Д.7.1. Структура задачі MAST

У секції "INPUTS" (рис.Д.7.2) оцифровані значення аналогових входів масштабуються шляхом множення на коефіцієнт (діапазон 0-10000 в 0-100°C). До входів контуру з підігрівачем 1 належить також показник положення $ZS1$ (діапазон 0-10000 в 0-100 %XPO). До входів контуру з підігрівачем 2 належать також датчики кінцевого положення типу "відкрито" - $ZSO2$, та "закрито" - $ZSC2$.

У секції "OUTPUTS" (рис.Д.7.3) для кожного виконавчого механізму на виходи ПЛК подаються сигнали "більше" (TV1_OPN, TV2_OPN) та "менше" (TV1_CLS, TV2_CLS).

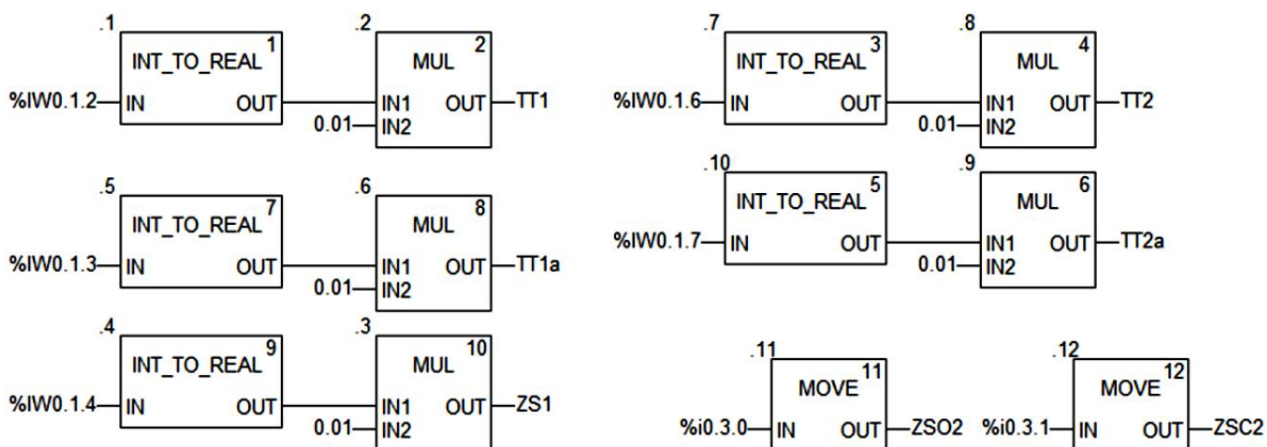


Рис.Д.7.2. Секція обробки входів

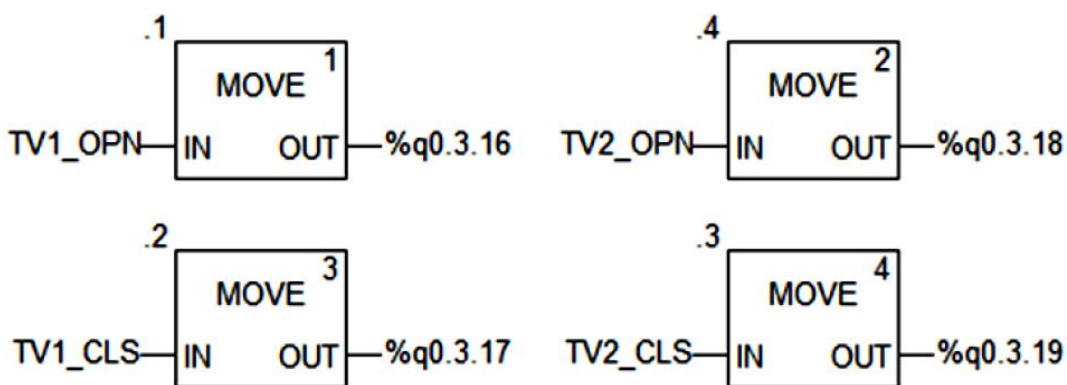


Рис.Д.7.3. Секція обробки виходів

Д7.3.Опис роботи програми реалізації контуру управління температурою в підігрівнику 1. Програма секції "CTRL1" наведена на рис.Д.7.4. Для UNITY PRO V<6.0 вхід/вихід *TC1.OUT* треба розірвати від *TS1* та *SERVO*, а зв'язок реалізувати через змінну *TC1_OUT*.

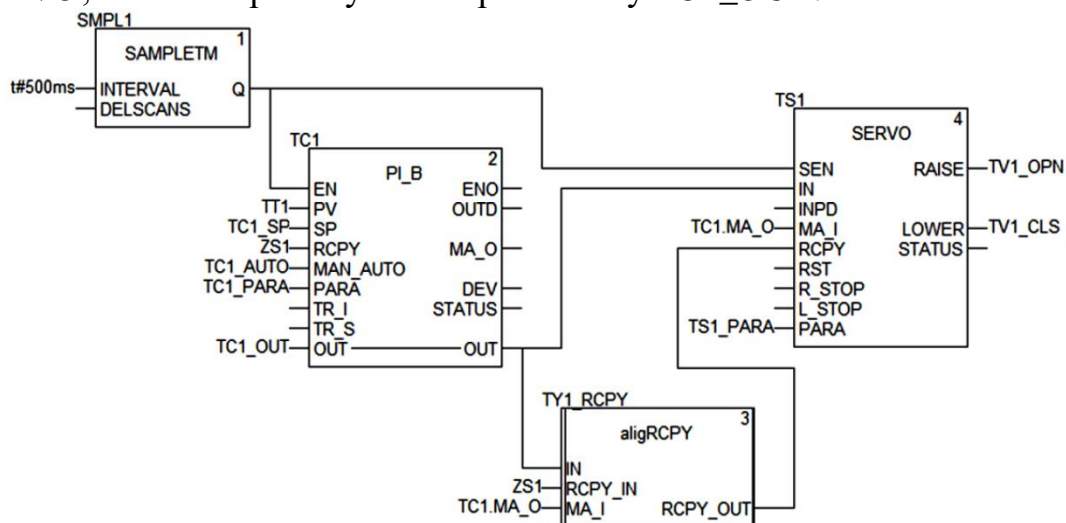


Рис.Д.7.4. Секція CTRL1 35

Оснoву контуру складає ПІ-регулятор $TC1$, який на основі плинної температури $TT1$ та завдання $TC1_SP$ формує на виході OUT (прив'язаний до $TC1_OUT$) числове значення, яке подається на блок управління серводвигуном $TS1$ для формування імпульсів "більше" та "менше". Регулятор $TC1$ працює в режимі використання входу $RCPY$ ($TC1_PARA.en_rcpy=TRUE$). Це значить, що нове значення виходу OUT регулятор буде розраховувати на базі значення входу $RCPY$, на який подається значення покажчика положення $ZS1$.

Блок управління серводвигуном $TS1$ теж працює в режимі використання входу $RCPY$ ($TS1_PARA.en_rcpy=TRUE$). Це значить, що він буде перетворювати значення різниці $IN-RCPY$ у дискретний сигнал $RAISE$ або $LOWER$ відповідної тривалості. Значення параметрів блоку дорівнюють $TS1_PARA.t_motor=t\#10s$ та $TS1_PARA.t_mini=t\#250ms$ відповідно до умов задачі.

Алгоритм роботи блоку $SERVO$ працює таким чином, що у ручному режимі він буде видавати сигнали "більше" та "менше" до тих пір, поки IN та $RCPY$ не будуть рівними. Для того, щоб блок управління серводвигуном в ручному режимі припиняв управління в зоні наближеній до положення $RCPY$, можна створити та використати блок вирівнювання (рис.Д.7.5). Принцип роботи алгоритму заключається в прирівнюванні виходу $RCPY_OUT=IN$ в тому випадку, коли в ручному режимі ($MA_I=TRUE$) вхід IN буде в зоні наближення до $RCPY$. Величина зони наближення визначається параметром $deadb$, збільшення значення якого зменшує кількість рухів однак збільшує похибку позиціонування.

The screenshot displays a software interface with a tree view on the left and a code editor on the right. The tree view shows the structure of the 'aligRCPY' block, including inputs (IN, RCPY_IN, MA_I), outputs (RCPY_OUT), and parameters (deadb). The code editor shows the following program code:

```

if ABS(IN-RCPY_IN) < deadb and NOT MA_I then
    RCPY_OUT := IN;
ELSE
    RCPY_OUT := RCPY_IN;
end_if;

```

Рис.Д.7.5. Структура і програма DFB типу $aligRCPY$.

Блок вирівнювання $TY1_RCPY$ включається в схему між $TC1.OUT$ та $SERVO.RCPY$. Тобто, коли $ZS1$ знаходиться в зоні $TC1_OUT$, на вхід $TS1.RCPY$ буде подаватися значення $TC1_OUT$ а не $ZS1$, що приведе до відключення виходів $LOWER$ та $RAISE$. 36

Д7.4.Опис роботи програми реалізації контуру управління температурою в підігрівнику 2. Програма секції "CTRL2" наведена на

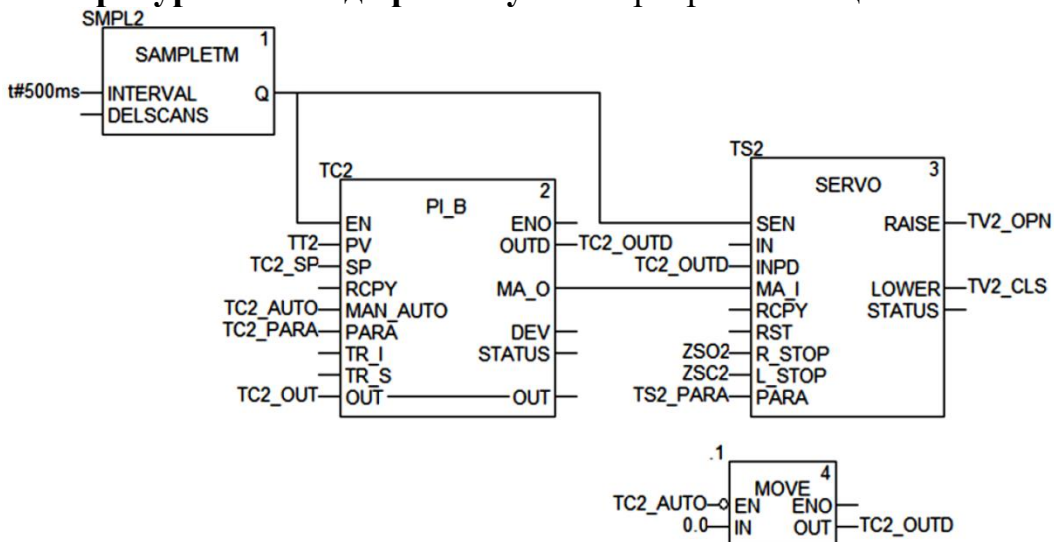


Рис.Д.7.6. Секція CTRL2

Основу контуру складає ПІ-регулятор *TC2*, який на основі плинної температури *TT2* та завдання *TC2_SP* формує на інкрементальному виході *OUTD* (прив'язаний до *TC2_OUTD*) числове значення, яке подається на блок управління серводвигуном *TS2* для формування імпульсів "більше" та "менше". Регулятор *TC2* працює в режимі без використання входу *RCPY* (*TC1_PARA.en_rcpy=FALSE*), саме тому використовується зв'язка інкрементального виходу *TC2.OUTD* та *TS2.INPD*. Це значить, що на кожному циклі блок *TC2* буде розраховувати нове інкрементальне значення виходу *OUTD*, а абсолютне значення *OUT* використовуватися не буде.

Блок управління серводвигуном *TS2* теж працює в режимі без використання входу *RCPY* (*TS1_PARA.en_rcpy=FALSE*). Це значить, що він буде перетворювати значення входу *INPD* у дискретний сигнал *RAISE* або *LOWER* відповідної тривалості. Значення параметрів блоку дорівнюють *TS1_PARA.t_motor=t#10s* та *TS1_PARA.t_mini=t#250ms* відповідно до умов задачі.

Алгоритм роботи блоку *SERVO* працює таким чином, що у ручному режимі він буде видавати сигнали "більше" та "менше" відповідно до значення входу *INPD* на кожному циклі. Враховуючи що змінна *TC2_OUTD* обновлюється з періодичністю виклику *TC2* (500 мс), а блок *TS2* з кожним циклом, протягом 500 мс блок *TS2* буде формувати нові імпульси. Для того щоб уникнути цього ефекту, в ручному режимі (*TC2_AUTO=FALSE*) змінна *TC2_OUTD* обнуляється після обробки контуру.

ЛАБОРАТОРНА РОБОТА № 8

Тривалість: 6 акад. годин

Мета: ознайомитися з технологією виробництва алюмінієвої катанки, алгоритмами керування та програмним забезпеченням промислових контролерів імітаційної моделі АСУТП.

Завдання для виконання роботи

Загальна постановка завдання. Необхідно створити проект в UNITY PRO і розробити програмне забезпечення ПЛК, який здійснює регулювання температури розплаву в плавильній печі, довжини заготовки перед прокаткою, теплового режиму кристалізатора та керування процесом укладання катанки в бунт. За допомогою інтерфейсів оператора виконати дослідження впливу налаштувань регулятора та характеристик об'єкта на якість роботи запрограмованих систем автоматизації.

Технологічна лінія з виробництва алюмінієвої катанки зазвичай складається з пічної дільниці, ливарної машини, лінії транспортування литої заготовки до прокатного стану, прокатного стану і моталок (рис.8.1). За допомогою скіпа завантажувальної машини 1 алюмінієві брикети подаються у газову плавильну піч шахтного типу 2. Після розплавлення метал прямує в одну з двох роздавальних печей 3, де він очищується від шлаку, піддається дегазації 4 та направляється у жолоб ливарного колеса 5, охопленого сталеву стрічкою. Під впливом охолодної води розплавлений алюміній кристалізується у виливниці колеса, що обертається, та у вигляді безперервного зливка подається у прокатний стан за допомогою правильного 6 і переднього тягового 7 пристроїв. У разі незадовільної якості лиття заготовку розрізають гідравлічними ножицями 8 на дрібні мірні шматки до тих пір, поки вона не відповідатиме встановленим вимогам.

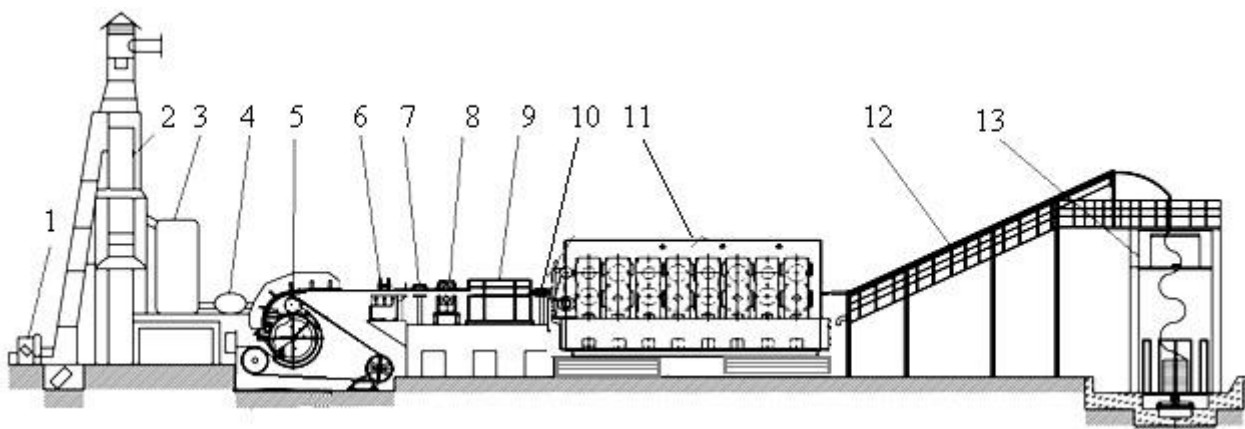


Рис. 8.1. Технологічна лінія неперервного лиття та прокатки

Під час виробництва дроту з алюмінієвих сплавів заготовку перед прокаткою додатково нагрівають індукційним нагрівачем 9 і за допомогою пристрою 10 подають у прокатний стан 11. Після прокатки катанка піддається процесам гартування, охолодження та сушіння 12 і подається на здвоєну моталку кошикового типу 13. Затискний тяговий пристрій разом з пристроєм утворення

кілець укладає катанку в приймальний кошик, після наповнення якого відбувається заміна його на новий.

Відповідно до технології, для одержання високоякісної катанки необхідно підтримувати задану температуру розплаву у шахтній та роздавальних печах, забезпечити стабільні умови кристалізації рідкого металу у ливарному колесі, регулювати довжину алюмінієвої заготовки перед прокаткою, керувати охолодженням катанки під час прокатування у стані та процесом формування бунта у моталці. Оскільки властивості алюмінієвої катанки формуються протягом всього технологічного циклу, від плавильної печі до моталки, виконання усіх перелічених умов має відбуватися якісно. Тому для імітаційного моделювання АСУТП з виробництва алюмінієвої катанки, необхідно визначитися з математичними моделями об'єктів автоматизації, формулами та рівняннями, що описують зв'язок між окремими параметрами, а також алгоритмами керування технологічним процесом, за допомогою яких можна створити системи автоматизації, дослідити вплив тих чи інших параметрів на якість регулювання і налагодити роботу систем автоматизації на бажаний кінцевий результат.

Визначення алгоритмів програмування контролера

Основними вихідними величинами, що характеризують процес неперервного лиття та прокатки є температура рідкого алюмінію в шахтній та роздавальних печах, температура та довжина заготовки на виході з ливарного колеса, температура катанки на виході з прокатного стану та після гартування і охолодження. Для побудови систем автоматичного регулювання динамічні характеристики усіх перелічених параметрів апроксимовані аперіодичними ланками з транспортним запізнюванням:

$$W_{(p)} = \frac{K}{T \cdot p + 1} \cdot e^{-p\tau}$$

де K – коефіцієнт передачі; T – стала часу; τ – транспортне запізнювання.

Регулювання температури в шахтних печах з достатньою якістю відбувається за рахунок коригування витрати спалюваного газу, тому і при імітаційному моделюванні АСУТП використовувався саме цей спосіб.

Регулювання довжини заготовки, зазвичай, здійснюється за показаннями датчиків непрямого вимірювання, що негативно впливає на якість роботи САР. Тому в імітаційній моделі застосована система автоматичного регулювання, яка змінює швидкість прокатного стану на підставі розрахунків поточної довжини заготовки:

$$L = L_{заг.} + \left(V_{Л} \cdot c_1 (T_{вих} - T_{ex}) + V_{Л} - \frac{c_2 n}{\lambda} \right) \cdot \Delta t$$

де L – загальна довжина алюмінієвої заготовки; $L_{заг.} = V_{Л} \cdot \Delta t$ – базова довжина отриманого зливка за цикл лиття; $V_{Л}$ – лінійна швидкість заготовки на виході з ливарного колеса; c_1 – коефіцієнт температурного подовження алюмінію; $T_{вих}$ – температура зливка на виході з ливарного колеса; T_{ex} – температура металу на

вході в ливарне колесо; c_2 – коефіцієнт передачі швидкості прокатного стана; n – швидкість обертання електропривода; λ – коефіцієнт витяжки прокатного стана; Δt – цикл лиття;

Стабілізацію теплового режиму в кристалізаторах, в більшості випадків, реалізують за рахунок коригування витрати води, що подають на охолодження зливка. Проте недоліком таких систем автоматизації є те, що регулювання теплового режиму здійснюють за непрямими показниками – температурою води до та після охолодження. Для підвищення якості регулювання в імітаційній моделі використовується параметр, що напряду характеризує тепловий режим кристалізатора, а саме розраховану величину теплоти, що виноситься із заготовкою з кристалізатора $Q_{заг}$:

$$Q_{заг} = \rho \frac{V}{60} \cdot F \cdot h_n - G_{роз} \cdot (h_n - h_k),$$

де ρ – щільність алюмінієвого сплаву при; V – швидкість лиття; F – площа поперечного перерізу заготовки; $G_{роз} \cdot (h_n - h_k)$ – теплота, що віддається гарячим теплоносієм; $G_{роз}$ **Error! Reference source not found.** – витрата розплаву, що визначається продуктивністю агрегату та швидкістю лиття; $h_n = L + c_n \cdot T_{роз}$ – ентальпія сплаву на початку виливниці; $h_k = L + c_k \cdot T_{заг}$ – ентальпія сплаву у кінці виливниці; L – теплота твердіння розплаву, $T_{роз}$ – початкова та $T_{заг}$ – кінцева температура алюмінієвого сплаву у виливниці; c_n , c_k – теплоємність алюмінієвого сплаву, відповідно, для початкової та кінцевої температур;

Формування бунта готової катанки в імітаційній моделі реалізується шляхом керування кутової швидкості обертання проводки виткоутворювача моталки. При цьому швидкість для i -того витка змінюється ступінчасто, відповідно до виразу:

$$\varpi_i = \frac{V}{R_{\max(\min)} \mp (i-1)d \cdot \alpha},$$

а тривалість керування, яка дорівнює часу формування i -того витка, визначається із співвідношення:

$$T_i = \frac{2\pi}{\varpi_i}.$$

Тут ϖ_i – кутова швидкість обертання проводки; V – швидкість подачі катанки; R_{\max} – радіус максимального витка; R_{\min} – радіус мінімального витка; d – діаметр катанки; $1 \leq i \leq m$, $m = \frac{R_{\max} - R_{\min}}{d \cdot \alpha}$ – кількість витків у шарі, α – крок укладання витків, $1 \leq \alpha \leq \frac{R_{\max} - R_{\min}}{2d}$, « $R_{\max} -$ » – при укладанні шару від периферії до центру, « $R_{\min} +$ » – при укладанні шару від центру до периферії.

Як видно з наведених формул та алгоритмів керування вони відображають зв'язки між окремими технологічними параметрами на кожній ділянці

технологічної лінії, що дозволяє досліджувати на імітаційній моделі вплив кожного з них на якість роботи систем автоматизації.

Імітаційна модель автоматизованої системи управління виробництвом алюмінієвої катанки створена на базі однієї ЕОМ за допомогою пакетів програмування Unity Pro виробництва Schneider Electric і TIA Portal компанії Siemens. Unity Pro використовувався для розробки програм моделювання та управління технологічними процесами, а TIA Portal, у якого є широкі можливості для створення людино-машинного інтерфейсу, застосовано для розробки автоматизованого робочого місця оператора-технолога. АРМ оператора містить три базових вікна візуалізації різних ділянок технологічної лінії з відображенням і анімацією механізмів, пультів управління їх роботою і засобами контролю за технологічними параметрами. На базі програмного забезпечення створеної імітаційної моделі АСУТП виробництвом алюмінієвої катанки розроблено лабораторний практикум, за допомогою якого можна ознайомитися з основними вікнами АРМ оператора АСУТП, кодами програм моделювання і управління технологічними процесами та придбати певний досвід в програмуванні промислових контролерів і дослідженні якості роботи систем автоматизації за допомогою середовища Unity Pro .

Опис функцій. В лабораторній роботі створюється програмне забезпечення контролерів та досліджується якість роботи чотирьох систем автоматизації, що забезпечують:

- 1) завантаження плавильної печі ломом і регулювання температури розплаву на рівні 750 °С;
- 2) регулювання теплового режиму кристалізатора на рівні 1363660Вт;
- 3) регулювання довжини заготовки на рівні 16,8м;
- 4) керування укладанням катанки в бунт із заданою кількістю витків у шарі бунта.

Лабораторна установка

Апаратне забезпечення: ПК.

Програмне забезпечення: UNITY PRO V \geq 9.0.

Порядок виконання роботи. Заходи безпеки

Необхідно дотримуватись стандартних заходів безпеки при роботі з ПК.

Лабораторна робота 8.1. Створення програмного забезпечення імітації роботи САР температури розплаву у плавильній печі

Мета: ознайомитися з операторським вікном візуалізації пічної дільниці АСУТП виробництва алюмінієвої катанки та основами створення системи автоматичної стабілізації температури розплаву в середовищі UNITY PRO.

Загальна постановка завдання. Необхідно створити проект в UNITY PRO для реалізації поставленої задачі. Налаштування системи автоматичного регулювання та дослідження якості її роботи виконати за допомогою готового програмного операторського екрану.

Опис об'єкта 1. У першому вікні, рис. 8.2, імітаційної моделі АСУТП виробництва алюмінієвої катанки зображені шахтна та дві роздавальні печі з

відповідними технологічними лініями зв'язку, а також пульти керування з кнопками вибору режимів роботи, ручного керування та засобами контролю за рівнем і температурою металу в печах. Програмою передбачено завантаження шахтної печі алюмінієвим ломом, увімкнення її в роботу, ручне і автоматичне регулювання температури розплаву на рівні 750 °С, зливання його в одну з роздавальних печей з автоматичною стабілізацією температури (700°С) та подачу розплавленого алюмінію у ливарну машину. Рівень металу в печах відслідковується синім кольором.

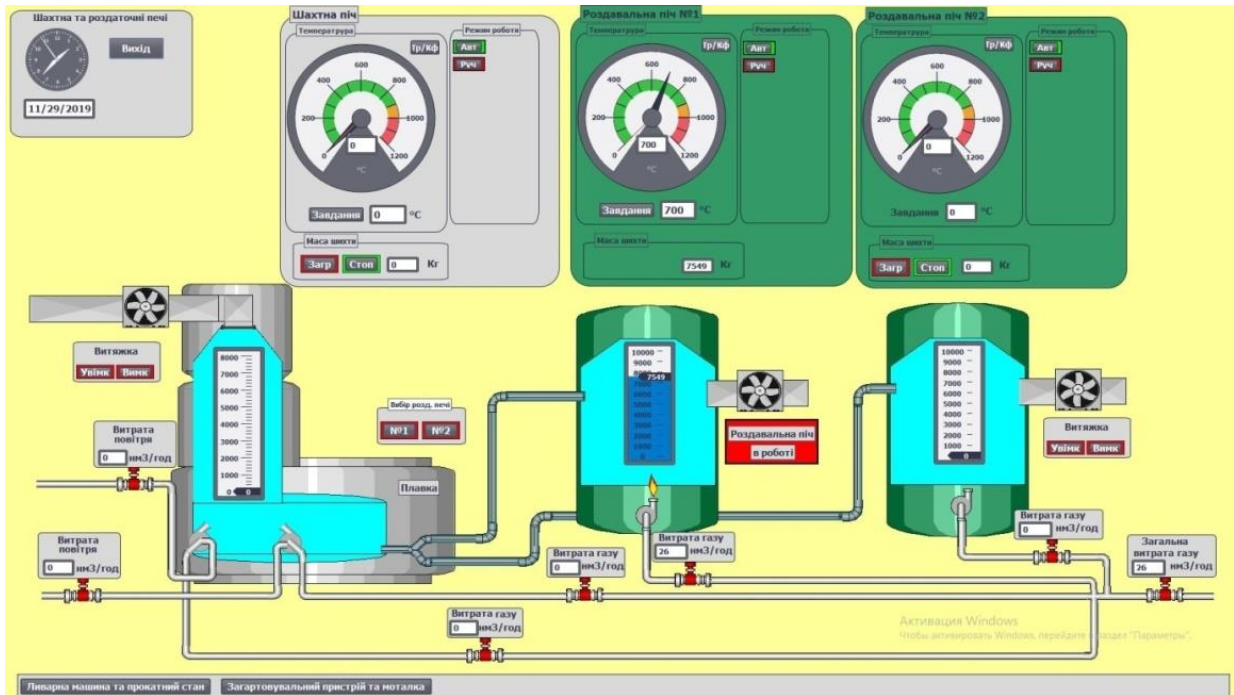


Рисунок 8.2– Вікно візуалізації пічної дільниці

Робочий режим печей попереджається відповідним повідомленням на червоному фоні під вентиляторами, а біля пальників з'являється полум'я. Після злиття розплаву печі зупиняють, попередження про їх робочий стан зникає, а на його місці з'являються кнопки керування роботою вентиляторів, які призначені для провітрювання внутрішнього простору печей.

Перед виконанням лабораторної роботи необхідно ознайомитися з додатком Д8.1.

1) Запустити на виконання UNITY PRO. Створити новий проект з ПЛК M340 на базі центрального процесора BMX P34 2000 02.70 CPU340-20 Modbus.

2) Імпортувати операторський екран:

Project Browser -> контекстне меню Operator Screens -> Import-> "Робочий стіл\тека Лаб_ПрПК_Шахтна\Screen_sahtna.XCR".

3) Активізувати можливість використання динамічних масивів:

Меню Tools->Project Settings-> Variableless->виставити опцію "Allow Dynamic Arrays" .

4) Імпортувати функціональний блок користувача:

Project Browser -> Derived FB Types -> Import "Робочий стіл\тека Лаб_ПрПК\Block_sahtna.XDB".

5) Імпортувати необхідні програмні секції: Project Browser -> Program-> Tasks -> Mast -> контекстне меню Sections -> Import "Робочий стил\тека Лаб_ПрПК\Program_sahtna.XPG".

6) Імпортувати змінні:

Project Browser -> Variables & FB Instances -> Import "Робочий стил\тека Лаб_ПрПК\ Variables_sahtna.XSY".

7) Після імпорту змінних, ознайомитись з їх призначенням (рис.8.3).

| Name | Type | Value | Comment |
|-----------------|---------------|--------|---|
| Add1 | BOOL | | Змінна для кнопки яка збільшує значення виходу регулятора в ручному режимі на 1.0 |
| Add10 | BOOL | | Змінна для кнопки яка збільшує значення уставки регулятора в автоматичному режимі на 10,0 |
| blok_upr | BOOL | | //////// -Допоміжна змінна анімації або блокування-//////// |
| Buf | ARRAY[0..6... | | Буфер обміну для роботи транспортного запізнення |
| Exstr_Stop | BOOL | | Зупинка плавки |
| Exstr_Stop_a... | BOOL | | //////// -Допоміжна змінна анімації або блокування-//////// |
| K_ob | REAL | 2.7 | K-обекта регулювання |
| max_sone_s... | REAL | 100.0 | Максимальна амплітуда збурень |
| min_sone_sa... | REAL | -100.0 | Мінімальна амплітуда збурень |
| Out_reg | REAL | | Вихід регулятора |
| Para_sahtna | Para_PI_B | | Налаштування регулятора |
| Para_skal | Para_SCALI... | | Налаштування блоку масштабування величини для роботи контуру в автоматичному режимі |
| plavka_stop | BOOL | | //////// -Допоміжна змінна анімації або блокування-//////// |
| riv_sahtna | REAL | | Вага сировини/розплаву в шахтній печі |
| Sagnuska | BOOL | | Загрузка сировини для шахтної печі |
| Sahtna_m_a | BOOL | | Пререключення режимів роботи (ручний/автоматичний) |
| Sahtna_upr... | REAL | | //////// -Допоміжна змінна анімації або блокування-//////// |
| Savant_sakin | BOOL | | //////// -Допоміжна змінна анімації або блокування-//////// |
| Savant_saki... | BOOL | | //////// -Допоміжна змінна анімації або блокування-//////// |
| Shum_on_off | BOOL | | Вимк/Увімк збурень |
| sig | BOOL | | //////// -Допоміжна змінна анімації або блокування-//////// |
| sig_Sliv | BOOL | | //////// -Допоміжна змінна анімації або блокування-//////// |
| Sliv | BOOL | | Злив розплаву |
| Stop | BOOL | | Зупинка завантаження |
| Stop_sliv | BOOL | | Зупинказливу з шахтної печі |
| Sub1 | BOOL | | Змінна для кнопки яка зменшує значення виходу регулятора в ручному режимі на 1.0 |
| Sub10 | BOOL | | Змінна для кнопки яка зменшує значення уставки регулятора в автоматичному режимі на 10,0 |
| Suum_t_saht... | REAL | 70.0 | Час формування нового збурення |
| T_ob | REAL | 32.0 | Постійна часу обекта регулювання |
| Tau | REAL | 13.0 | Транспортне запізнення |
| Temp_anim | BOOL | | //////// -Допоміжна змінна анімації або блокування-//////// |
| Temp_sahtna | REAL | | Температура в шахтній печі |
| Temp_sahtn... | REAL | | Задана температура |
| Var1 | BOOL | | //////// -Допоміжна змінна анімації або блокування-//////// |
| Var1_Sliv | BOOL | | //////// -Допоміжна змінна анімації або блокування-//////// |
| Var2 | BOOL | | //////// -Допоміжна змінна анімації або блокування-//////// |
| Var2_Sliv | BOOL | | //////// -Допоміжна змінна анімації або блокування-//////// |
| Var3 | BOOL | | //////// -Допоміжна змінна анімації або блокування-//////// |
| Var4 | BOOL | | //////// -Допоміжна змінна анімації або блокування-//////// |

Рисунок 8.3– Змінні проекту

8) Активізувати екран оператора для дослідження роботи САР температури і ознайомитись з його структурою, рис.8.4:

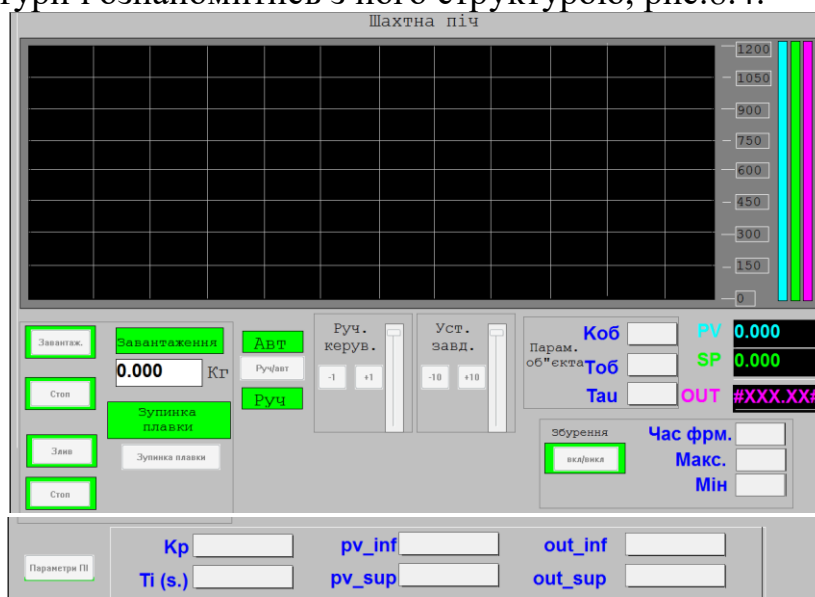


Рисунок 8.4 – Екран оператора.

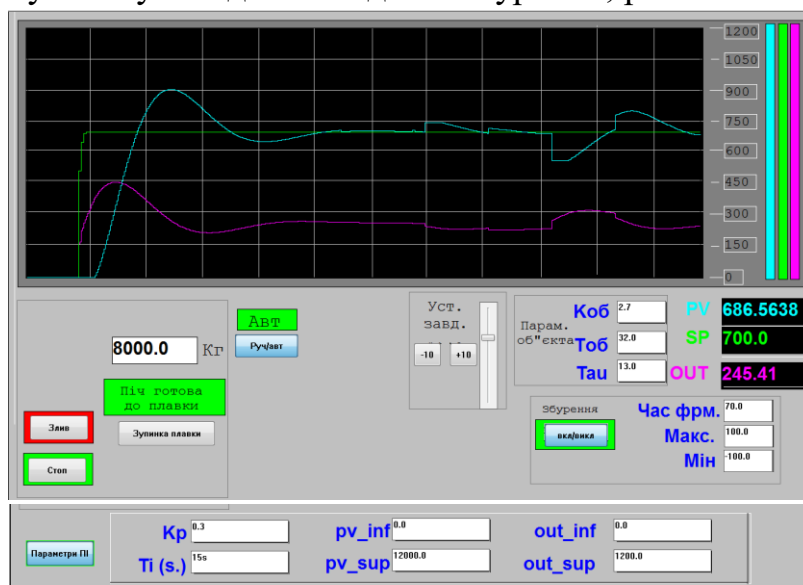
9) Перевірити проект на наявність помилок. Якщо помилки є повторити пункти 1-7 знову.

10) Створити програму секції об'єкта автоматизації «Obj», користуючись варіантом приведеним у додатках Д8.1.5.

Увага! Використовувати вже існуючі (імпортовані) змінні та екземпляри функціональних блоків, інакше операторський екран не буде функціонувати!

11) Скопіювати проект і завантажити його в симулятор ПЛК.

12) За допомогою панелі управління завантажити піч ломом кількістю 8000 кг, а після цього перевести проект в автоматичний режим роботи. Установити завдання температури 700°C і спостерігати за перехідним процесом. Після завершення його увімкнути в дію випадкові збурення, рис.8.6.



Рисунок– 8.6. Екран оператора при автоматичному регулюванні і наявності випадкових збурень.

13) Дослідити якість регулювання температури при наявності збурень, а також при 2-3 інших значеннях параметрів об'єкта і регулятора ($\pm 3-5\%$ від встановлених за замовчуванням) і зробити відповідні висновки про якість регулювання.

Аналіз одержаних результатів

Викладачем перевіряється виконання поставленого завдання. Студент повинен пояснити FBD-програму регулювання температури розплаву, дати характеристику задіяним функціональним блокам і зробити висновки за проведеними дослідженнями.

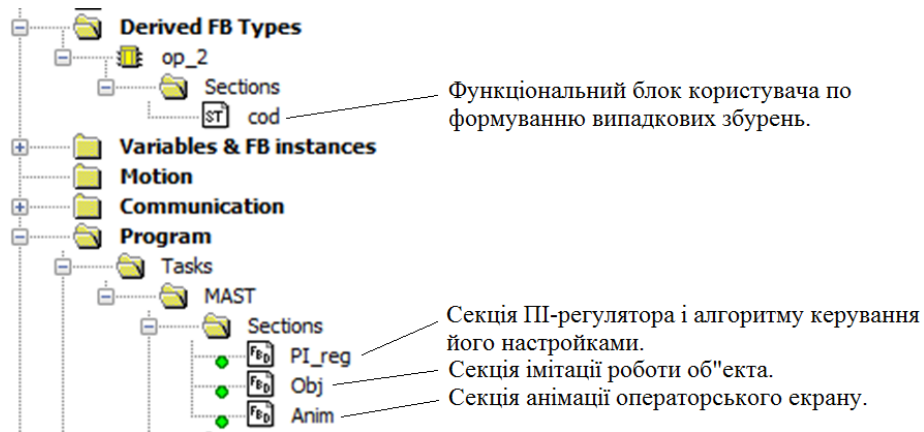
Запитання для самоперевірки

1. Для чого призначена шахтна піч?
2. Який функціональний блок користувача створено у проекті?
3. Чому при програмуванні генерації випадкових збурень для змінної OUT використовується тип *DINT*.
4. Яке завдання реалізоване у секції «PI_reg» за допомогою тригерів R_Trig та арифметичних функцій SUB і ADD?

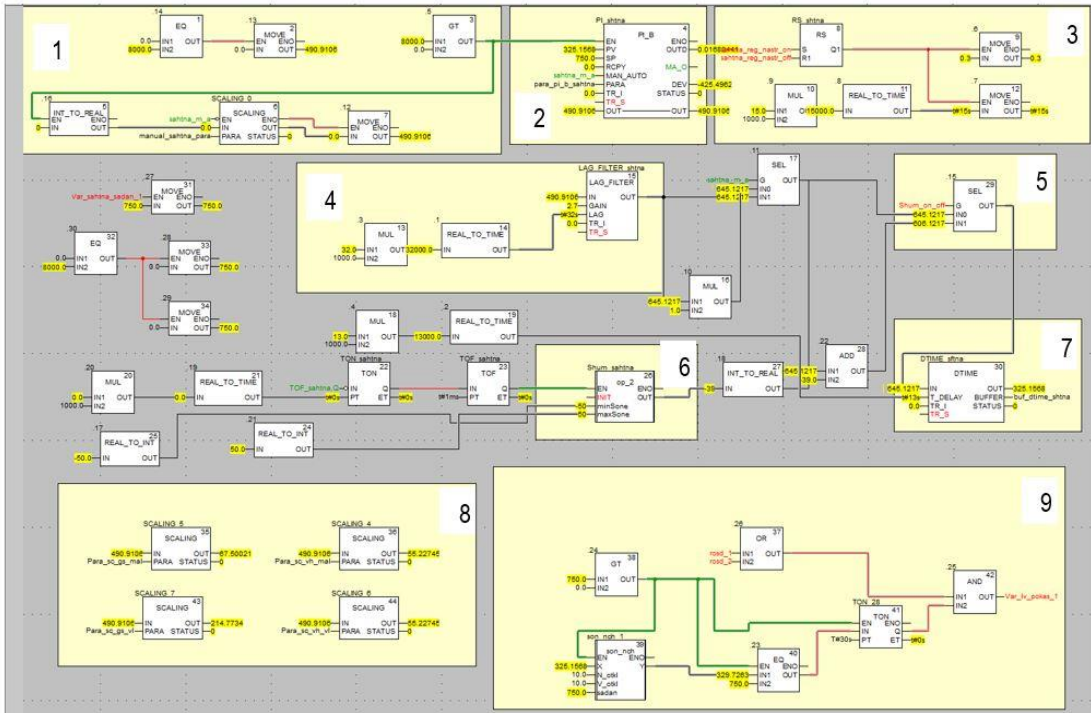
5. Який коментар мають функціональні блоки LAG_FILTER і DTIME, що за допомогою них запрограмовано?
6. Анімація чого запрограмована в секції «Anim» і, як вона працює?

ДОДАТОК 8.1 Опис програмних секцій моделювання роботи САР температури шахтної печі в UNITY PRO.

Д8.1.1. Структура програмного забезпечення. Для реалізації даної задачі використовуються один функціональний блок користувача і 3 секції (рис.Д8.1.1): функціональний блок користувача «op_2» – для формування випадкових збурень в системі автоматизації, секція "PI_reg" – для програмування ПІ-регулятора і алгоритму керування його настройками, секція «Obj» – для імітації роботи об'єкта автоматизації, секція «Anim» – для анімації операторського екрану.



Д8.1.2 Загальний вигляд FBD-програми моделювання роботи системи автоматизації шахтної печі.



Функціональний блок 1 реалізує пуск програми в роботу, блоками 2 і 3 запрограмовані ПІ регулятор і алгоритм керування його настройками. Інерційною ланкою 4 і транспортним запізнюванням 7 представлена шахтна піч. Формування збурень, що випадково з'являються у системі регулювання, реалізовано у блоці 6, а у фрагменті 8 вихідні сигнали регуляторів перетворюється у реальну витрату газу та повітря, що надходять до пальників. Дозвіл на злив розплаву з шахтної печі реалізовано у блоці 9.

Д8.1.3. Функціональний блок користувача «ор_2» по формуванню випадкових збурень.

Для генерації цілих чисел у діапазоні $0-M$ випадкова величина r розраховується за формулою: $r_{i+1} = MOD(kr_i + b, M)$, де r_{i+1} – значення випадкової величини на новому циклі перерахунку, M, k, b – коефіцієнти, r_0 – початкове значення. При ініціалізації програми користувача - $r_i = r_0 = 7$, в іншому випадку r_i береться з попереднього розрахунку, $M = 2^{31} - 1 = 16\#7FFF, 69069, b = 7$, MOD – залишок від ділення першого операнду на другий.

Далі проводиться масштабування сгенерованого числа за лінійним законом і вказаними межами для збурення ($minSone$, $maxSone$). Для коректності операцій множення та ділення вони проводяться з типом $DINT$.

| | | | |
|-------------|---|-------|--|
| op_2 <DFB> | | | |
| <inputs> | | | |
| INIT | 1 | BOOL | |
| minS... | 2 | INT | |
| max... | 3 | INT | |
| <outputs> | | | |
| OUT | 1 | INT | |
| <inputs/... | | | |
| <public> | | | |
| nch | | INT | |
| <private> | | | |
| n | | UDINT | |

```

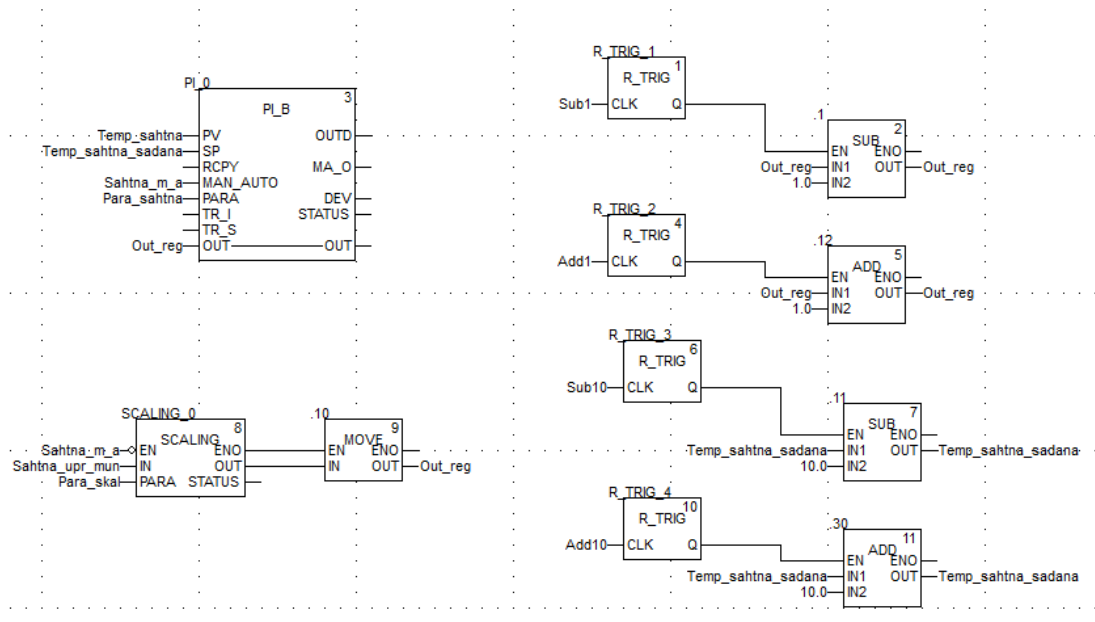
if INIT then
ri:=7;
end_if;

ri:=MOD(69069*ri+7,16#7FFF);

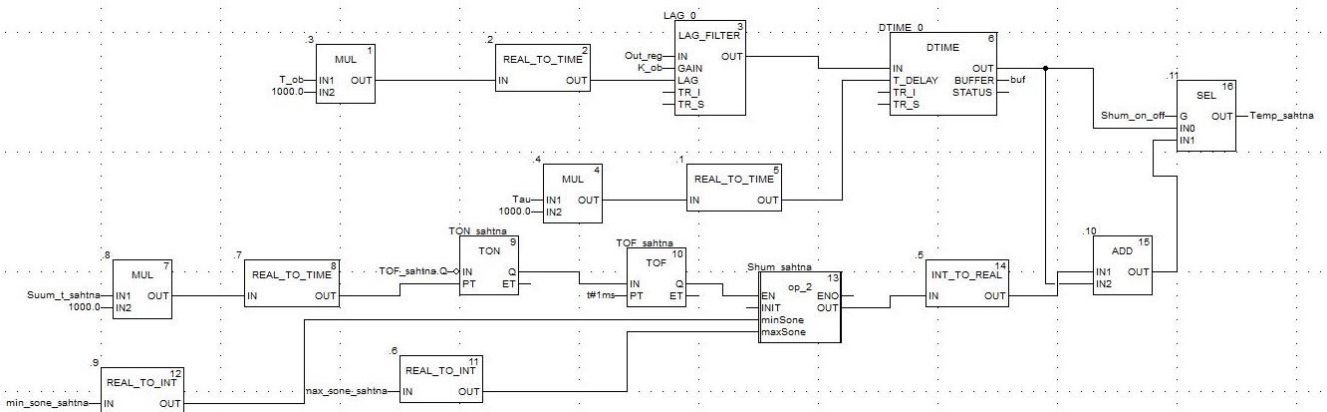
OUT:=DINT_TO_INT(INT_TO_DINT(maxSone-minSone)
*UDINT_TO_DINT(ri)/32767)+minSone+nch;

```

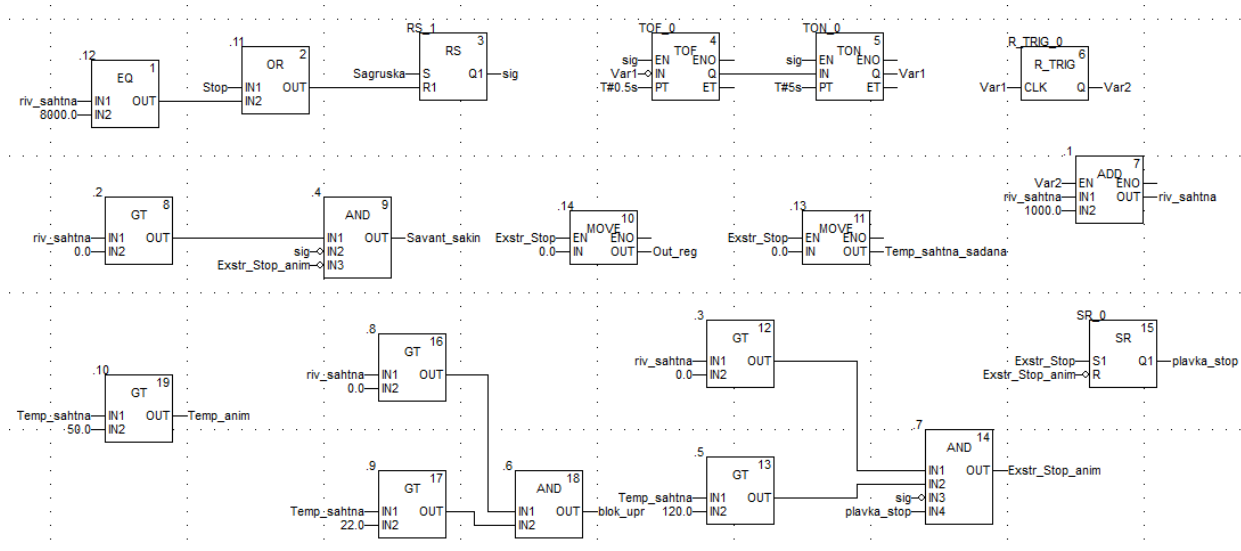
Д8.1.4. Секція реалізації ПІ-регулювання «PI_reg» на базі блоків PI_V і SCALING, а також алгоритму зміни завдання регулятора (+10,-10) та сигналу ручного керування (+1,-1) на екрані оператора.



Д8.1.5. Секція об'єкта автоматизації «Obj» з кодами формування корисного сигналу на базі функціональних блоків LAG_FILTER і DTIME та випадкових збурень з використанням функціонального блока користувача «ор_2».



Д8.1.6. Секція анімації операторського екрану з розрахунком процесу завантаження печі ломом вагою 1000 кг через кожні 5с до повного наповнення 8000кг, а також блокування цього процесу після початку плавлення металу та екстрений останов



Лабораторна робота 8.2. Створення програмного забезпечення імітації роботи САР довжини заготовки перед прокатним станом

Мета: ознайомитися з операторським вікном візуалізації ливарної машини і прокатного стану АСУТП виробництва алюмінієвої катанки та основами створення системи автоматичного регулювання довжини заготовки в середовищі UNITY PRO.

Загальна постановка завдання. Необхідно створити проект в UNITY PRO для реалізації поставленої задачі. Настроювання системи автоматичного регулювання та дослідження якості її роботи виконати за допомогою готового програмного операторського екрану.

Опис об'єкта 2. У другому вікні імітаційної моделі АСУТП виробництва алюмінієвої катанки показані ливарна машина із заготовкою та прокатний стан з пультами керування роботою агрегатів і приладами, що показують значення регульованих параметрів (рис.8.7).

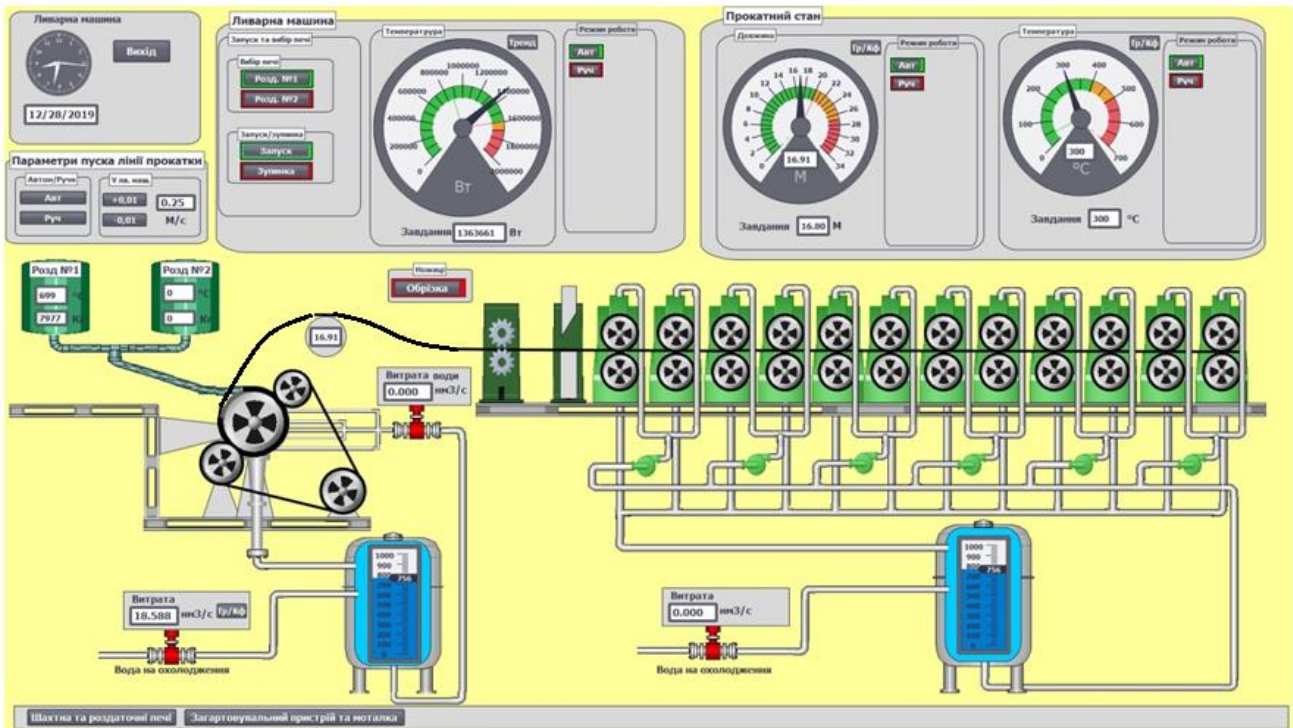


Рисунок 8.7– Вікно візуалізації ливарної машини та прокатного стану

Програмне забезпечення цього вікна дозволяє в реальному масштабі часу здійснити з анімацією вмикання в роботу ливарної машини та прокатного стану, керування їх швидкістю в ручному режимі, відслідковування вторинними приладами значень контрольованих і регульованих параметрів, формування траєкторії руху катанки від кристалізатора до прокатного стану, відрізання гідравлическими ножицями бракованої частини заготовки та автоматичне регулювання теплового режиму кристалізатора, довжини заготовки і температури катанки після прокатного стану.

Перед виконанням лабораторної роботи ознайомтесь з додатком Д8.2.

- 1) Запустити на виконання UNITY PRO. Створити новий проект з ПЛК M340 на базі центрального процесора BMX P34 2000 02.70 CPU340-20 Modbus.
- 2) Імпортувати операторський екран:
Project Browser -> контекстне меню Operator Screens -> Import-> "Робочий стіл\тека Лаб_ПрПК_Довж\Screen_dlinna.XCR".
- 3) Активізувати можливість використання динамічних масивів:
Меню Tools->Project Settings-> Variableless->виставити опцію "Allow Dynamic Arrays" .
- 4) Імпортувати змінні:
Project Browser -> Variables & FB Instances -> Import "Робочий стіл\тека Лаб_ПрПК\ Variables_dlinna.XSY".
- 5) Імпортувати програмну секцію: **Project Browser -> Program-> Tasks -> Mast -> контекстне меню Sections -> Import "Робочий стіл\тека Лаб_ПрПК\Program_dlinna.XPG".**
- 6) Після імпорту змінних, ознайомитись з їх призначенням (рис.8.8).

| Name | Type | Value | Comment |
|------------------|-----------|--------|--|
| add2 | BOOL | | +1 до величини завдання в автоматичному режимі |
| add2_0 | BOOL | | +1 до величини завдання в автоматичному режимі |
| add5 | BOOL | | +5 до виходу регулятора в ручному режимі |
| add5_0 | BOOL | | +5 до виходу регулятора в ручному режимі |
| D_katan | REAL | 8.0 | Діаметр катанки |
| D_katan_0 | REAL | 8.0 | Діаметр катанки |
| Dovgina_sag | REAL | | Довжина заготовки |
| Dovgina_sag... | REAL | | Довжина заготовки |
| N_stana | REAL | | Швидкість прокатного стану |
| out | REAL | | Вихід регулятора |
| out_0 | REAL | | Вихід регулятора |
| out_m_a | REAL | | ////Допоміжна змінна//// |
| out_m_a_0 | REAL | | ////Допоміжна змінна//// |
| para_pi_b_pr... | Para_PI_B | | Параметри регулятора |
| Pr_stan_dov... | REAL | | Початкова величина завдання |
| Pr_stan_dov... | REAL | | Початкова величина завдання |
| Pr_stan_mas... | BOOL | TRUE | Режим роботи руч/авт |
| Q_vh | REAL | 1253.0 | ????????? |
| Q_vh_0 | REAL | 1253.0 | ????????? |
| red_green | BOOL | | ////Допоміжна змінна//// |
| red_green_0 | BOOL | | ////Допоміжна змінна//// |
| Start_Pr_stan | BOOL | | Старт |
| Start_Pr_stan... | BOOL | | Старт |
| Stop_Pr_stan | BOOL | | Стоп |
| Stop_Pr_stan... | BOOL | | Стоп |
| sub1 | BOOL | | -1 до величини завдання в автоматичному режимі |
| sub1_0 | BOOL | | -1 до величини завдання в автоматичному режимі |
| sub5 | BOOL | | -5 до виходу регулятора в ручному режимі |
| sub5_0 | BOOL | | -5 до виходу регулятора в ручному режимі |
| T_sag_Lv_m... | REAL | 450.0 | температура заготовки на виході з прокатного стану |
| T_sag_Lv_m... | REAL | 450.0 | температура заготовки на виході з прокатного стану |
| temp_rod_1 | REAL | 750.0 | Температура розплаву на вході в ливарне колесо |
| temp_rod_2 | REAL | 750.0 | Температура розплаву на вході в ливарне колесо |
| V_lv_maswh | REAL | 0.2 | ????????? |
| V_lv_maswh_0 | REAL | 0.2 | ????????? |
| var_1 | BOOL | | ////Допоміжна змінна//// |
| Var_Nosn_1 | BOOL | | ////Допоміжна змінна//// |
| Var_Nosn_2 | BOOL | | ////Допоміжна змінна//// |

Рисунок 8.8. –Змінні проекту

7) Активізувати екран оператора і ознайомитись з його структурою, рис.8.9:

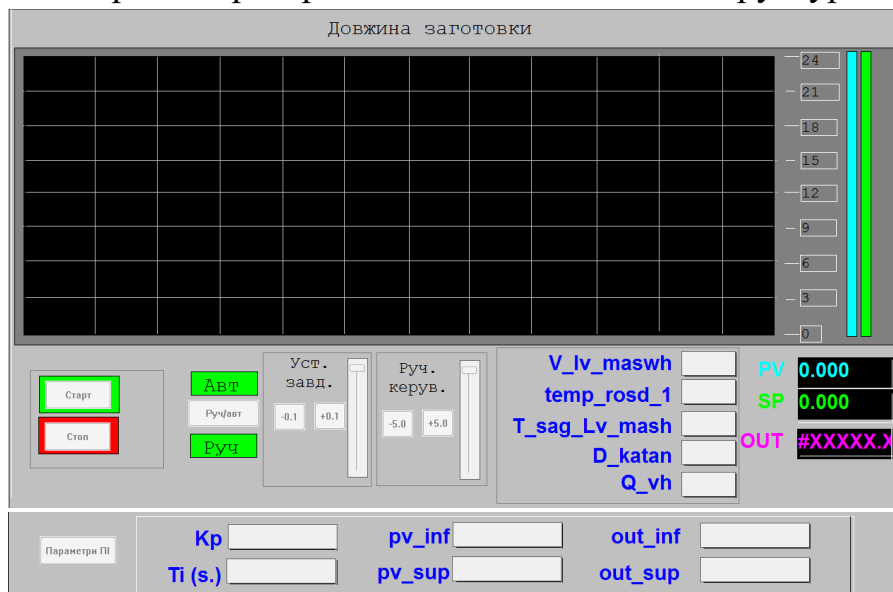


Рисунок 8.9 – Екран оператора

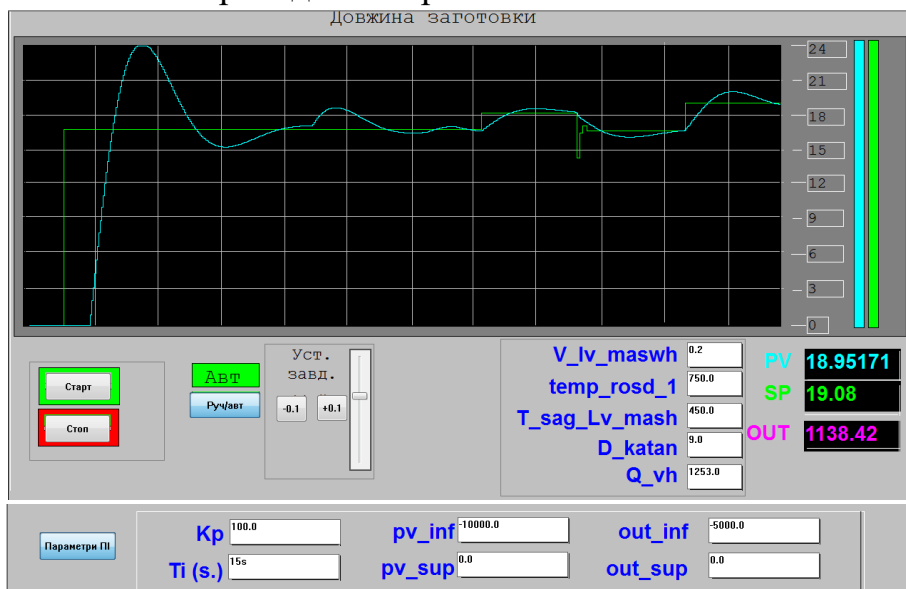
8) Доопрацювати програму секції «dlin_zag» контуром автоматичного регулятора довжини заготовки, користуючись варіантом приведеним у додатках Д8.2.5.

Увага! Використовувати вже існуючі (імпортовані) змінні та екземпляри функціональних блоків, інакше операторський екран не буде функціонувати!

9) Перевірити проект на наявність помилок. Якщо помилки є - виправити і знову перевірити проект.

10) Скомпілювати проект і завантажити його в симулятор ПЛК.

11) За допомогою панелі управління дослідити якість регулювання довжини заготовки при заданому її значенні 16,8 м і параметрах об'єкта автоматизації та настройок регулятора, що встановлені за замовчуванням. Окрім того спостерігати за перехідним процесом системі регулювання при 2-3 інших значеннях параметрів об'єкта і регулятора ($\pm 3-5\%$ від встановлених за замовчуванням) і зробити відповідні висновки про якість регулювання. Екран оператора при дослідженнях САР довжини заготовки приведено на рис.8.10.



Рисунок– 8.10. Екран оператора при автоматичному регулюванні

Аналіз одержаних результатів

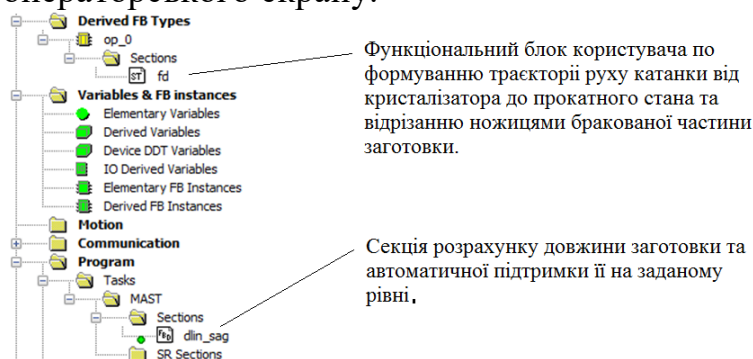
Викладачем перевіряється виконання поставленого завдання. Студент повинен пояснити FBD-програму регулювання довжини заготовки, дати характеристику задіяним функціональним блокам і зробити висновки за проведеними дослідженнями.

Запитання для самоперевірки

1. Яке призначення має і як працює ливарне колесо при виробництві катанки?
2. Яку структуру має програмне забезпечення проекту?
3. Як в програмі контролюється поточна довжина заготовки?
4. Як коментується функціональний блок move?
5. Як впливає на якість регулювання довжини заготовки змінення настройок регулятора та параметрів об'єкта?
- 6.

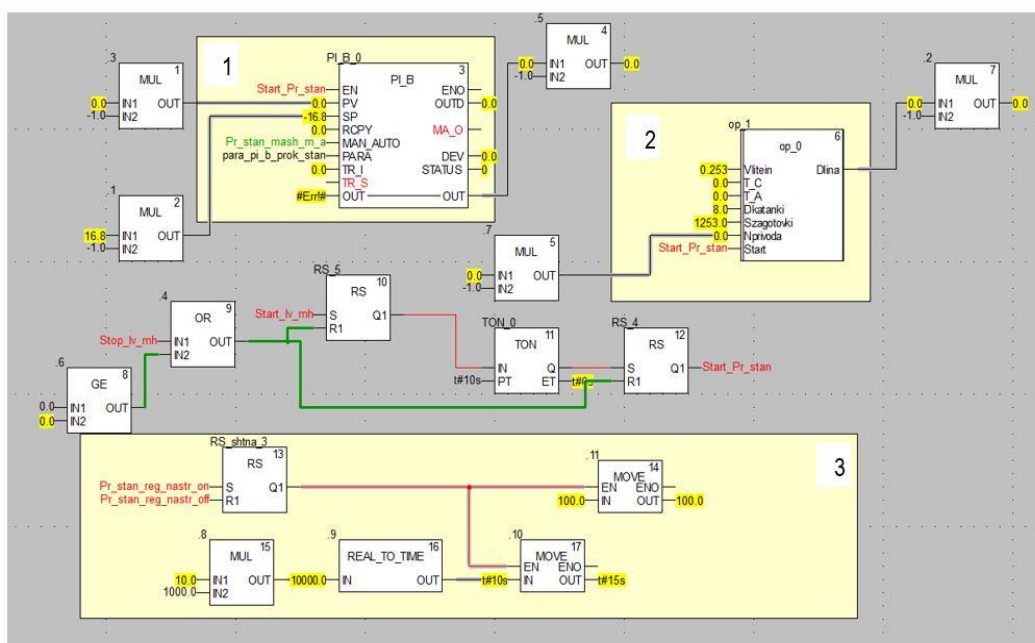
ДОДАТОК 8.2 Опис програмних секцій імітації формування заготовки та автоматичного регулювання її довжини в UNITY PRO.

Д8.2.1 Структура програмного забезпечення проекту. Для реалізації даної задачі використовуються один функціональний блок користувача і одна секція (рис.Д8.2.1): функціональний блок користувача «op_0» – для формування траєкторії руху катанки від кристалізатора та відрізання ножицями бракованої частини, а секція "dlin_sag" – для розрахунку довжини заготовки між кристалізатором і прокатним станом та автоматичної підтримки її на заданому рівні, а також анімації операторського екрану.



Д8.2.2. Загальний вигляд FBD-програми імітації формування заготовки та автоматичного регулювання її довжини.

Програма імітації роботи системи автоматичного регулювання довжини заготовки. Основними у її структурі є функціональні блоки, що формують петлю заготовки та розраховують її довжину між ливарним колесом і прокатним станом 2, моделюють ПІ-регулятор 1 і надають можливість коригувати його настройки 3.



Д8.2.3 Блок розрахунку поточної довжини заготовки «op_0», за формулою $L = L_{заг.} + \left(V_{Л} \cdot c_1 (T_C - T_A) + V_{Л} - \frac{c_2 n}{\lambda} \right) \cdot k \Delta t$, а також керування роботою механічних ножиць, за допомогою яких відрізаються перші 10 метрів

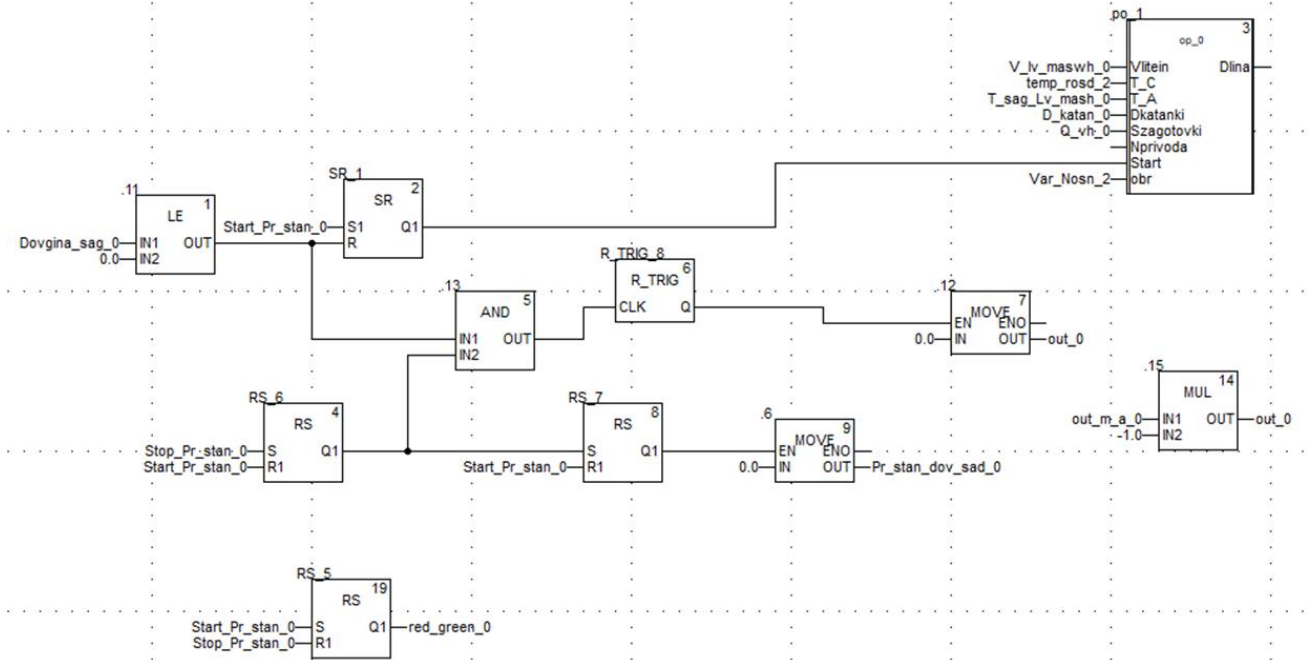
неякісної заготовки на початку роботи ливарного колеса.

| op_0 | | <DFB> | |
|------------|---|-------|---------------|
| inputs> | | | |
| Vliten | 1 | REAL | 0.253 |
| T_C | 2 | REAL | 450.0 |
| T_A | 3 | REAL | 700.0 |
| Dkat... | 4 | REAL | 9.0 |
| Szag... | 5 | REAL | 1253.0 |
| Npri... | 6 | REAL | 1309.7 |
| Start | 7 | BOOL | |
| obr | 8 | BOOL | |
| outputs> | | | |
| Dlina | 1 | REAL | |
| inputs/... | | | |
| public> | | | |
| c1 | | REAL | 2.200000E-005 |
| c2 | | REAL | 0.00338 |
| dIT | | REAL | |
| dIM | | REAL | |

```

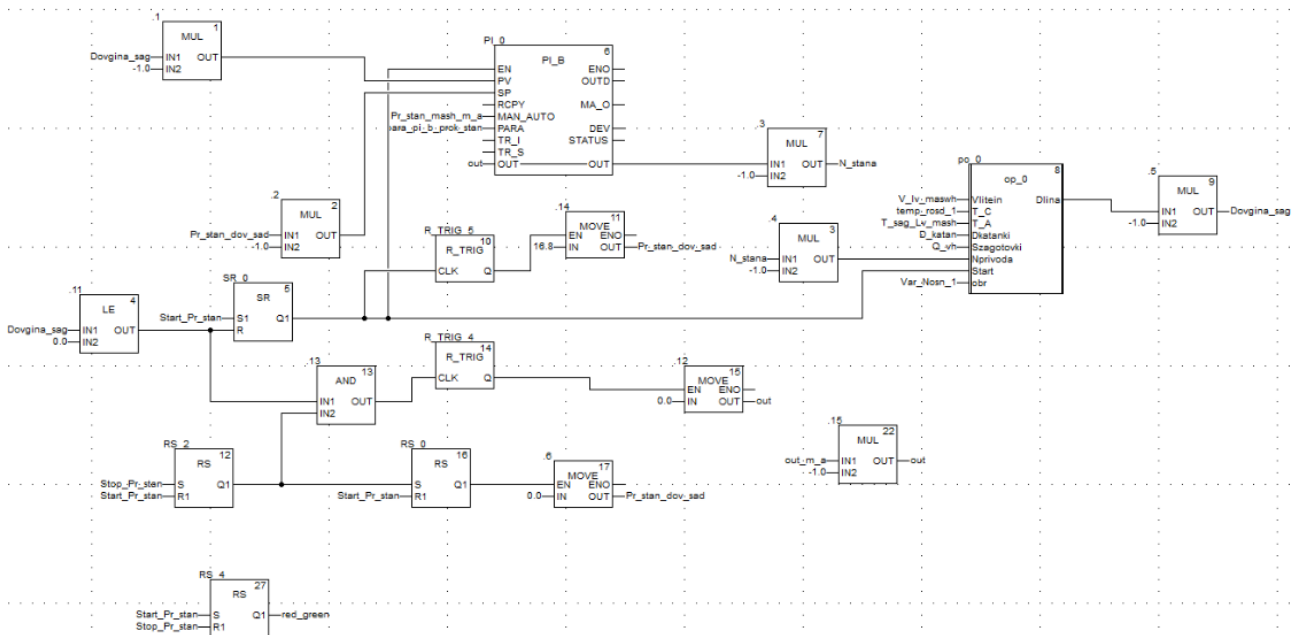
if Start then
dIT:=Vliten*c1*(T_C-T_A);
dIM:=Vliten*c2*(Nprivoda*(-1.0))*3.14*(Dkatanki*Dkatanki)/(4.0*Szagotovki);
Dlina:=Dlina+(dIT+dIM)*0.1;
else
Dlina:=0.0;
end_if;
if obr then
Dlina:=Dlina+10.0;
end_if;
    
```

Д8.2.4. Фрагмент програмного коду секції «dlin_zag», що забезпечує керування процесом розрахунку довжини заготовки в блоці «ор_0»:

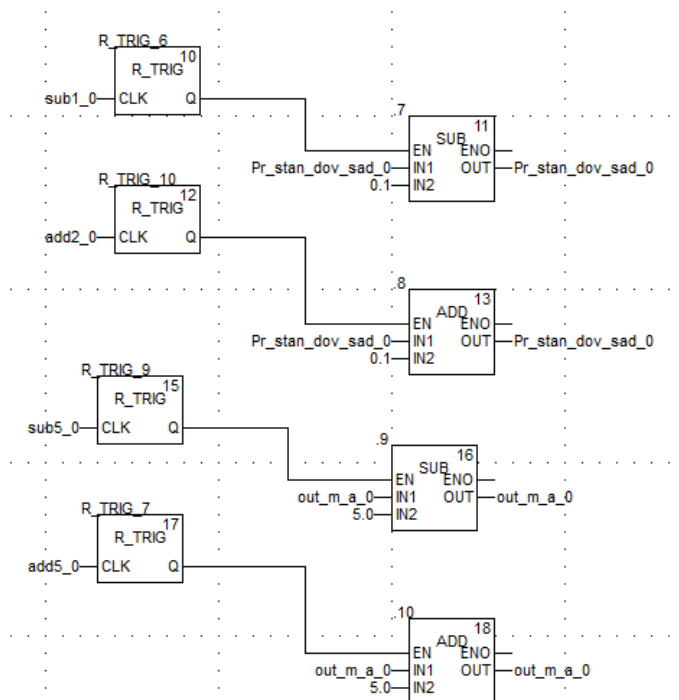


Д8.2.5. Повний програмний код секції імітації автоматичного регулювання довжини заготовки «dlin_zag».

Основна частина:



Додатковий фрагмент програмного коду секції «dlin_zag», що забезпечує на екрані оператора зміни завдання регулятора (+10,-10) та сигналу ручного керування (+1,-1).



Лабораторна робота 8.3. Дослідження роботи імітаційної моделі САР теплового режиму кристалізатора

Мета: ознайомитися з основами створення системи автоматичного регулювання теплового режиму кристалізатора із залученням функціональних блоків користувача в середовищі UNITY PRO та дослідити якість її роботи.

Загальна постановка завдання. Необхідно створити проект в UNITY PRO для реалізації поставленої задачі. Настроювання системи автоматичного регулювання та дослідження якості її роботи виконати за допомогою готового програмного операторського екрану.

Опис об'єкта. Імітаційна модель АСУТП виробництва алюмінієвої катанки з ливарною машиною, заготовкою, прокатним станом, а також пультами керування роботою агрегатів і приладами, що показують значення регульованих параметрів приведена у розділі 8.2 на рис.8.7.

Перед виконанням лабораторної роботи ознайомтесь з додатком Д8.3.

1) Запустити на виконання UNITY PRO. Створити новий проект з ПЛК М340 на базі центрального процесора BMX P34 2000 02.70 CPU340-20 Modbus.

2) Імпортувати операторський екран:

Project Browser -> контекстне меню Operator Screens -> Import-> "Робочий стіл\тека Лаб_ПрПК_Тепл_Реж\Screen_teplo.XCR".

3) Активізувати можливість використання динамічних масивів:

Меню **Tools->Project Settings-> Variableless->виставити опцію "Allow Dynamic Arrays"** .

4) Імпортувати програмну секцію: **Project Browser -> Program-> Tasks -> Mast -> контекстне меню Sections -> Import "Робочий стіл\тека Лаб_ПрПК_Тепл_Реж \Program_teplo.XPG"**.

5) Імпортувати змінні:

Project Browser -> Variables & FB Instances -> Import "Робочий стіл\тека Лаб_ПрПК_Тепл_Реж \ Variables_ teplo.XSY".

6) Після імпорту змінних, ознайомитись з їх призначенням (рис.8.11).

| Name | Type | Value | Comment |
|-------------------------|------------------------|-----------|--|
| dr | ARRAY[0..6000] OF REAL | | Буфер обміу для роботи |
| Kp_reg_Lv_mash | REAL | 0.001 | Кр регулятора |
| lag_f_gain_Lv_mash | REAL | 0.9 | К объекта регулювання |
| lag_f_t_Lv_mash | REAL | 25.0 | Т объекта регулювання |
| Lv_mash_m_a | BOOL | TRUE | Режим роботи (Ручний/Автоматичний) |
| Lv_mash_reg_nastr_off | BOOL | | Вимкнуті налаштування регулятора |
| Lv_mash_reg_nastr_on | BOOL | | Увімкнуті налаштування регулятора |
| Lv_mash_shum_on_off | BOOL | | Увімк/Вимк збурення |
| manual_Lv_mash | INT | | Управління роботою в ручному режимі |
| manual_Lv_mash_para | Para_SCALING | | Маштабування значень для роботи системи в ручному режимі |
| max_sone_Lv_mash | REAL | | Максимальна амплітуда збурення |
| min_sone_Lv_mash | REAL | | Мінімальна амплітуда збурення |
| out_regtemp_Lv_mash | REAL | | Вихід регулятора |
| para_pi_b_Lv_mash | Para_PI_B | | Налаштування регулятора |
| Qsag_Lv_mash | REAL | | Теплота заготовки |
| Qsag_Lv_mash_sadan | REAL | 1363660.5 | Задана кількість теплоти |
| Qsag_Lv_mash_sadan_v... | REAL | | //////////Допоміжна змінна////////// |
| start | BOOL | | Старт |
| stop | BOOL | | Зупинка |
| Suum_Lv_mash_t | REAL | | Час формування збурення |
| t_dtime_Lv_mash | REAL | 14.0 | Транспортне запізнення |
| T_sag_Lv_mash | REAL | | Температура заготовки |
| Ti_reg_Lv_mash | REAL | 25.0 | Ti регулятора |
| V_lv_maswh | REAL | 0.2 | Швидкість ливарної машини |
| Var_1_lv_mash | BOOL | | //////////Допоміжна змінна////////// |
| Var_d | REAL | | //////////Допоміжна змінна////////// |

Рисунок 8.11. –Змінні проекту

7) Активізувати екран оператора і ознайомитись з його структурою, рис.8.12:

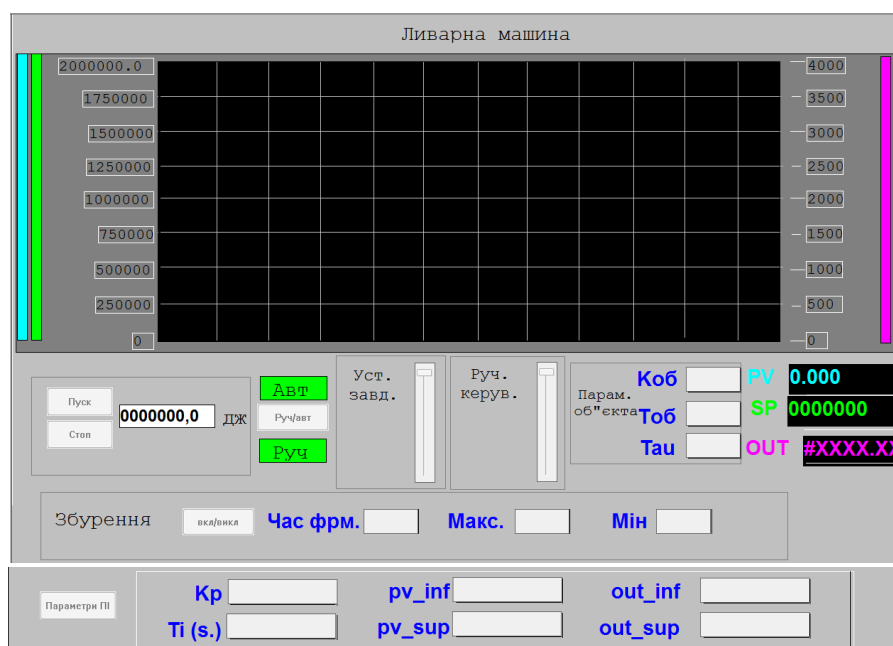


Рисунок 8.12 – Екран оператора

- 8) Створити 4 функціональних блоки користувача (див.додаток Д8.3.3):
- функціональний блок розрахунку кількості теплоти «ор1», що виноситься із заготовкою з кристалізатора (див.додаток Д8.3.4);
 - функціональний блок розрахунку кількості теплоти «ор2», що втрачається з охолодною водою (див.додаток Д8.3.5);
 - функціональний блок генерації випадкових чисел «ор_2», для імітації випадкових збурень(див.додаток Д8.3.6);
 - функціональний блок розрахунку «SmlevelCyl» (див.додаток Д8.3.7);

Увага! Використовувати вже існуючі змінні та екземпляри функціональних блоків, інакше операторський екран не буде функціонувати!

9) Перевірити проект на наявність помилок. Якщо помилки є - виправити і знову перевірити.

10) Скопіювати проект і завантажити його в симулятор ПЛК.

11) За допомогою панелі управління, рис.8.13, дослідити якість регулювання теплового режиму кристалізатора при 2-3 інших параметрах об'єкта і настройках регулятора ($\pm 3-5\%$ від установлених за замовчуванням) та активізації збурень, що випадково з'являються у системі автоматизації.

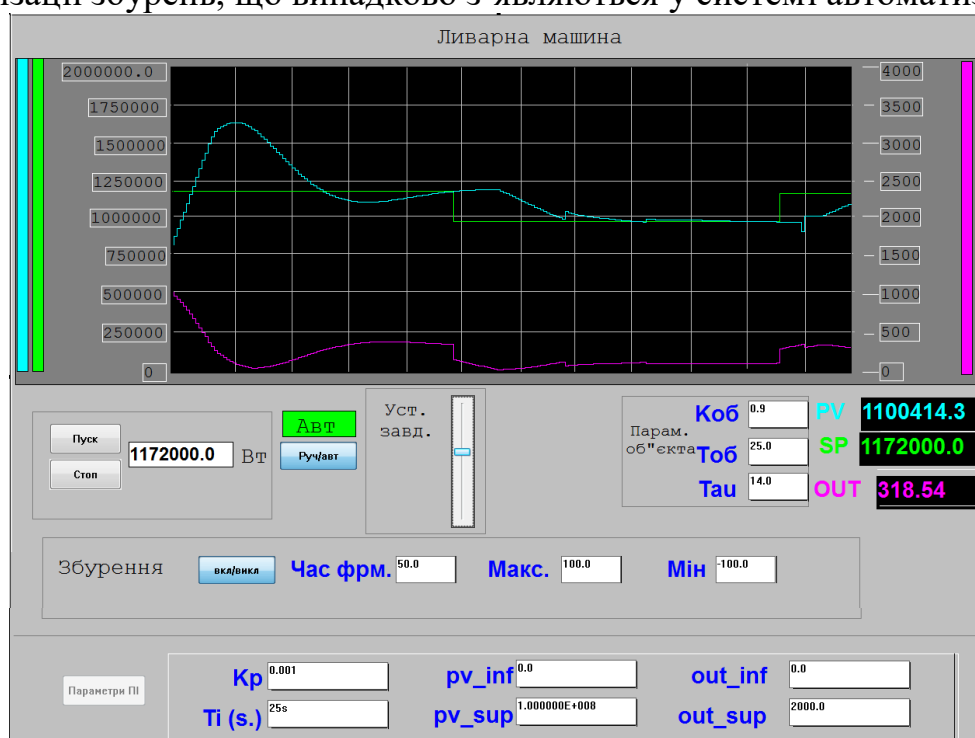


Рисунок 8.13– Екран оператора при автоматичному регулюванні

Аналіз одержаних результатів

Викладачем перевіряється виконання поставленого завдання. Студент повинен пояснити FBD-програму регулювання теплового режиму кристалізатора, дати характеристику задіяним функціональним блокам і зробити висновки за проведеними дослідженнями.

Запитання для самоперевірки

1. Чому ливарне колесо також має назву кристалізатор?
2. Чим пояснюється необхідність створення чотирьох функціональних блоків користувача?
3. Як створюється функціональний блок користувача в UNITY PRO?
4. Як розраховується кількість теплоти, що виноситься із заготовкою з кристалізатора.
5. Прокоментуйте секцію "lv_mash".
6. Як працює блок налаштувань регулятора?

ДОДАТОК 8.3 Опис програмних секцій імітаційної моделі автоматичного регулювання теплового режиму кристалізатора в UNITY PRO.

Д8.3.1 Структура програмного забезпечення проекту. Для реалізації даної задачі використовуються чотири функціональних блока користувача і одна секція (рис.Д8.2.1): функціональний блок користувача «op1» – для розрахунку кількості теплоти, що виноситься із заготовкою з кристалізатора, функціональний блок користувача «op2»– для розрахунку кількості теплоти, що втрачається з охолодною водою, функціональний блок користувача «op_2» – для генерації шумів, що накладаються на корисний сигнал, функціональний блок користувача «SmlevelCyl», а секція "lv_mash" – для імітації системи автоматичного регулювання теплового режиму кристалізатора та анімації операторського екрану.

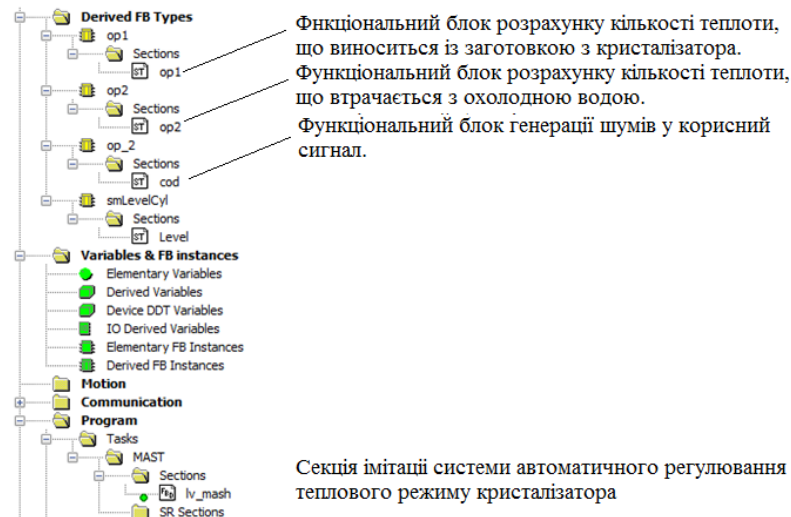
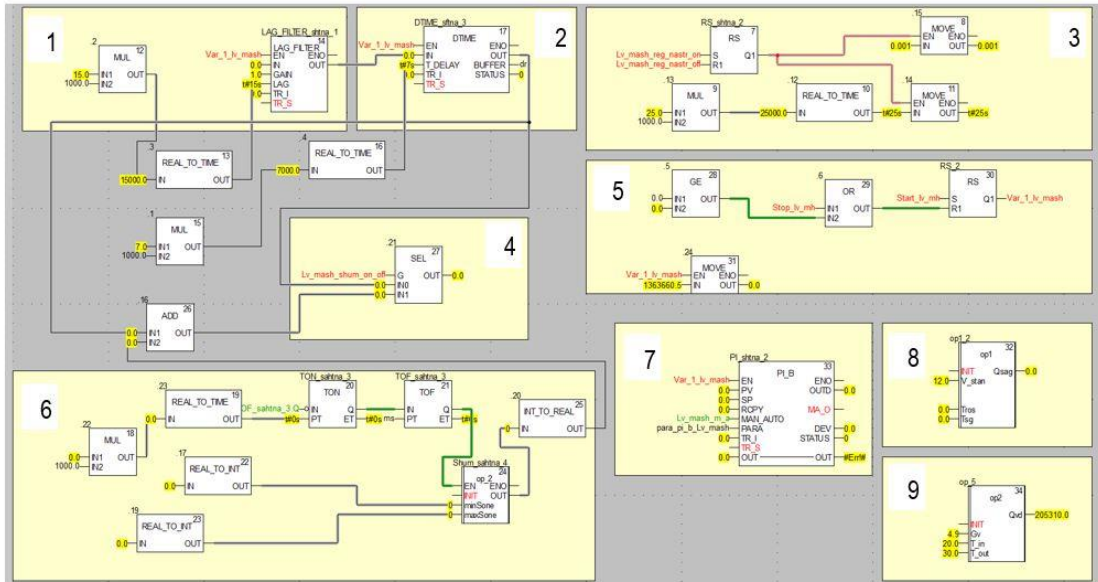


Рисунок Д.8.3.1 – Структура програмного забезпечення проекту.

Д8.3.2. Загальний вигляд FBD-програми імітації роботи системи автоматичного регулювання теплового режиму кристалізатора.

Програма регулювання теплового режиму кристалізатора за розрахованими значеннями кількості теплоти, що виноситься із заготовкою з кристалізатора, і кількості теплоти, що втрачається з охолодною водою:



Програма включає 9 основних фрагментів: моделі кристалізатора 1 і транспортного запізнювання 2, блоки формування 6 і увімкнення 4 збурень, блоки розрахунку кількості теплоти, що виноситься із заготовкою з кристалізатора 8, і кількості теплоти, що втрачається з охолодною водою 9, та блоки ПІ-регулятора 37 і зміни його настрійок 3.

Д8.3.3 Створення функціональних блоків користувача (DFB).

Для використання функціональних блоків користувача в Unity Pro, спочатку у розділі проекту Derived FB Types розробляється DFB-тип, на основі якого потім створюються екземпляри. Екземпляри функціональних блоків користувача створюються та використовуються аналогічно EFB.

Процес створення DFB типу схожий на створення структурного DDT. При цьому вказується ім'я типу, описується інтерфейс та внутрішня структура типу (рис.Д8.3.2). Інтерфейс блока (формальні параметри) описується входами (Inputs) виходами (Outputs) та входами і виходами (Inputs/Outputs), використання яких у програмі аналогічно як для елементарних функціональних блоків. На рис.Д8.3.2 показано створення DFB-типу, для якого описані 4 вхідні формальні параметри (Start, LS1, LS2, ManPump) та один вихідний (M1). Порядок розміщення інтерфейсних параметрів вказується у властивості «по».

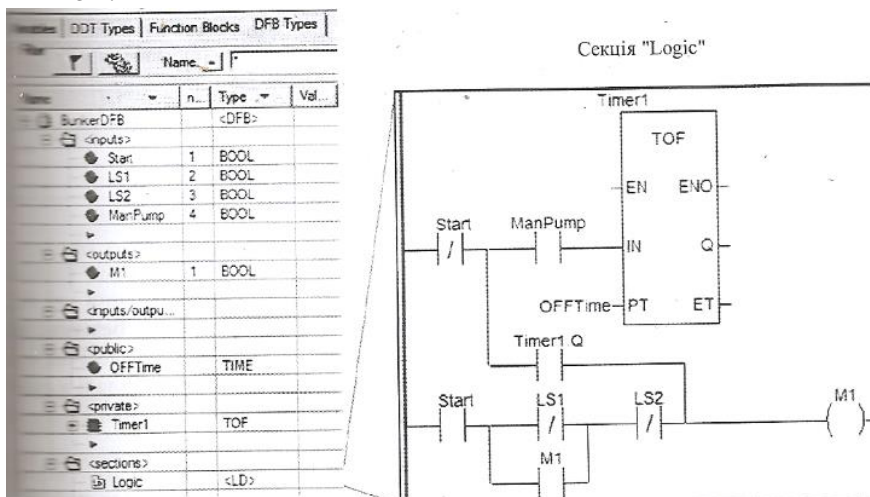


Рисунок рис.Д8.3.2 – Створення типу функціонального блока користувача DFB.

Програма функціонального блока описуються в розділі Sectins. Секції можуть бути створені з використанням мов LD, ST, IL або FBD. У секціях можна використовувати тільки входи, виходи, входи/виходи, локальні змінні або екземпляри та глобальні змінні функціонального блока, бібліотечні функції і процедури. Дозволяється також використовувати системні біти та слова. На рис. Д8.3.2 функціональний блок вміщує тільки одну програмну секцію – «Logic».

Д8.3.4 Функціональний блок розрахунку кількості теплоти «op1», що виноситься із заготовкою з кристалізатора за формулою

$$Q_{заг} = \rho \frac{V}{60} \cdot F \cdot h_{роз} - G_{роз} \cdot (h_{роз} - h_{заг}) :$$

The screenshot shows the configuration table for block 'op1' and its logic code. The configuration table lists inputs and outputs with their data types and values.

| op1 | | <DFB> | |
|---------|---|-------|--------|
| Inputs | | | |
| INIT | 1 | BOOL | |
| V_stan | 2 | REAL | |
| Tros | 5 | REAL | |
| Tsg | 6 | REAL | |
| Outputs | | | |
| Qsaq | 1 | REAL | |
| Public | | | |
| C_sg | | REAL | 1027.0 |
| L | | REAL | 362300 |
| C_met | | REAL | 1250.0 |
| ρ_met | | REAL | 2700.0 |
| F | | REAL | 0.0023 |
| G_ros | | REAL | 1.26 |

```

if V_stan>0.0 and Tros>0.0 and Tsg>10.0 and init then
  Qsaq:=((p_met*V_stan)/60.0)*F*(L+C_met*(Tros+273.0))
  -(G_ros*((L+C_met*(Tros+273.0))-(L+C_sg*(Tsg+273.0))));
else
  Qsaq:=0.0;
end_if;
  
```

Д8.3.5 Функціональний блок розрахунку кількості теплоти «op2», що втрачається з охолодною водою, за формулою $Q_{вод} = c_e \cdot G_e \cdot \Delta T_e$:

The screenshot shows the configuration table for block 'op2' and its logic code. The configuration table lists inputs and outputs with their data types and values.

| op2 | | <DFB> | |
|---------|---|-------|--------|
| Inputs | | | |
| INIT | 2 | BOOL | |
| Gv | 3 | REAL | |
| T_in | 4 | REAL | |
| T_out | 5 | REAL | |
| Outputs | | | |
| Qvd | 1 | REAL | |
| Public | | | |
| C_vd | | REAL | 4190.0 |

```

if INIT then
  Qvd:=0.0;
end_if;
if Gv>0.0 and T_in>0.0 and T_out>0.0 then
  Qvd:=C_vd*Gv*abs(T_in-T_out);
end_if;
  
```

Д8.3.6 Функціональний блок генерації випадкових чисел «op_2», для імітації випадкових збурень.

Для генерації цілих чисел у діапазоні 0-M випадкова величина r розраховується за формулою: $r_{i+1} = MOD(kr_i + b, M)$,

де r_{i+1} – значення випадкової величини на новому циклі перерахунку, M, k, b – коефіцієнти, r_0 – початкове значення. При ініціалізації програми користувача - $r_i = r_0 = 7$, в іншому випадку r_i береться з попереднього розрахунку, $M = 2^{31} - 1 = 16\#7FFF, 69069, b = 7$, MOD – залишок від ділення першого операнда на другий.

Далі проводиться масштабування згенерованого числа за лінійним законом і вказаними межами для збурення ($minSone, maxSone$). Для коректності операцій множення та ділення вони проводяться з типом $DINT$.

| | | |
|---------------------------------|---|-------|
| op_2 | | <DFB> |
| <inputs></inputs> | | |
| INIT | 1 | BOOL |
| minS... | 2 | INT |
| max... | 3 | INT |
| <inputs ...<="" td=""></inputs> | | |
| <outputs></outputs> | | |
| OUT | 1 | INT |
| <inputs ...<="" td=""></inputs> | | |
| <public></public> | | |
| nch | | INT |
| <private></private> | | |
| n | | UDINT |

```

if INIT then
  ri:=7;
end_if;

ri:=MOD(69069*ri+7,16#7FFF);

OUT:=DINT_TO_INT(INT_TO_DINT(maxSone-minSone)
*UDINT_TO_DINT(ri)/32767)+minSone+nch;

```

Д8.3.7 Функціональний блок розрахунку рівня «smLevelCyl»

| | | |
|---------------------------------|---|--------------|
| smLevelCyl | | <DFB> |
| <inputs></inputs> | | |
| INIT | 1 | BOOL |
| Fin1 | 2 | REAL |
| Fin2 | 3 | REAL |
| Fout | 4 | REAL |
| <inputs ...<="" td=""></inputs> | | |
| <outputs></outputs> | | |
| L | 1 | REAL |
| <inputs ...<="" td=""></inputs> | | |
| <public></public> | | |
| d_t | | REAL 0.1 |
| V0 | | REAL 0.0 |
| Vmax | | REAL 10000.0 |
| S | | REAL 10.0 |
| V | | REAL 8000.0 |

```

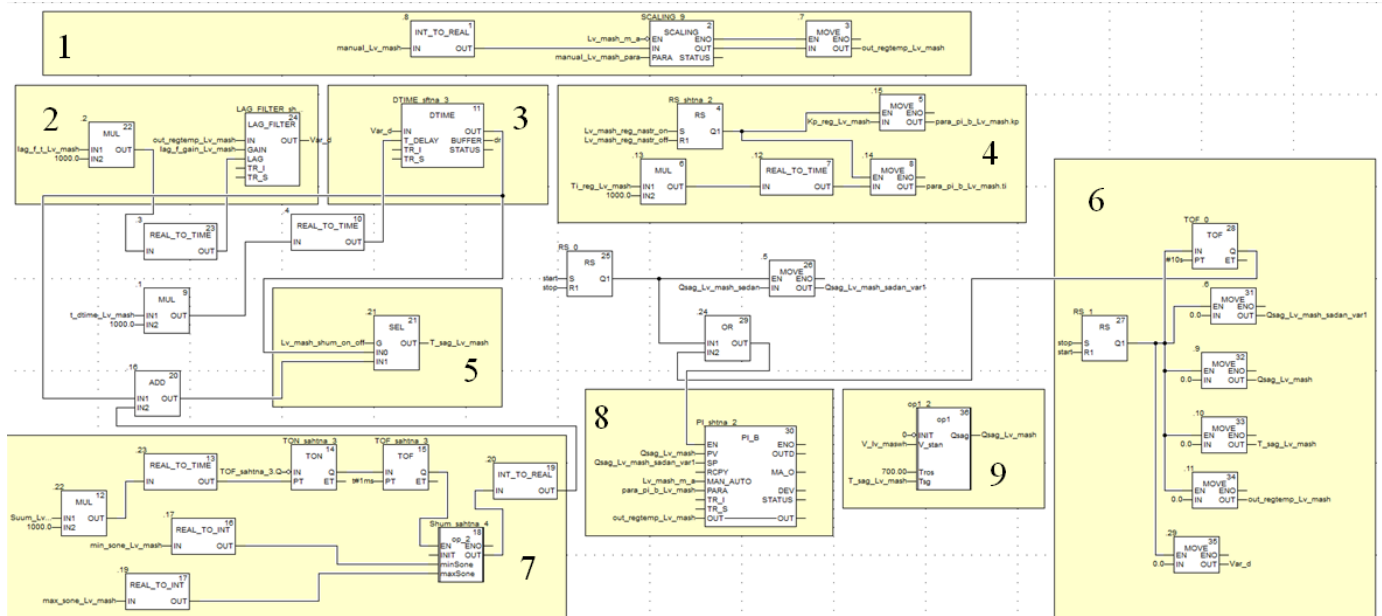
if INIT then V:=V0;
else
  V:=V+d_t*(Fin1+Fin2-Fout);

  if V<0.0 then V:=0.0; end_if;
  if V>Vmax then V:=Vmax; end_if;

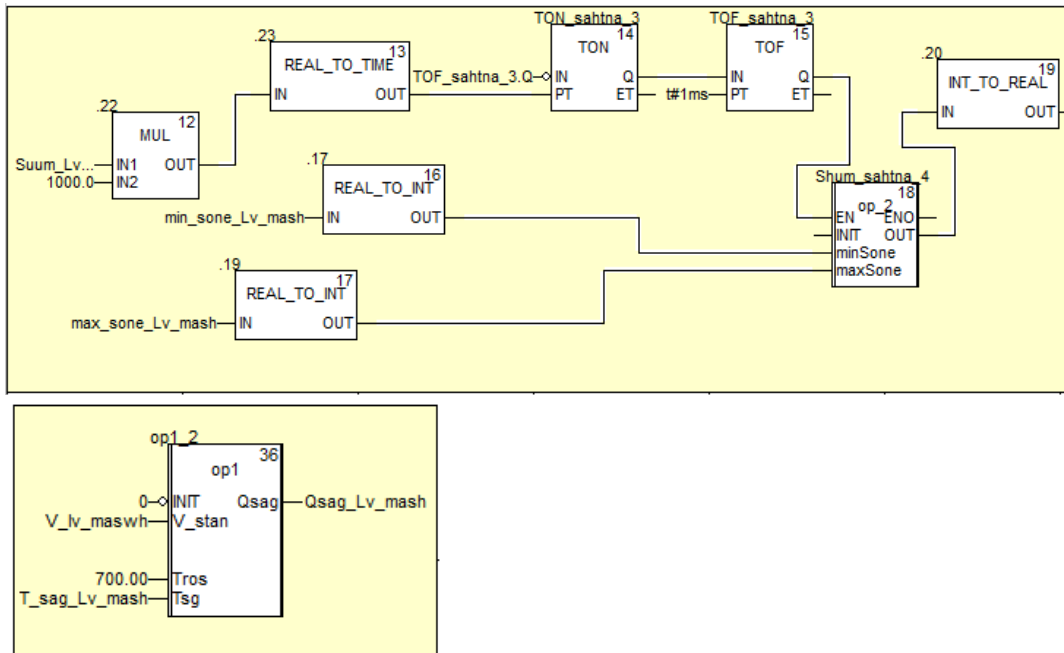
  L:=V/S;
end_if;

```

Д8.3.8 Секція "lv_mash" – для імітації системи автоматичного регулювання теплового режиму кристалізатора та анімації операторського екрану.



Програма включає 9 основних фрагментів: **что 1** ? моделі кристалізатора 2 і транспортного запізнювання 3, встановлення настройок ПІ регулятора з операторського екрану 4, увімкнення 5 та формування випадкових збурень 7, зупинки та старту всієї програми 6, розрахунку кількості теплоти, що виноситься із заготовкою з кристалізатора 9, ПІ-регулятора 8.



Лабораторна робота 8.4. Створення програмного забезпечення імітації роботи системи автоматичного керування швидкістю виткоутворювача моталки.

Мета: ознайомитися з операторським вікном візуалізації пристрою для гартування й охолодження катанки і здвоєної моталки кошикового типу АСУТП виробництва алюмінієвої катанки та основами створення системи автоматичного керування швидкістю виткоутворювача моталки в середовищі UNITY PRO.

Загальна постановка завдання. Необхідно створити проект в UNITY PRO для реалізації поставленої задачі. Настроювання системи автоматичного керування та дослідження якості її роботи виконати за допомогою готового програмного операторського екрану.

Опис об'єкта 3. У третьому вікні імітатора представлено пристрій для гартування й охолодження катанки та здвоєна моталка кошикового типу. Також передбачені візуальні засоби контролю та керування відповідними технологічними параметрами (рис.8.12).

З пульта оператора за допомогою кнопок можна увімкнути в роботу та вимкнути моталку, настроїти її роботу на відповідний діаметр катанки із заданим кроком укладання. При цьому на екрані відбувається емуляція її роботи та графік зміни швидкості виткоутворювача залежно від установлених параметрів формування бунта.

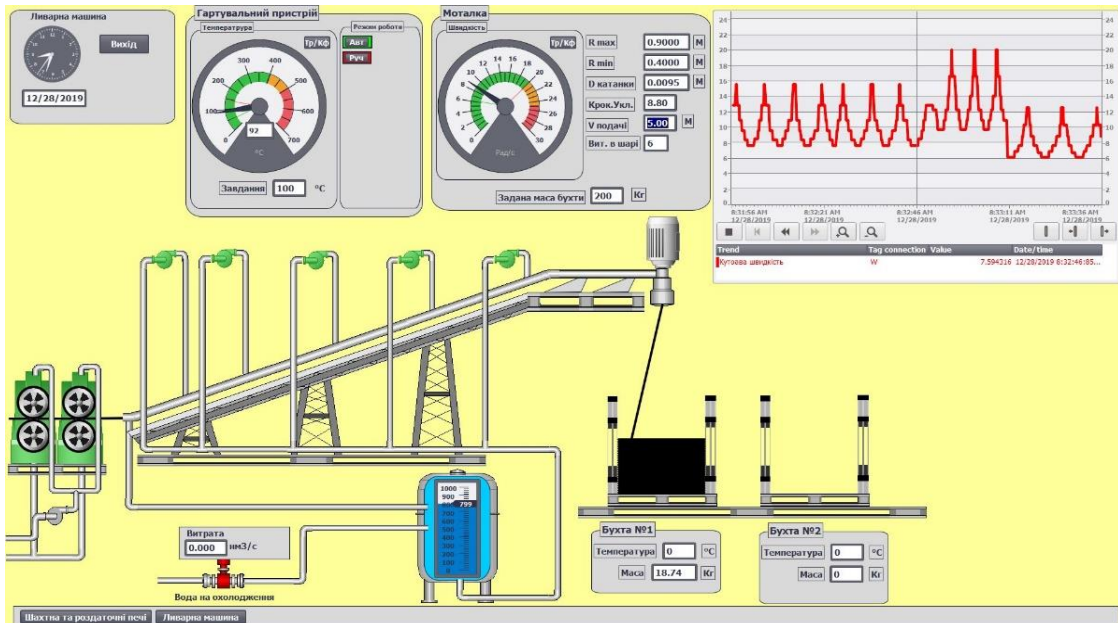


Рисунок 8.12 – Вікно візуалізації гартувального пристрою та моталки

Перед виконанням лабораторної роботи ознайомтесь з додатком Д8.4.

- 1) Запустити на виконання UNITY PRO. Створити новий проект з ПЛК M340 на базі центрального процесора VMX R34 2000 02.70 CPU340-20 Modbus.
- 2) Імпортувати операторський екран:
Project Browser -> контекстне меню Operator Screens -> Import-> "Робочий стіл\тека Лаб_ПрПК_Мотал\Screen.XCR".
- 3) Активізувати можливість використання динамічних масивів:
Меню Tools->Project Settings-> Variableless->виставити опцію "Allow Dynamic Arrays" .
- 4) Імпортувати програмну секцію: **Project Browser -> Program-> Tasks -> Mast -> контекстне меню Sections -> Import "Робочий стіл\тека Лаб_ПрПК_Мотал\Program.XPG".**
- 5) Імпортувати змінні:
Project Browser -> Variables & FB Instances -> Import "Робочий стіл\тека Лаб_ПрПК_Мотал\ Variables.XSY".
- 6) Після імпорту змінних, ознайомитись з їх призначенням (рис.8.13).

| Name | Type | Value | Comment |
|-----------------|------|--------|--|
| a | REAL | 8.8 | Крок укладання витків |
| D_kat_mot | REAL | 0.0095 | Діаметр катанки |
| dop_1 | BOOL | | //////////Допоміжна змінна////////// |
| dop_2 | BOOL | | //////////Допоміжна змінна////////// |
| dop_3 | BOOL | | //////////Допоміжна змінна////////// |
| i | REAL | | //////////Допоміжна змінна////////// |
| i_2 | REAL | | //////////Допоміжна змінна////////// |
| Mas_1_shar | REAL | | Маса одного шару |
| mas_b2 | REAL | | //////////Допоміжна змінна////////// |
| mas_sadan | REAL | 50.0 | Задана маса бухти |
| mas_tek | REAL | | Поточна маса бухти |
| N | INT | | Кількість витків у одному шарі |
| N_TEK | INT | | Поточний виток |
| Otgr_buht | BOOL | | //////////Допоміжна змінна////////// |
| out_regtem_s... | REAL | | //////////Допоміжна змінна////////// |
| R_max | REAL | 0.9 | Максимальний радіус |
| R_min | REAL | 0.3 | Мінімальний радіус |
| Shsr | INT | | //////////Допоміжна змінна////////// |
| St_kn | BOOL | | //////////Допоміжна змінна////////// |
| Start | BOOL | | Старт |
| Stop | BOOL | | Стоп |
| Stop_lv_mh | BOOL | | //////////Допоміжна змінна////////// |
| Str_kn | BOOL | | //////////Допоміжна змінна////////// |
| time_1 | TIME | | Час для кожного значення кутової швидкості |
| time_2 | TIME | | Час для кожного значення кутової швидкості |
| V_pod | REAL | 6.2 | Швидкість подачі |
| Var1_motal | BOOL | | //////////Допоміжна змінна////////// |
| var2 | BOOL | | //////////Допоміжна змінна////////// |
| Var2_motal | BOOL | | //////////Допоміжна змінна////////// |
| var3 | BOOL | | //////////Допоміжна змінна////////// |
| var4 | BOOL | | //////////Допоміжна змінна////////// |
| var5 | BOOL | | //////////Допоміжна змінна////////// |
| Var_ctu | BOOL | | //////////Допоміжна змінна////////// |
| var_Sagr_pstr | BOOL | FALSE | //////////Допоміжна змінна////////// |
| W | REAL | | Загальна кутова швидкість |
| W_1 | REAL | | Кутова швидкість обертання проводки, рад/с |
| W_2 | REAL | | Кутова швидкість обертання проводки, рад/с |

Рисунок 8.13 – Змінні проекту

7) Активізувати екран оператора і ознайомитись з його структурою:

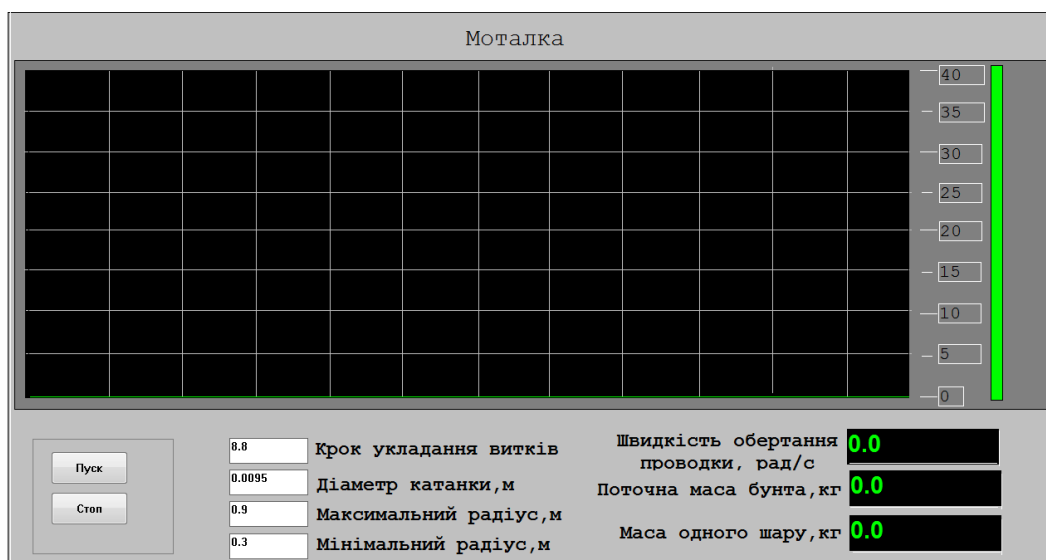


Рисунок –8.14 Екран оператора.

8) Перевірити проект на наявність помилок. Якщо помилки є повторити пункти 1-7 знову.

9) Створити 4 функціональні блоки користувача програму «mas_kat», «N_K», «V_ugl_max» і «V_ugl_min», користуючись варіантами, що приведені

у додатках Д8.4.3, Д8.4.4, Д8.4.5.

Увага! Використовувати вже існуючі (імпортовані) змінні та екземпляри функціональних блоків, інакше операторський екран не буде функціонувати!

10) Скопіювати проект і завантажити його в симулятор ПЛК.

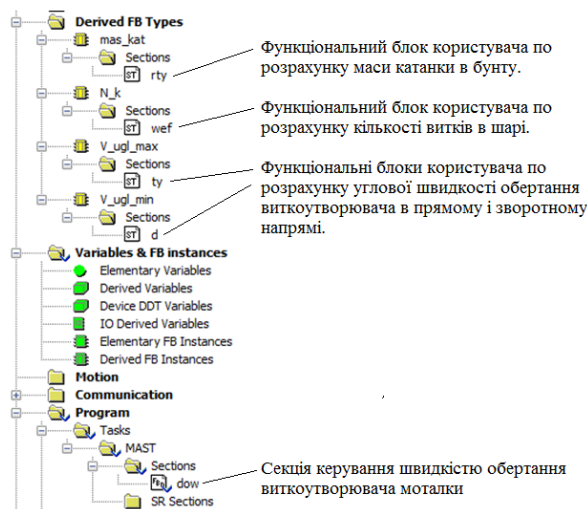
11) За допомогою панелі управління, рис.8.15, дослідити якість керування укладанням катанки в бунт при діаметрах 0,0095м і 0,012 м та кроці укладання витків у шарі 8,8; 7,0;5,0.



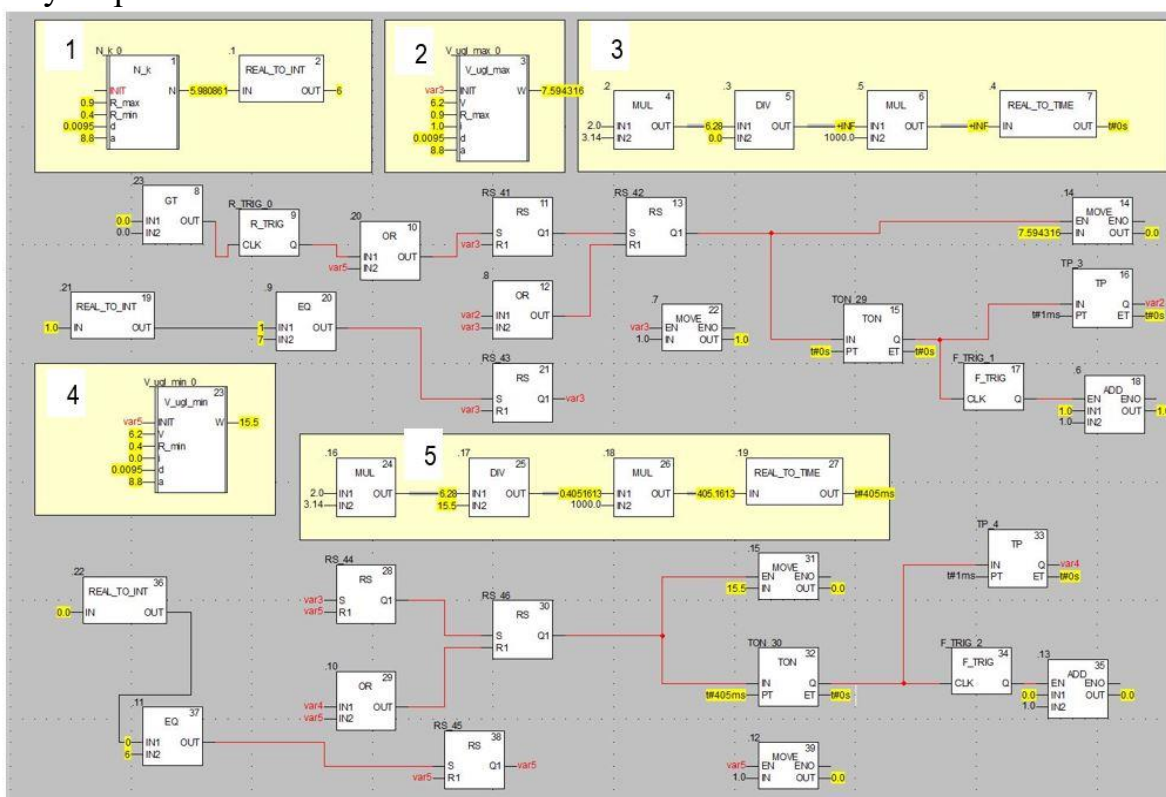
Рисунок– 8.15. Екран оператора при автоматичному регулюванні

ДОДАТОК 8.4 Опис програмних секцій імітації роботи системи автоматичного керування швидкістю виткоутворювача моталки в UNITY PRO.

Д8.4.1 Структура програмного забезпечення проекту. Для реалізації даної задачі використовуються чотири функціональних блока користувача і одна секція (рис.Д8.3.1): функціональний блок користувача «mas_kat» – для розрахунку маси катанки в бунту, функціональний блок користувача «N_K» – для розрахунку кількості витків у шарі, функціональний блок користувача «V_ugl_max» – для розрахунку кутової швидкості обертання виткоутворювача у прямому напрямі, «V_ugl_min»– для розрахунку кутової швидкості обертання виткоутворювача у зворотному напрямі, а секція «dow» – для імітації автоматичного керування швидкістю виткоутворювача моталки і анімації операторського екрану.



Д8.4.2. Код програми формування бунту катанки і керування швидкістю виткоутворювача металки:



У блоці 1 розраховується кількість витків у шарі, у блоках 2 і 4 – кутова швидкість провідки виткоутворювача при укладанні витків, відповідно, від периферії до центру і навпаки, а у блоках 3 і 5 розраховується час формування витків при кожній швидкості виткоутворювача.

Д8.4.3 Функціональний блок користувача «mas_kat» по розрахунку маси катанки, що укладена в бунт, за формулою $m_s = l \frac{\pi d^2}{4} \cdot \rho$,

де l – довжина середнього витка бухти катанки; d – діаметр катанки; ρ – густина алюмінію.

| | | |
|---|---|-------------|
| mas_kat | | <DFB> |
| <inputs>< td=""><td></td><td></td></inputs><> | | |
| R_max | 1 | REAL |
| R_min | 2 | REAL |
| d | 3 | REAL |
| init | 4 | BOOL |
| n | 5 | REAL |
| <outputs>< td=""><td></td><td></td></outputs><> | | |
| m | 1 | REAL |
| m_s | 2 | REAL |
| <inputs ...><="" td=""><td></td><td></td></inputs> | | |
| public> | | |
| pi | | REAL 3.14 |
| ro | | REAL 2700.0 |
| Rcp | | REAL |
| l | | REAL |
| private> | | |
| sections> | | |
| rty | | <ST> |

```

s_t rty <DFB> : [mas_kat]
If init=1 then
m:=0.0;
else
Rcp:=(R_max+R_min)/2.0;

l:=2.0*pi*Rcp;

m:=(1*((pi*d*d)/4.0))*ro*n;

m_s:=(1*((pi*d*d)/4.0))*ro;
end_if;

```

Д8.4.4 Функціональний блок користувача «N_K» по розрахунку кількості витків у шарі за формулою $m = \frac{R_{\max} - R_{\min}}{d \cdot \alpha}$,

де R_{\max} – радіус максимального витка; R_{\min} – радіус мінімального витка; d – діаметр катанки, α – крок укладання витків.

| | | |
|---|---|-------|
| N_k | | <DFB> |
| <inputs>< td=""><td></td><td></td></inputs><> | | |
| INIT | 1 | BOOL |
| R_max | 2 | REAL |
| R_min | 3 | REAL |
| d | 4 | REAL |
| a | 5 | REAL |
| <outputs>< td=""><td></td><td></td></outputs><> | | |
| N | 1 | REAL |
| <inputs ...><="" td=""><td></td><td></td></inputs> | | |
| public> | | |
| N0 | | REAL |
| private> | | |
| sections> | | |
| wef | | <ST> |

```

s_t wef <DFB> : [N_k]
If init then
N:=N0;
end_if;

N:=(R_max-R_min)/(a*d);

```

Д8.4.5 Функціональні блоки користувача «V_ugl_max» і «V_ugl_min» по розрахунку кутової швидкості обертання виткоутворювача для i -того витка бунта за формулою $\omega_i = \frac{V}{R_{\max(\min)} \mp (i-1)d \cdot \alpha}$,

де V – швидкість подачі катанки, R_{\max} – радіус максимального витка; R_{\min} – радіус мінімального витка; d – діаметр катанки, α – крок укладання витків.

| | | |
|---|---|----------|
| V_ugl_max | | <DFB> |
| <inputs>< td=""><td></td><td></td></inputs><> | | |
| INIT | 1 | BOOL |
| V | 2 | REAL |
| R_max | 3 | REAL |
| i | 4 | REAL |
| d | 5 | REAL |
| a | 6 | REAL |
| <outputs>< td=""><td></td><td></td></outputs><> | | |
| W | 1 | REAL |
| <inputs ...><="" td=""><td></td><td></td></inputs> | | |
| public> | | |
| V0 | | REAL 0.0 |
| private> | | |
| sections> | | |
| ty | | <ST> |

```

s_t ty <DFB> : [V_ugl_max]
If init=1 then
W:=0.0;
else
W:=V/(R_max-i*d*a);
end_if;

```

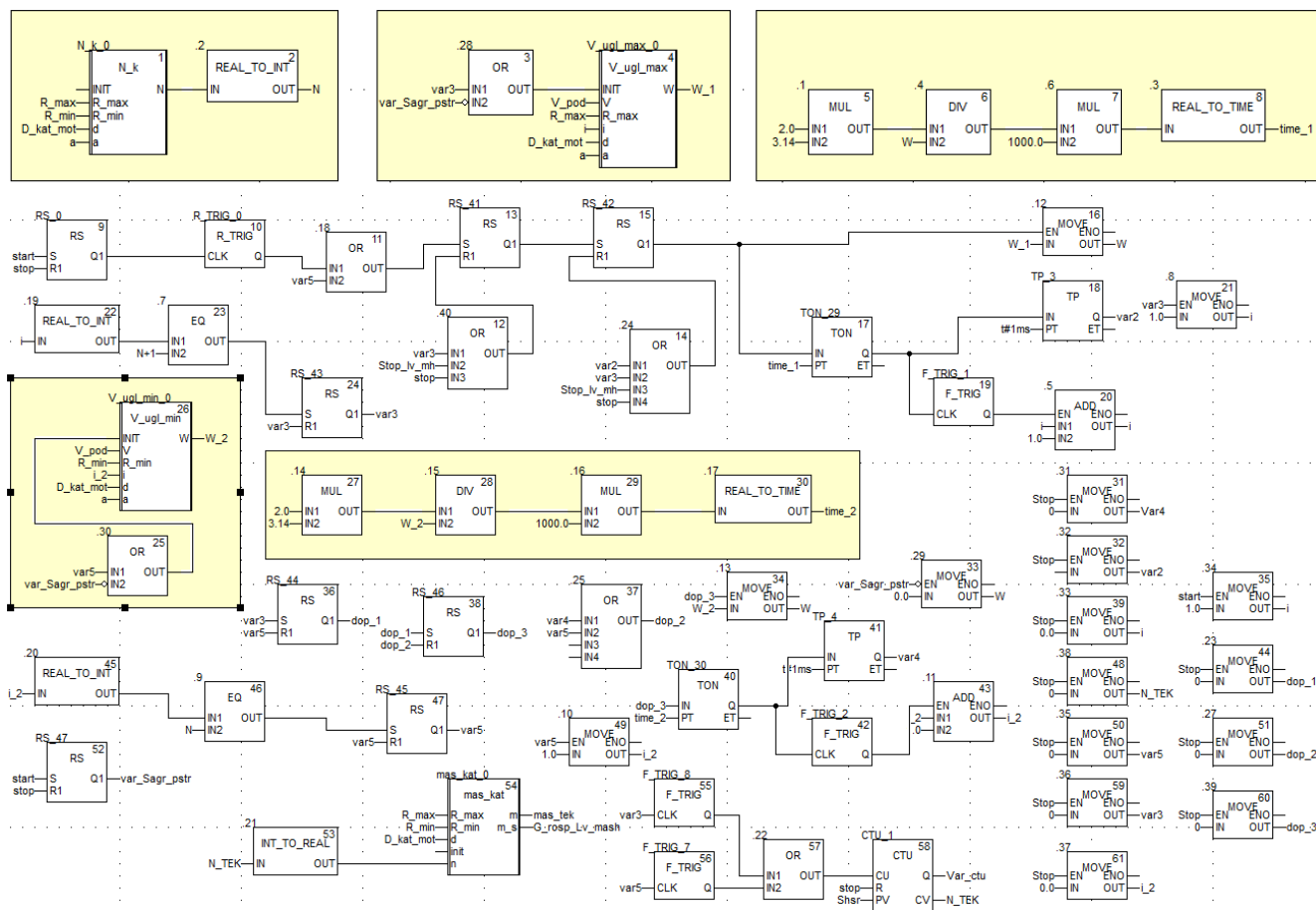
| | | |
|---|---|----------|
| V_ugl_min | | <DFB> |
| <inputs>< td=""><td></td><td></td></inputs><> | | |
| INIT | 1 | BOOL |
| V | 2 | REAL |
| R_min | 3 | REAL |
| i | 4 | REAL |
| d | 5 | REAL |
| a | 6 | REAL |
| <outputs>< td=""><td></td><td></td></outputs><> | | |
| W | 1 | REAL |
| <inputs ...><="" td=""><td></td><td></td></inputs> | | |
| public> | | |
| V0 | | REAL 0.0 |
| private> | | |
| sections> | | |
| d | | <ST> |

```

s_t d <DFB> : [V_ugl_min]
If init=1 then
W:=0.0;
else
W:=V/(R_min+i*d*a);
end_if;

```

Д8.4.6 Секція коду програми формування бунту катанки і керування швидкістю виткоутворювача моталки



Надо вкратце прокоментировать, что к чему.

Аналіз одержаних результатів

Викладачем перевіряється виконання поставленого завдання. Студент повинен пояснити FBD-програму керування укладанням катанки в бунт, дати характеристику задіяним функціональним блокам і зробити висновки за проведеними дослідженнями.

Запитання для самоперевірки

1. За якими розрахунками відбувається автоматичне керування укладанням катанки у бунти?
2. Прокоментувати програмний код секції «dow».
3. Як створюється функціональний блок користувача?
4. Прокоментувати фрагмент програми розрахунку часу формування витків при кожній швидкості виткоутворювача.
5. Які результати виконаних досліджень?
6. Що запрограмовано у жовтих прямокутниках секції «dow»?

Рекомендована література

1. Ніколаєнко А.М., Пупена О.М., Ельперін І.В. Програмування промислових контролерів. Конспект лекцій. Для студентів спеціальності 151 "Автоматизація та комп'ютерно-інтегровані технології" денної та заочної форм навчання.
2. Пупена, О.М. Програмування промислових контролерів у середовищі
3. Unity Pro : навч. посіб. / О.М. Пупена, І.В. Ельперін. – К. : вид-во «Ліра-К», 2013. – 376 с.
4. www.schneider-electric.com.ua