

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»



**ПРОГРАМУВАННЯ ПРОМИСЛОВИХ КОНТРОЛЕРІВ  
MITSUBISHI MELSEC СЕРІЇ FX**

Харків  
2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

**ПРОГРАМУВАННЯ ПРОМИСЛОВИХ КОНТРОЛЕРІВ  
MITSUBISHI MELSEC СЕРІЇ FX**

Методичні вказівки  
для проведення комп'ютерного практикуму з навчальної дисципліни  
«Програмне забезпечення промислових контролерів»  
для студентів спеціальності 151  
«Автоматизація та комп'ютерно-інтегровані технології»  
усіх форм навчання

Затверджено  
редакційно-видавничою  
радою університету,  
протокол № 1 від 28.01.2022 р.

Харків  
НТУ «ХП»  
2022

**Методичні** вказівки для проведення комп'ютерного практикуму «Програмування промислових контролерів Mitsubishi MELSEC серії FX» з навчальної дисципліни «Програмне забезпечення промислових контролерів» для студентів спеціальності 151 – «Автоматизація та комп'ютерно-інтегровані технології» усіх форм навчання / Уклад.: Лисаченко І.Г., Дзевочко А.І. – Харків: НТУ «ХП», 2022. – 81 с.

Укладачі: І. Г. Лисаченко  
А. І. Дзевочко

Рецензент: І. Л. Красніков, доц. канд. техн. наук,  
проф. каф. автоматизації технологічних систем  
та екологічного моніторингу

Кафедра автоматизації технологічних систем  
та екологічного моніторингу

## ВСТУП

Методичні вказівки розроблені з метою отримання здобувачами вищої освіти за освітньою програмою «Автоматизація та комп'ютерно-інтегровані технології» необхідних компетентностей (К17) та програмних результатів (ПР10) [1]. Вони використовуються для проведення комп'ютерного практикуму із навчальної дисципліни *«Програмне забезпечення промислових контролерів»* та вміщують необхідні відомості для розробки прикладного програмного забезпечення (ППЗ) для промислових контролерів.

У результаті вивчення навчальної дисципліни *«Програмне забезпечення промислових контролерів»* здобувач вищої освіти повинен:

### **знати:**

- основні принципи стандарту ІЕС61131-3 [2, 3];
- оператори та синтаксис технологічних мов програмування;
- класифікацію, технічні характеристики, принципи побудови та функціонування промислових контролерів (програмованих логічних контролерів – ПЛК);
- порядок створення, налагодження та завантаження програм користувача до ПЛК;
- особливості підключення до ПЛК датчиків та виконавчих механізмів на локальному рівні та його місце в автоматизованих системах управління;

### **вміти:**

- створювати програми користувача для ПЛК;
- проводити конфігурування ресурсів ПЛК;
- встановлювати зв'язок з ПЛК та завантажувати до нього програми користувача (управляючи програми);

Також, у процесі вивчення навчальної дисципліни здобувач повинен ознайомитися із інтерфейсом користувача середовища програмування та вивчити методику розроблення ППЗ промислових контролерів для автоматизації технологічних систем та комплексів.

Для розроблення та завантаження ППЗ до складу програмно-технічного комплексу (ПТК) входить ПК із встановленим спеціальним програмним забезпеченням (СПЗ) – середовищем GX Works2 від компанії *Mitsubishi Electric Corp.* [4, 5]. У подальшому будемо використовувати скорочення для назви середовища – GX\_W2. Керівництво з програмування промислових контролерів можна завантажити за посиланням [6].

Для подальшого виконання завдань практикуму потрібно завантажити файл архіву за посиланням [GXW2.ZIP](#) та встановити середовище. Цей програмний продукт потребує ліцензування, тому його демо-версія має обмеження використання протягом 60 діб. Тому завдання потрібно виконати впродовж наступних 60 діб після встановлення. Необхідно відзначити, що середовище GX Works2 відповідає стандарту IEC61131-3, який визначає принципи програмування промислових контролерів. Середовище GX Works2 пропонує вибір мов програмування на бажання розробника. Це може бути як графічні мови (Ladder, SFC), так і текстові (ST) у випадку вибору типу «простий проєкт» під час ініціювання. У випадку розроблення проєкту типу «структурований проєкт» перелік мов збільшується додаванням мови Structured Ladder/FBD.

Існує інше середовище для програмування контролерів від *Mitsubishi Electric Corp.*. Це GX Developer (посилання для завантаження [GX\\_Developer.zip](#)), яке теж є IEC-орієнтованим, але воно старіше ніж GX Works2. Завантажити ПЗ можна на сайті виробника контролерів та програмного забезпечення, або на порталі вільного програмного забезпечення за посиланням <https://gx-developer.software.informer.com/8.3/>. У середовищі GX Developer додатково до стандартних п'яти мов програмування є мова MELSEC IL, яка є спрощеною мовою програмування, ніж стандартна мова Instruction List (IL).

У цілому, обидва згадані середовища є подібними, та мають цілком зрозумілий інтерфейс користувача (є *Windows*-орієнтованими додатками). Крім того, формат та нотація мов програмування відповідають стандарту *IEC61131-3* із невеликими відмінностями. Але, для засвоєння принципів програмування промислових контролерів це не є великою проблемою. Вбудовані редактори дозволяють визначати помилки синтаксису мов програмування вже на етапи компілювання та вказують на можливі способи виправлення помилок ще до завантаження програми користувача до ПЛК.

Середовище *GX Developer* підтримує мови із списком інструкцій (*IL*), сходою діаграмою (*LD*) та діаграмою послідовної функції (*SFC*). У цьому середовищі можливе перемикання між мовами *IL* і *LD* за бажанням користувача під час роботи (рівень локалізації перемикання – це програмний компонент). Також можливо запрограмувати функціональні блоки користувача (серія *MELSEC QnA/QnAS/System Q*), а також доступний широкий спектр утиліт для налаштування спеціальних функціональних модулів для *MELSEC System Q*. Причому термін «налаштувати» – це головне слово, тобто можливе лише конфігурування параметрів функціональних модулів.

Зауважимо, що для окремих завдань може застосовуватися обидві системи програмування відповідно до обраної серії та типу контролера. У навчальних стендах використані контролери серії *MELSEC FX* різних модифікацій. Але можливе використання інших моделей контролерів з наступним відладженням проекту на доданої до *GX Works2* програмі-симуляторі.

Для імітування роботи систем керування технологічним процесом використовуються комбіновані стенди із додатковим обладнанням. Ці стенди мають у своєму складі ПЛК, об'єкт керування (модель теплообмінника) та дискретні і аналогові датчики та вихідні пристрої (електромеханічні реле). Модель теплообмінника складається із нагрівача (лампа розжарювання номінальної потужності *20W/220VAC*) та охолоджувача (вентилятор обдування, *+12 VDC*). Контрольованим парамет-

ром є температура повітря всередині теплообмінника, яка вимірюється за допомогою термометра опору з нормальним перетворювачем у електричний сигнал  $0...+10\text{ V}$ . Для імітування сигналів від дискретних датчиків до ПЛК підключені перемикачі типу «сухий контакт» із температурною залежністю (тобто, термоперемикачі типу NO та NC із температурою перемикання  $+40^{\circ}\text{C}$ ). Також, на окремих стендах є можливість дискретного встановлення (чотири позиції) аналогового сигналу на вході контролера за допомогою електричного модуля формування аналогових сигналів із зовнішнім живленням  $+5\text{ VDC}$ .

Метою виконання завдань комп'ютерного практикуму є закріплення теоретичних знань та отримання практичних навичок із програмування ПЛК під час розроблення систем управління технологічними процесами. Завдання комп'ютерного практикуму розроблені за принципом їхнього поступового ускладнення – від звичайного логічного керування до реалізації законів регулювання (наприклад, двопозиційного закону регулювання). Також, конструкція стенду дозволяє реалізувати аналогове керування дискретним вихідним елементом, так зване ШІМ-регулювання.

Усі додаткові відомості щодо програмування та використання контролерів серії MELSEC FX наведені у Додатках А та Б.

Отримані знання та навички передбачається використати під час вивчення навчальних дисциплін здобувачами другого освітнього рівня (магістра) за спеціальністю 151 – «Автоматизація та комп'ютерно-інтегровані технології».

### *Практичне завдання 1*

## ЗАГАЛЬНІ ПРИНЦИПИ ПРОГРАМУВАННЯ КОНТРОЛЕРІВ MELSEC СЕРІЇ FX У СЕРЕДОВИЩІ GX WORKS 2

### **1.1 Мета завдання**

- ознайомлення із обладнанням стенда, вивчення його принципової та структурної схем;
- закріплення теоретичних знань щодо побудови промислових логічних контролерів серій MELSEC FX різних модифікацій та структури і інтерфейсу користувача програмного середовища GX WORKS 2;
- отримання практичних навичок та прийомів роботи у середовищі GX WORKS 2 у процесі створення ППЗ для ПЛК: *створення проєкту, конфігурування ресурсів контролера, написання програми користувача, компіляція, підключення до ПЛК та завантаження проєкту до нього.*

### **1.2 Опис стенда**

Загальний вигляд стенда та опис його складових та принцип дії наведений у *Додатку А*. Насамперед відзначимо, що стенд розроблений для виконання комп'ютерних практикумів із декількох дисциплін. Тому до його складу входять елементи, які не використовують під час вивчення навчальної дисципліни «Програмне забезпечення промислових контролерів». Це, наприклад панелі оператора та SCADA-системи, які з'єднані із контролером послідовними інтерфейсами типу TTL/5V. Осо-



близькістю стенда є те, що контролер має лише один інтерфейс для обміну даними. Тому робота контролера можлива, або із середовищем програмування, або із вказаними пристроями та SCADA-системами.

Перевірка правильності роботи програми користувача здійснюється у режимі ONLINE під час з'єднання із середовищем програмування.

### **1.3 Порядок виконання завдання**

Виконання практичного завдання розраховано на 4 ак. год. та складається з таких етапів:

#### 1) Створення проєкту у середовищі GX\_W2:

- вибір цільового ПЛК (тип та модель контролера) та мови програмування основного програмного організаційного компоненту типу Program із ім'ям MAIN, збереження файлу проєкту на жорсткому диску ПК;
- конфігурування параметрів ПЛК (за необхідністю);
- заповнення коментарю до проєкту (за необхідністю);
- підключення додаткових бібліотек (за необхідністю);
- розроблення програми користувача;
- компілювання проєкту,

#### 2) Підключення середовища GX\_W2 (ПК) до ПЛК та завантаження до нього машинного коду проєкту.

- підключення до ПЛК та завантаження машинного коду проєкту;
- перевірка роботи програми користувача (у режимі ONLINE або за допомогою симулятора).

#### 3) Налаштування проєкту в покроковому режимі та перевірка роботи програми користувача для управління технологічним процесом.

### **1.4 Хід виконання завдання**

#### *1.4.1 Створення проєкту у середовищі GX\_W2*

Для самостійного виконання завдань практикуму здобувач освіти повинен завантажити архівний файл із вказаних у вступі ресурсів та

встановити середовище GX\_W2 на ПК. Середовище GX DEVELOPER є вільно розповсюджуваним програмним забезпеченням та не потребує ліцензування. Обмеженням є те, що це програмне забезпечення на цей час вже не підтримується офіційним виробником – компанією *Mitsubishi Electric Corp.*. Сучасне ПЗ для програмування промислових контролерів серій MELSEC від компанії *Mitsubishi Electric Corp.* є ліцензованим. Наприклад, таким середовищем є програмний комплекс GX Works2/3.

Отже, якщо ПЗ GX\_W2 встановлено, то потрібно його запустити стандартним способом для Windows-додатків. Це може бути команда


меню у переліку Пуск або піктограма  на робочому столі ПК.

У вступі було зазначено що ПЗ GX\_W2 є Windows-орієнтованими. Це означає, що у середовищі діють основні правила роботи із Windows-додатками, усе що стосується роботи з файлами та буфером обміну, правил редагування та використання «гарячих» клавіш та команд. У подальшому деякі дії розробника програмного забезпечення у середовищі позначимо так: натиснення на ліву кнопку миші (ЛКМ) – вибір, подвійне натискання на ліву кнопку миші (ПЛКМ) – виконання команди, натиснення на праву кнопку миші (ПКМ) – виклик контекстного меню.

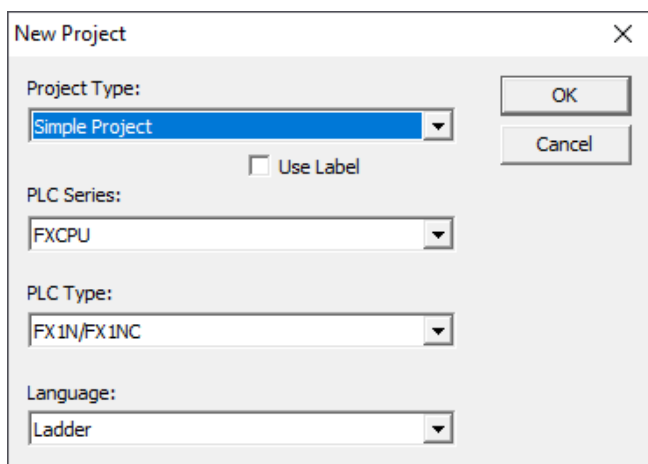
Після запуску ПЗ GX\_W2 відкриється вікно із робочим простором. Якщо це перший запуск після встановлення ПЗ, то робочий простір, перелік меню та піктограми швидких команд будуть не усі активними (доступними). Нижче зображено фрагмент робочого простору після запуску ПЗ GX\_W2:



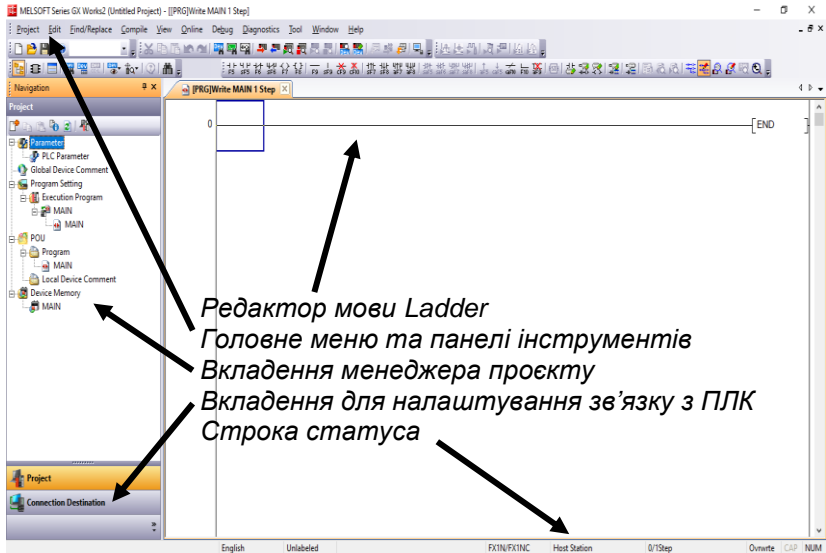
Для початку процесу розроблення прикладного програмного забезпечення потрібно створити проєкт. Це такий собі контейнер, який вміщує усі результати розроблення прикладного програмного забезпечення, тобто, програму керування із допоміжними елементами: організаційними компонентами, переліками змінних, параметрами налаштувань апаратних засобів, доданими бібліотеками, тощо.

Для створення проєкту потрібно ЛКМ використати відповідну піктограму (  ), яка інтуїтивно нагадує створення нового документа, як в інших *Windows*-додатках. В результаті з'явиться вікно **New Project**. Це вікно **New Project** розділено на поля. Необхідно, за допомогою миші, яке випадає, заповнити ці поля. Особливістю є вибір типу проєкту – простий або структурований. У останньому випадку це проєкт, який може складатись із декількох програмних компонентів.


Якщо вибрати налаштування такими, що відповідають зображенню нижче, то для продовження потрібно ЛКМ натиснути на кнопку **OK**.



Далі з'явиться вікно із робочим простором, як це зображено нижче:




Редактор мови Ladder  
 Головне меню та панелі інструментів  
 Вкладення менеджера проекту  
 Вкладення для налаштування зв'язку з ПЛК  
 Строка статусу

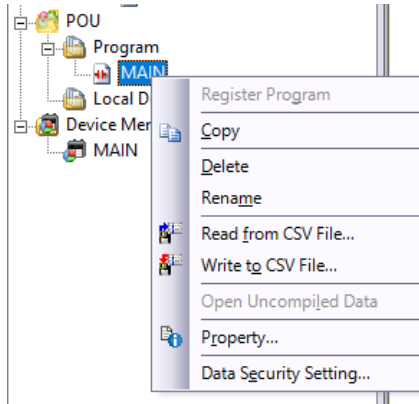
Для збереження проекту на жорсткому диску потрібно ЛКМ натиснути на піктограму  (або за допомогою команди **Save** у переліку меню **Project** стандартним для *Windows*-додатків способом). Отже, якщо вибрати команду **Save**, то відкриється наступне вікно та з'явиться можливість для створення робочої папки для зберігання файлів проекту та можливість надання імені проекту із додаванням коментаря. Для завершення дії збереження проекту потрібно ЛКМ натиснути на кнопку **Зберегти**. У випадку, якщо папка не існувала і проект з таким ім'ям відсутній, то середовище зробить запит на їхнє створення. Потрібно лише підтвердити свої дії натисненням ЛКМ на кнопку **Так**. Зовнішній вигляд активного робочого простору із позначеннями зон зображений вище.

Робочий простір із відкритим проектом поділений на декілька зон із різним функціональним призначенням, як це прийнято у багатьох *Windows*-додатках. Наведемо короткий опис основних зон робочого простору.

Зона **Navigation** (вікно навігації у проект), це зона, в якій відо-

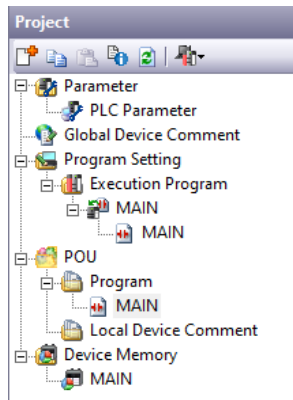
бражені усі результати дій розробника щодо компонентів проєкту та взаємодії з ПЛК. Ця зона складається з двох вкладень: **Project** та **Connection Destination**. Перше вкладення (**Project**) складається із усіх компонентів проєкту, які розробляються або налаштовуються. Також у цьому вкладенні створюються або видаляються нові компоненти у процесі розроблення. Друге вкладення (**Connection Destination**) призначене для налаштування зв'язку з ПЛК. У новому проєкті ця зона вже має деякі компоненти, які додані за умовчанням та мають стандартні налаштування, які відповідають більшості завдань щодо розроблення ППЗ. Для збільшення розміру зони редагування програмного компоненту відображення зона **Navigation** може бути схованою за допомогою команди **Navigation Window** у переліку меню **View/Docking Window**. На панелі інструментів є також піктограма кнопки-перемикача , яка теж вмикає або вимикає відображення зони **Navigation**.

Структура вкладення **Project** має плаский ієрархічний вигляд, тобто, подібна так званому «дереву». Керування компонентами відповідає принципу роботи будь-якого *Windows*-додатка. Наприклад, розкриття/закриття гілок здійснюється натисканням ЛКМ на інтерактивні символи **+** та **-**, які розміщені на вузлах (гілках) дерева проєкту. Якщо потрібно викликати контекстне меню, то потрібно натиснути ПКМ у відповідному вузлі (гілці). Нижче зображено фрагмент вкладення **Project** із активованим контекстним меню програмного компонента **MAIN**:



Зауважимо, що склад вікна із контекстним меню у цілому залежить від позиції, де воно викликане. Наприклад, у вкладенні Project це додавання, копіювання, видалення або перейменування компонента. Також можливий перегляд властивостей обраного об'єкту та читання або записування файлів формату file.CSV. У цих файлах зберігаються дані та опис компонентів проекту.

Нижче зображено вигляд вкладення Project.

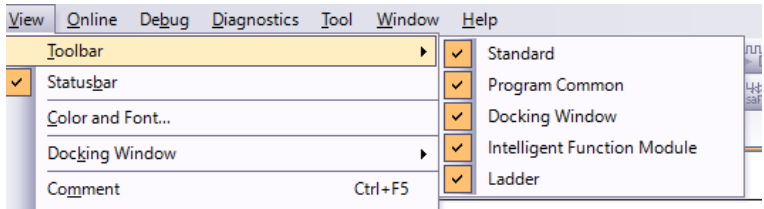


За умовчанням до складу вкладення Project входять такі компоненти: Parameter, Program Setting, POU та Device memory. Зараз

звернемо увагу лише на гілку **POU**. Ця гілка складається із програмних компонентів проєкту. У даному випадку проєкт вміщує лише один програмний компонент. Це основна програма з ім'ям **MAIN**. Із цієї програми починається виконання циклу сканування системою виконання ПЛК. Зауважимо, що якщо виконати дію ПЛКМ по об'єкту **MAIN**, то у зоні редактора відкриється вміст програми **MAIN**. Під час створення проєкту було визначено мову реалізації проєкту – **Ladder**. Це релейно-контактна схема. Відповідно програма **MAIN** реалізована мовою **Ladder**. На цьому етапі вона вміщує лише один ланцюг, який є завершальним у будь-якому програмному компоненті. Це команда **END**. Вона передає дозвіл системі виконання ПЛК для переходу на наступний цикл сканування. Фактично контролер буде виконувати перезапуск циклів сканування без виконання інших дій, тобто виконувати порожню програму.

Інструменти в панелях згруповані за функціональними признаками. Це панель керування файлами проєкту, панель керування буфером обміну, панель взаємодії із контролером та панелі з інструментами програмування на мовах **Ladder** та **SFC**. Інструменти мови **SFC** для створеного проєкту на панелі сховані. Панелі з інструментами можуть вільно переміщуватись на екрані монітора ПК за допомогою миші, якщо натиснути ЛКМ в кутовий точці панелі та не відпускати.

Зона з панелями інструментів складається із декількох панелей. За умовчанням відображаються панелі, які найбільш часто використовуються. Керувати станом відображення панелей можливо за допомогою команди **Toolbar** із переліку меню **View**. За цією командою викликається вікно налаштувань панелей інструментів **Toolbar**, яке зображене нижче:



У цьому вікні перелічені усі можливі панелі інструментів:

- **Standard** – панель із стандартними командами керування файлами проєкту (створення, відкриття, збереження)



- **Program Common** – панель взаємодії із ПЛК та буфером обміну



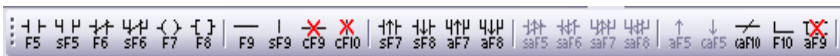
- **Docking Window** – панель керування робочим простором



- **Intelligent Function Module** – панель для взаємодії із спеціальними функціональними модулями



- **Ladder** – панель із командами мови релейно-контактних схем та визначення режиму редактора (читання або записування) та режиму роботи відгадчика (моніторинг або примусове призначення значень операндам)



Зауважимо, що з'являється коментар щодо призначення кожної піктограми на панелі інструментів, якщо позиціонувати покажчик миші до неї.

Взагалі усі команди та інструкції для редагування мовою Ladder



згруповані у переліку меню Edit, якщо активовано вікно з програмним компонентом.

#### *1.4.2 Розроблення програми користувача*

Для керування виходами ПЛК від стану його входів розробимо програму користувача. Далі визначені умови роботи виходів ПЛК в залежності від стану його входів (див. Дод. А.1).

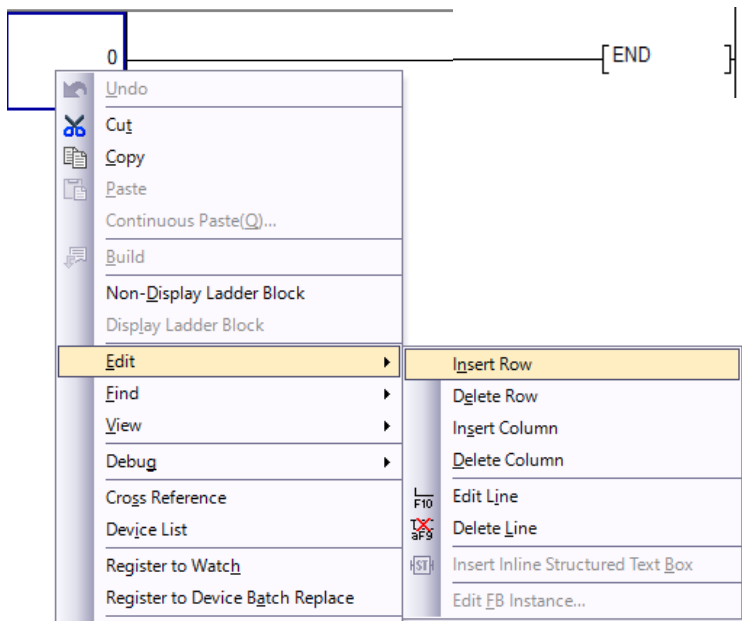
#### **Навчальне завдання.**

1. Вихід Y0 активується, якщо натиснути кнопку на вході X0.
2. Вихід Y1 активується, якщо натиснути кнопку на вході X1.
3. Вихід Y2 активується та фіксується, якщо натиснути кнопку на вході X2. Вихід вимикається за допомогою кнопки X0.
4. Вихід Y3 активується та фіксується, якщо натиснути кнопку на вході X3. Вихід вимикається за допомогою NC-термоперемикача на вході X4. Після замикаання X4 вихід Y3 залишається вимкненим. Повторне вмикання за допомогою кнопки X3.

*Алгоритм розроблення програми користувача.*


#### **Дія 1.**


Спочатку до програми із ланцюгом її завершення необхідно додати новий рядок для створення ланцюга. Для цього потрібно ЛКМ виділити ланцюг 0—[END] ліворуч, де вказано його номер (0) або можна кліпнути у будь-якому місці цього ланцюгу. З'явиться покажчик – рамка синього кольору. Далі ПКМ потрібно викликати контекстне меню. Контекстне меню зображено нижче:



Команда **Insert Row** додасть порожній новий рядок, який позначений сірим фоном. ЛКМ необхідно додати новий порожній ланцюг (або використавши гарячі клавіші **Shift+Ins**). В результаті вище ланцюга 0—[END] створиться рядок (смуга) сірого кольору.


## Дія 2.

Для заповнення створеного рядка необхідно використати інструменти панелі **Ladder**. Позначити ЛКМ порожній рядок покажчиком сирого кольору у будь-якому місці. Для додавання нормально-відкритого (NO) контакту ЛКМ кліпніть по піктограмі . В результаті середови-

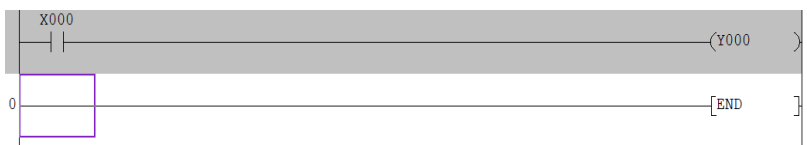
ще активує запит . Необхідно у порожнє поле записати адресу операнда. У даному випадку символ NO-контакта відповідає дискретним входам ПЛК та має позначення X. Для першої умови навчального завдання (див. стор. 15) це операнд X0.

Введіть цю адресу та ЛКМ натисніть на кнопку ОК або просто натисніть на клавішу Enter.

### Дія 3.

Далі необхідно додати операнд, який пов'язаний з дискретним виходом ПЛК. Для цього потрібно клікнути по піктограмі . У цьому випадку середовище теж активує запит Enter Symbol на адресу операнда «обмотка». Згідно першої умови завдання цей операнд має адресу Y0. Занесіть цю адресу та підтвердіть свій вибір.

В результаті робочий простір програми буде таким:




### Дія 4.

Для збереження внесених змін потрібно виконати команду **Build** з меню **Compile** або використати ФК F4. В результаті сірий фон ланцюга зміниться на білий фон.

Таким чином отримуємо ланцюг, який відповідає першій умові:





### Дія 5.

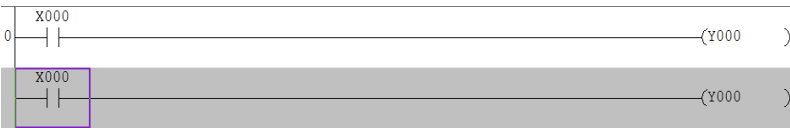
Друга умова завдання подібна до першої. Відмінність полягає лише у адресах операндів. Тому далі можна діяти двома способами: або також послідовно створити рядок та заповнити його операндами з потрібними адресами, або скористатися можливістю копіювання за допомогою буферу обміну ОС *Windows*. Зауважимо, що має бути включеним режим редагування – **Write Mode (F2)**. Перемикання режимів читання та редагування може здійснюватись за допомогою піктограм  на

панелі Ladder.

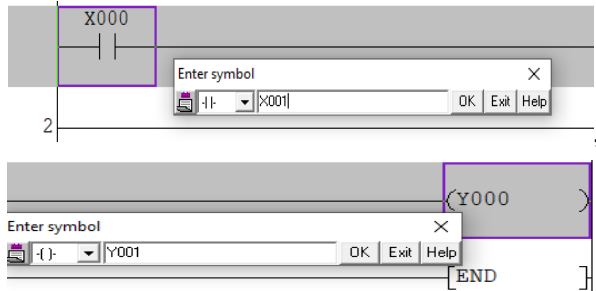
Для прискорення розроблення програми користувача скористаємось другим способом. Але, щоб отримати належний результат ПЗ GX\_W2 повинне бути переведено у режим додавання. Тобто, в середовищі є можливість редагування шляхом заміни та шляхом додавання. У режимі заміни буде редагуватися поточний ланцюг. Якщо його скопіювати та додати, він просто замінить поточний. В режимі додавання буде доданий такий же рядок, але після або до джерела копіювання. Режим редагування середовища визначається у рядку статусу (внизу робочого простору праворуч). Режим заміни має позначення *Ovrwrte*, режим додавання – *Insert*. Режими перемикаються за допомогою клавіші *Insert*.

Для копіювання поточного ланцюга потрібно ЛКМ позначити його початок та не відпускаючи ЛКМ провести по ньому до його кінця (тобто виконати стандартну дію в ОС *Windows*). В результаті ланцюг отримує фон синього кольору. Далі скориставшись стандартними діями ОС *Windows* скопіюємо виділену зону до буферу обміну (*Ctrl+C* або піктограма ) та додамо до програми (*Ctrl+V* або піктограма )

Результат додавання ланцюга зображений нижче (доданий ланцюг на фоні сірого кольору):

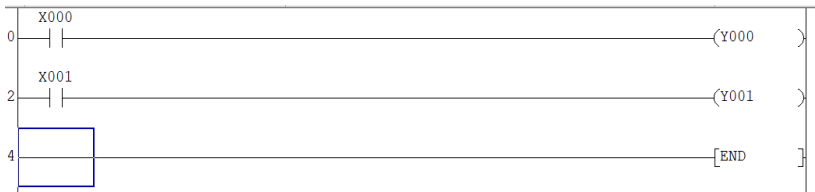


Далі потрібно зробити зміну адрес операндів відповідно до другої умови завдання. Але насамперед необхідно режим редагування змінити на режим заміни (*Ovrwrte*). Далі потрібно ЛКМ кліпнути по операндам у виділеному ланцюгу та замінити їхні адреси на відповідні другій умові завдання:



Далі необхідно зафіксувати зроблені зміни (використати ФК F4).



Отже, результатом виконання Дії 5 буде частина програми користувача, яка зображена нижче:




### Дія 6.

Третя умова завдання у цілому подібна попереднім, але є додаткова умова. Потрібно, щоб вихід залишався активованим після однократного натискання на кнопку після її відпускання. Але у цьому випадку обов'язково потрібно передбачити вимикання виходу. Інакше, цей вихід ніколи не буде вимкненим, за виключенням перезавантаження ПЛК.

Отже, спочатку зробить копіювання та додавання ще одного ланцюгу. Далі зробить його редагування шляхом заміни адрес операндів. Проте, щоб виконати додаткову умову, тобто фіксування результату вмикання виходу, потрібно реалізувати «реле з підтягненням». Це найпростіший спосіб програмної реалізації реле з додатковим контактом. Функцію додаткового контакту виконує операнд типу «контакт», але з адресою виходу, який фіксується. Для цього потрібно паралельно операнду X2 до ланцюга додати операнд типу «NO контакт» із адресою Y2. Побудова паралельної гілки створюється за допомогою додаткових

інструментів. Це піктограми  для додавання вертикальних та горизонтальних ліній. Але можливо використати піктограму .

Для вимикання виходу Y2 необхідно до ланцюгу додати ще один компонент – NC контакт адресою X0. Для цього за допомогою піктограми  достатньо його додати після з'єднання паралельних гілок ланцюгу.

Результат виконаних дій зображений нижче:



**Дія 7.**

Четверта умова завдання подібна до третій, тому скористаємося буфером обміну для копіювання ланцюга, створеного раніше. Подібно до попереднього пункту (Дія 5) змінимо адреси задіяних операндів.

Надамо пояснення щодо використання статусу термоперемикача у ланцюгу вимикання нагрівача. Логіка роботи ланцюга така. У нормальному стані термоперемикач знаходиться у замкненому стані. Якщо буде ввімкнутий вихід ПЛК, до якого підключений нагрівач, то поступово температура всередині ОУ підвищиться. Якщо температура досягне рівня перемикачання (на стенді встановлений термоперемикач з  $T_{пер}=40^{\circ}C$ ), то його внутрішні контакти розімкнуться. Таким чином, якщо у ланцюг керування виходом Y3 додати операнд типу «NC контакт» (X4), то він буде розмикати логічний ланцюг вимикання нагрівача, якщо температура всередині ОУ досягне значення, більшого за  $T_{пер}=40^{\circ}C$ .

Результат додавання наступного ланцюга зображений нижче:



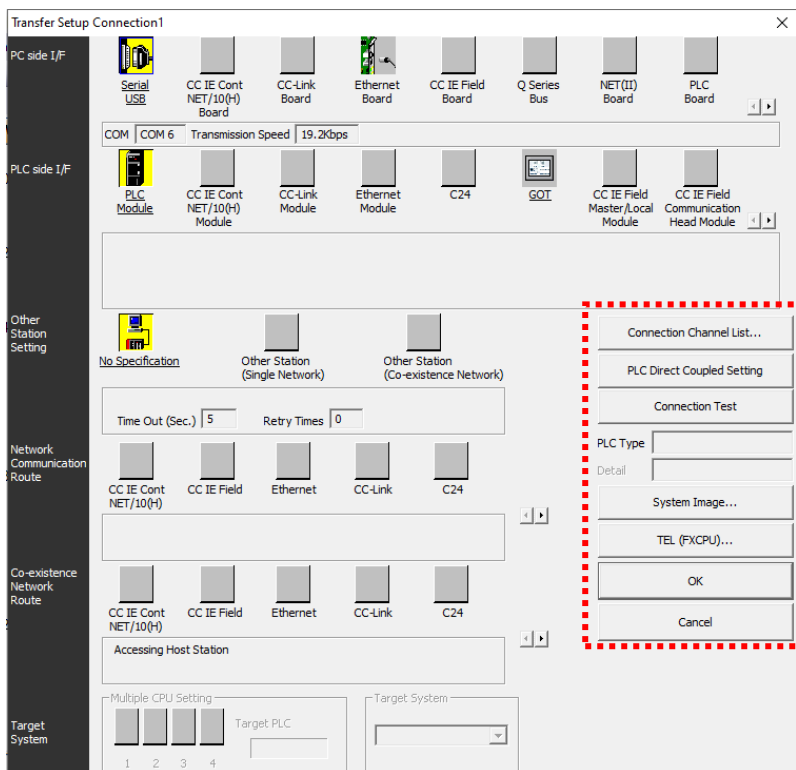
Зауважимо, що цей приклад має навчальний характер і лише є завданням для ознайомлення із принципами розроблення програми користувача та знайомства з інтерфейсом користувача ПЗ GX\_W2.

#### 1.4.3 Підключення до ПЛК та завантаження до нього коду проекту

Завантаження коду готових проектів до ПЛК серії MELSEC FX можливе за допомогою лише послідовного інтерфейсу. Але в ПЛК послідовний інтерфейс має рівень *TTL*, тому його потрібно узгодити з рівнем інтерфейсу *COM*-порта ПК за допомогою адаптеру рівнів інтерфейсів *TTL/RS232*. Додатково потрібно використати кабель-подовжувач типу *RS232(M)--RS232(F)*, який входить до комплекту стенда. Якщо використовується *USB*-порт ПК або ноутбука, то потрібно використати перетворювач інтерфейсів *RS232/USB*. Саме така схема комунікації застосована для підключення ПЛК до ПК у навчальних стендах. Але використання інтерфейсу *USB* потребує додаткових налаштувань ОС *Windows*. Це потребує встановлення на ПК драйвера перетворювача інтерфейсів в ОС *Windows*. За допомогою цього драйверу емулюється віртуальний *COM*-порт у диспетчері пристроїв ОС *Windows*.

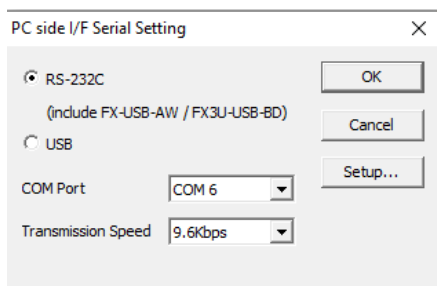
*Алгоритм налагодження зв'язку з ПЛК з боку ПК.*

Для взаємодії з ПЛК в середовищі GX\_W2 використовують команди, які згруповані у переліку меню *Online*. Для налаштування каналу зв'язку з ПЛК ЛКМ виберіть вкладення *Connection Destination* у вікні *Navigation* та виберіть ПЛКМ *Connection1* із переліку *Current Connection*. У результаті відкриється вікно *Transfer Setup Connection1*. Вікно налаштувань зв'язку з ПЛК зображено нижче та для зручності розділено на декілька зон:



Зона **PC side I/F** призначена для налаштувань послідовного інтерфейсу з боку ПК. Середовище **GX\_W2** автоматично за допомогою ОС **Windows** визначає наявність існуючих інтерфейсів для зв'язку з ПЛК. Наприклад, це буде віртуальний **COM**-порт, який згенерований ОС **Windows** та позначений піктограмою **Serial USB**. Якщо драйвер встановлений вірно, то у нижньому рядку зони **PC side I/F** вказано номер віртуального **COM**-порта та швидкість обміну даними. У випадку першого налаштування потрібно ПЛКМ на піктограму **Serial USB**. Тоді відкриється вікно налаштувань послідовного порту, як це зображено нижче:





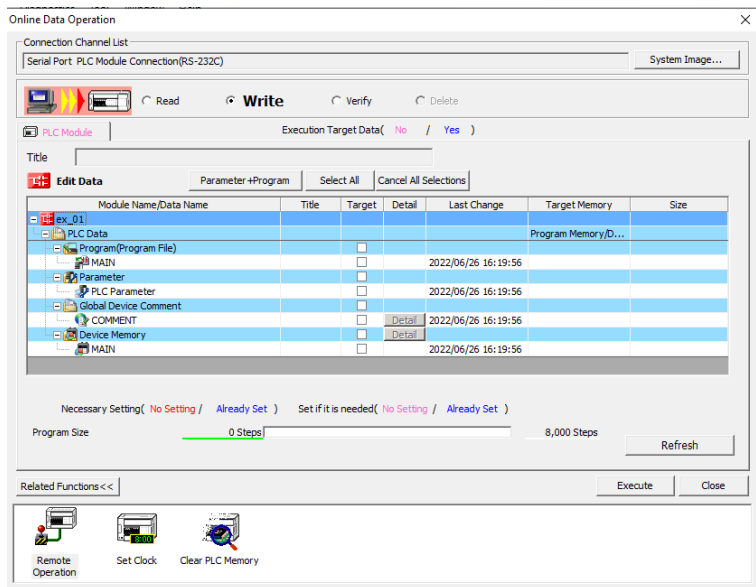
Необхідно вказати номер *COM*-порта, який згенерований ОС *Windows* та швидкість обміну даними. Якщо потрібно налаштувати формат кадру, то ЛКМ натисніть на кнопку **Setup...** . Параметри налаштувань такі: швидкість передавання **Transmission Speed** (9,6 Kbps), формат кадру – кількість біт **Data Bits** (7 біт), перевірка на парність **Parity (Even)** та кількість стоп-біт **Stop Bits** (1 біт). Введіть вказані значення параметрів та підтвердить їх ЛКМ натиснувши на кнопку **OK**.

Наступна зона – це зона налаштування зв'язку з боку ПЛК *PLC side I/F*. Якщо тип та модель ПЛК визначені, то він відображується у вигляді піктограми PLC module. Додаткові налаштування тут не потрібні.

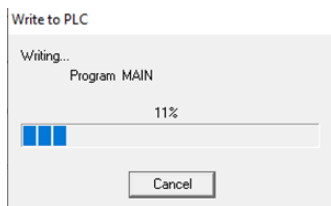
Інші зони вікна **Transfer Setup Connection1** зараз не потрібні, тому вони не розглядаються.

На зображенні вікна **Transfer Setup** червоним пунктиром позначена зона з кнопками. Перевірити зв'язок з ПЛК можна за допомогою кнопки **Connection test**. У випадку вдалої перевірки ПЗ *GX\_W2* видасть відповідне повідомлення. У протилежному випадку ПЗ *GX\_W2* видасть рекомендації для усунення проблеми. Отже, якщо параметри зв'язку встановлені вірно та тест позитивний, то можна ЛКМ закрити вікно **Transfer setup** за допомогою кнопки **OK**.

Зараз можна завантажити код програми до ПЛК. Для цього необхідно із переліку меню **Online** використати команду **Write to PLC ...** . В результаті відкриється вікно **Write to PLC**:



У цьому вікні потрібно ЛКМ позначити компоненти дерева проекту, які потрібно завантажити. Це, такі компоненти: Program (MAIN), Parameter (PLC parameter), Global Device Comment (COMMENT) та Device Memory (MAIN). Якщо ЛКМ натиснути на кнопку Execute, то ПЗ GX\_W2 зробить запит на зупинення виконання програми, якщо в контролері раніше вона була завантажена та запущена на виконання. Якщо підтвердити операцію завантаження, то середовище зупинить ПЛК та почне процес завантаження. Нижче зображено вікно із плинном процесу завантаження програм до ПЛК:



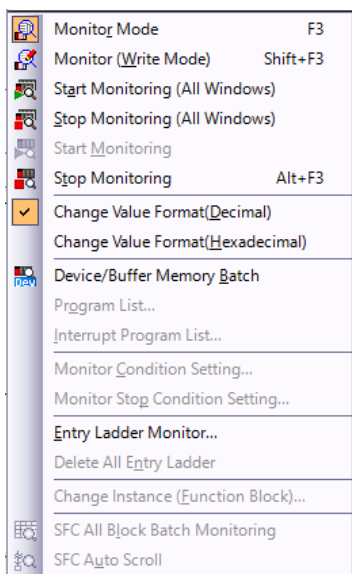
Після завершення процесу завантаження коду програми ПЗ GX\_W2 зробить запит на її запуск. Після підтвердження ПЛК перемкнеться у режим виконання програми користувача. Далі, для закінчення процесу завантаження потрібно натиснути ЛКМ на кнопку Close.

#### 1.4.4 Взаємодія з ПЛК в режимі ONLINE

Отже, код програми користувача завантажений до ПЛК. Завершальним етапом розроблення проекту для системи автоматизації на реальному об'єкті вважається перевірка роботи СУ в режимі Online.

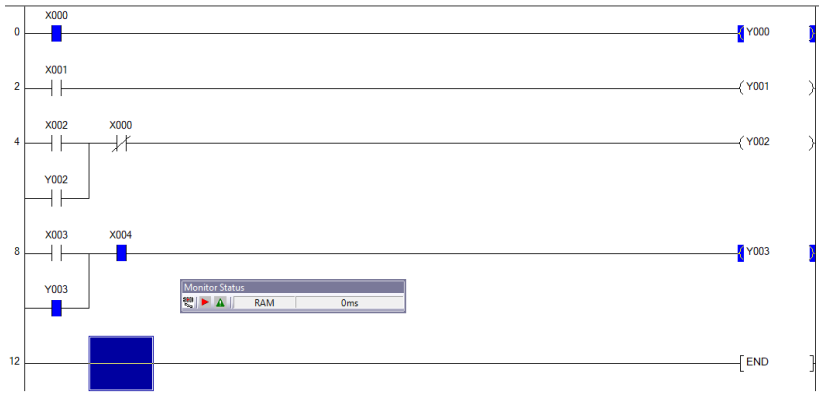
В режимі Online можливі два способи взаємодії з ПЛК: спостереження за станом операндів (Monitor) та налагодження програми із введенням значень операндів (Debug).


Усі команди режиму Monitor з переліку Online зображені нижче:

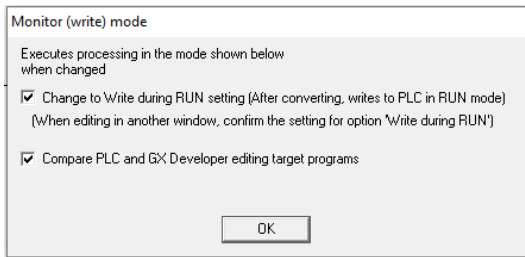


Команди вибору режимів взаємодії з ПЛК дублюються піктограмами на панелі інструментів Ladder. Для спостереження за станом операндів за допомогою ФК F3 ввімкніть режим Monitor. В результаті

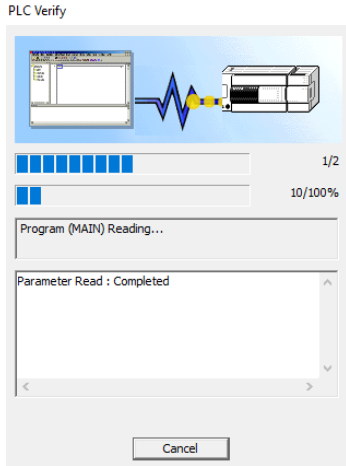
отримаємо робочий простір ПЗ GX\_W2 у режим Monitor, який зображено нижче:



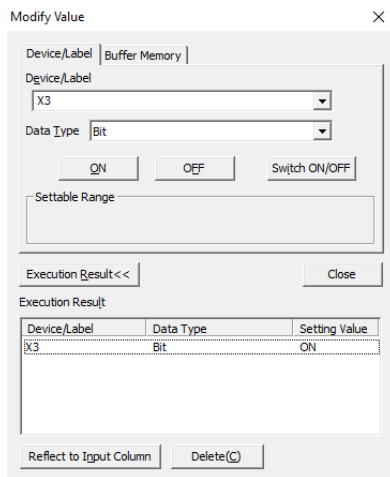
Якщо потрібно перевірити реакцію СУ та ОУ на примусове присвоєння значення будь-якому операнду, то необхідно за допомогою  перемкнути ПЗ GX\_W2 у режим запису (або Shift+F3). Середовище GX\_W2 відповість запитом про дозвіл вносити зміни, якщо ПЛК у режимі RUN та дозвіл на порівняння код проєкту в ПЛК і в середовищі (за умовчанням обое запити позначені):



Підтвердить згоду на запит натисканням ЛКМ на кнопку ОК. Далі відкриється вікно PLC Verify, в якому буде відображатися плин процесу звіряння коду у ПЛК та коду у середовищі. Вікно плин процесу верифікації зображено нижче:



Для примусового вмикання операнда із середовища GX\_W2 необхідно у переліку меню Online ЛКМ натиснути на команду Modify Value... . Відкриється вікно Modify Value, яке зображене нижче:



На зображенні заповнені поля для зміни стану операнда X3. Кнопки ON, OFF та Switch ON/OFF виконують потрібні дії щодо стану бітових операндів.

Для примусового встановлення бітового операнда використовують вкладення **Device/Label**. До поля **Device** необхідно додати потрібний операнд. Наприклад, для вмикання нагрівача потрібно додати операнд **X3**. Далі, потрібно ЛКМ натиснути на кнопку **ON**. Якщо потрібно імітувати роботу перемикача (кнопки з фіксуванням), то використовують кнопку **Switch ON/OFF**.

У подальшому, якщо потрібно записати значення до регістрів зберігання використовують вкладення **Device/Label** із типом даних **Word[Signed]** у полі **Data Type**.

Особливість цього режиму полягає у тому, що якщо вимкнути режим примусового запису, то усі записи втрачаються.

Насамкінець, зауважимо, що абсолютні адреси операндів можливо замінити символьними іменами. Для цього потрібно заповнити відповідну таблицю імен. Проте, створення таблиці символьних імен можливо, якщо у проєкті під час його створення поставити позначку **Use Label**.

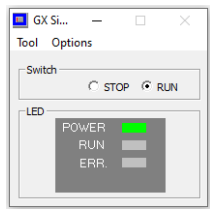
#### *1.4.5 Порядок використання симулятора ПЛК*

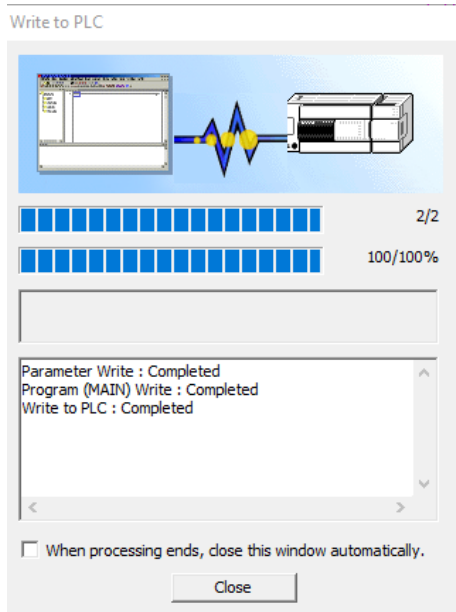
Якщо завдання виконується самостійно, тобто без використання стенда із реальним ПЛК, у середовищі **GX\_W2** можливе налагодження проєкту на симуляторі. Це окрема програма, яка запускається у середовищі **GX\_W2**, працює у фоновому режимі. Симулятор повністю імітує роботу реального ПЛК.

Отже, у разі самостійного виконання завдання скористайтесь симулятором роботи ПЛК. Для цього у переліку меню **Debug** натисніть ЛКМ на команду **Start/Stop Simulation**, або ЛКМ виберіть піктограму

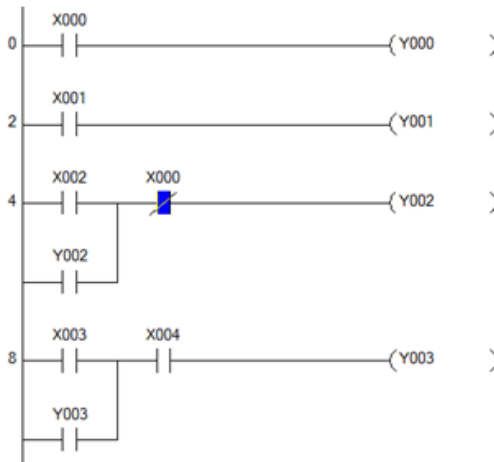


на панелі інструментів **Program Common**. У результаті одночасно відкриється вікно керування станом віртуального контролера та вікно завантаження коду проєкту до симулятора, які зображені нижче:





Закрийте вікно Write to PLC натисненням ЛКМ на кнопку Close.  
 Далі відкриється вікно із програмою користувача у режимі Online.



Для примусового вмикання операнда або передавання йому значення із середовища GX\_W2 необхідно також скористатись командою Modify Value... у переліку меню Online, як це зазначено у попередньому пункті.

### 1.5 Перевірка роботи навчального завдання

1) Перевірте роботу програми користувача на стенді або за допомогою симулятора. Для цього вмикайте кнопки або примусово встановлюйте значення операндів та спостерігайте за всіма змінами у програмі користувача. Зробіть декілька скриншотів робочого простору ПЗ GX\_W2 у режимі моніторингу або у разі використання симулятора ПЛК.

2) Складіть звіт у електронному вигляді відповідно до правил оформлення звітів: відомості про виконавця, назву та мету роботи, схему з'єднань, лістинг (зображення) програми користувача та скриншоти екранів у режимі Online. Для документування проекту використайте вбудовану в середовище GX\_W2 відповідну утиліту, яка активується натисненням ЛКМ на команди Print... в переліку меню Project.

### 1.6 Завдання для самостійної роботи

1) На основі навчального прикладу розробити програму користувача, в якій автоматично перемикаються нагрівач та охолоджувач в залежності від стану термоперемикачів, тобто температури всередині ОУ.

2) На основі навчального прикладу розробити програму користувача, в якій умовою вмикання та вимикання вихідних елементів будуть логічні вирази, які наведені в табл. 1.1. При цьому необхідно скласти таблицю істинності, тобто визначити необхідний стан входів для вмикання виходу.

Таблиця 1.1 – Варіанти умов для вмикання вихідних елементів

№	Логічна умова ( $X1 \dots X6$ – дискретні входи)	Дискретний вихід ( $Y$ )
1	$(X1 \text{ AND } X3) \text{ OR } (X4 \text{ AND } X5 \text{ AND } \text{NOT } X6)$	$Y0$



2	$(X2 \text{ OR } X3) \text{ AND } (\text{NOT } X1 \text{ AND } X5 \text{ AND } X6)$	Y1
3	$(X1 \text{ AND } X3) \text{ OR } X4 \text{ AND } \text{NOT } X5 \text{ AND } X6)$	Y2
4	$(X2 \text{ AND } X3) \text{ XOR } (X1 \text{ AND } X5 \text{ AND } X6)$	Y3
5	$(\text{NOT } X1 \text{ XOR } X3) \text{ AND } (X4 \text{ OR } X5 \text{ OR } X6)$	Y0
6	$(X2 \text{ AND } \text{NOT } X3) \text{ OR } (X4 \text{ AND } X5 \text{ AND } X6)$	Y1
7	$X2 \text{ AND } X3 \text{ AND } \text{NOT } X4 \text{ AND } (X1 \text{ XOR } X6)$	Y2
8	$(X1 \text{ OR } X3 \text{ OR } X4) \text{ AND } (\text{NOT } X5 \text{ OR } X6)$	Y3

3) Розробіть програму користувача для дискретного управління нагрівачем. В програмі користувача необхідно реалізувати такі функції:

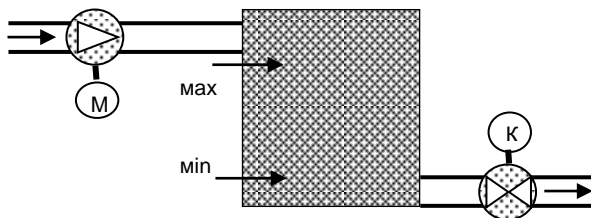
- вмикання сигналізації при виникненні будь-якої з аварій на ОУ;
- вмикання об'єкта за допомогою кнопки, за умови відсутності аварій;
- вимикання об'єкта при виникненні будь-якої з аварій або за допомогою кнопки.

Вмикання живлення нагрівача TEN та лампи сигналізації LAMP здійснюється залежно від результату логічних операцій (РЛО) над чотирма вхідними дискретними сигналами: PUSK, STOP, AVAR та POGAR. У разі спрацьовування AVAR та POGAR вмикається індикатор червоного кольору. Нагрівач повинен бути ввімкненим до моменту натиснення на кнопку STOP або до спрацьовування датчиків AVAR або POGAR. В табл. 1.2 наведено варіанти призначення входів та виходів ПЛК.

Таблиця 1.2 – Варіанти призначення входів та виходів ПЛК

№	PUSK	STOP	AVAR	POGAR	TEN	LAMP
1	X0	X1	X2	X4	Y3	Y0
2	X0	X1	X3	X5	Y3	Y0

4) Розробіть програму користувача для дискретного керування об'єктом типу «ємність для накопичення та зберігання», яка реалізує алгоритм автоматичного керування установкою водопостачання. Схема установки водопостачання зображена нижче:



Умови завдання:

Об'єктом керування є ємність для накопичення води, яка наповнюється із свердловини за допомогою насоса (nasos), який встановлений на вхідному трубопроводі. Вода витрачається для орошення. В ємності встановлено два контактних датчика, які визначають мінімальний (min) та максимальний рівні (max). Датчики рівня поплавкового типу підключені по схемі NC, тобто, нормально-замкнений контакт. Це означає, якщо ємність порожня, то вони у замкненому стані, але, якщо рівень води в ємності поступово збільшується, вони по черзі розмикаються, спочатку датчик мінімального рівня, потім – датчик максимального рівня. У вихідному трубопроводі встановлений електромагнітний клапан (klapan).

Якщо контакти датчика min та max замкнені, то ємність порожня. Автоматично вмикається насос та вимикається клапан. Якщо контакти датчика min та max розімкнуті, то ємність заповнена: автоматично вимикається насос, та вмикається клапан. Якщо контакти датчика max розімкнені, контакти датчика min замкнені (рівень в ємності середній), то статус клапану та насосу не змінюється.

### 1.7 Контрольні питання

1) Наведіть коротку характеристику контролерів MELSEC серії FX від компанії *Mitsubishi Electric Corp.*

2) Який склад та можливості програмного забезпечення GX Works2/3 від компанії *Mitsubishi Electric Corp.*?

3) З яких компонентів та ресурсів складається проєкт у середовищі GX Works2?

4) Які фізичні інтерфейси використовують для завантаження проєкту до контролерів MELSEC серії FX?

*Практичне завдання 2*  
**ОБРОБЛЕННЯ АНАЛОГОВИХ СИГНАЛІВ  
В КОНТРОЛЕРАХ MELSEC СЕРІЇ FX**

**2.1 Мета завдання**

- ознайомлення з основними принципами оброблення аналогових сигналів в контролерах MELSEC серії FX різних модифікацій у середовищі GX Works2;
- закріплення теоретичних знань щодо оброблення аналогових сигналів в контролерах MELSEC серій FX різних модифікацій;
- організація взаємодії з індикаторним модулем.

**2.2 Порядок виконання завдання**

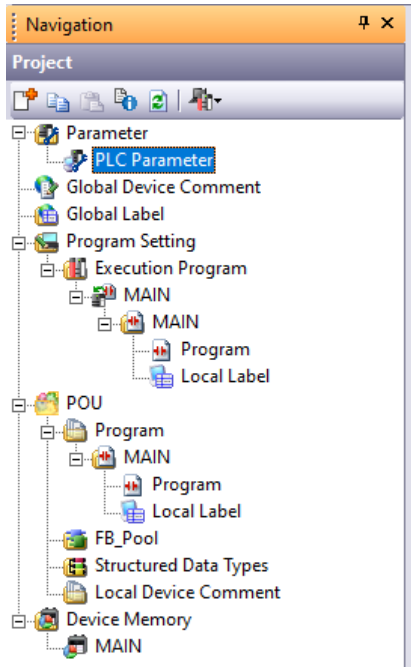
Виконання завдання розраховано на 2 ак. год. та складається з таких етапів:

- 1) Створення проєкту та розроблення програми користувача з обробленням аналогових вхідних та вихідних сигналів.
- 2) Налаштування відображення значення аналогового вхідного сигналу та введення значення аналогового вихідного на індикаторному модулі.
- 3) Створення програмного організаційного компоненту (POU) типу функціональний блок (FB).

**2.3 Хід виконання завдання**

*2.3.1 Розроблення проєкту для ознайомлення з принципами оброблення аналогових сигналів*

Проведіть підготовчі роботи відповідно до дій, які описані у *Практичному завданні 1* щодо створення проєкту у ПЗ GX\_W2. Ім'я проєкту призначить таке: ex\_02. Але тип проєкту оберіть **Structured Project**. Для цього типу проєкту позначка **Use Label** встановлена за умовчанням. Це означає, що можливе використання символічних імен операндів. Нижче зображено вміст компонентів проєкту ex\_02:



### 2.3.2 Розроблення програми користувача на першому етапі

Розробимо програму користувача для оброблення аналогових сигналів. Адреси спеціальних регістрів АЦП та ЦАП, які зберігають код перетворених аналогових сигналів визначені у Дод. А.1.

*Навчальне завдання.*

1. Зчитати значення коду в АЦП контролера та передати його до регістру загального зберігання.

2. Записати значення коду від регістру загального зберігання в ЦАП контролера.

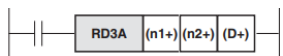
*Алгоритм розроблення програми користувача.*

Необхідно додати новий рядок для створення ланцюга у програми користувача, як це було зроблено у *Практичному завданні 1*.

Для оброблення аналогових сигналів в контролерах FX1N засто-

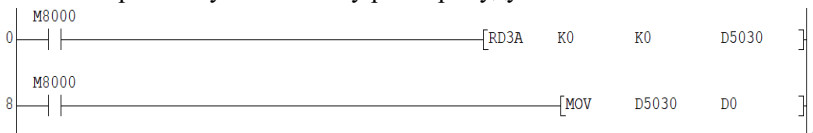
совано спеціальних механізм читання та запису значень кодів АЦП та ЦАП (обидва перетворювача мають розрізнення 12 біт) із спеціальних регістрів пам'яті, які безпосередньо пов'язані з АЦП та ЦАП контролера. Для обох дій (читання та записування) в ПЗ GX\_W2 передбачено дві спеціальних інструкції. Це інструкція RD3A (FNC176) – для читання значення аналогового входу та інструкція WR3A (FNC177) – для записування значення до аналогового виходу. Для ознайомлення з правилами використання цих інструкцій необхідно звернутися до технічної документації [4 - мануал].

Інструкція читання RD3A має такий вигляд:

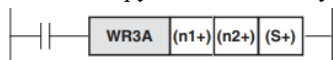


. Надамо пояснення щодо цієї інструкції. Після назви інструкції розміщені її параметри: (n1+), (n2+) та (D+). Перші два параметра визначають адресу модуля (K0...K7) та номер каналу (K0...K2). У випадку застосування клона контролера FX1N (green) обидва ці параметри дорівнюють нулю. Третій параметр – це адреса спеціального регістра, де зберігається код АЦП аналогового входу. Для FX1N (green) адреса цього регістру – це D5030. Для FX1N (red) адреса цього регістру – це D5000. Для того, щоб обробляти отримані кодові значення потрібно передати їх у звичайні регістри пам'яті. Крім того, щоб інструкція виконувалась кожен цикл, потрібно використати спеціальний меркер M8000, який зберігає статус ПЛК на кожний цикл сканування. Це також є особливістю механізму читання значення аналогового входу.

Таким чином, інструкції читання значення аналогового входу ПЛК та зберігання у звичайному регістрі будуть такими:



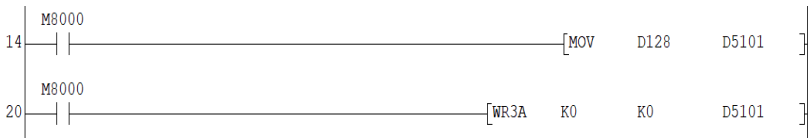
Інструкція запису WR3A має такий вигляд:



. Надамо пояснення щодо цієї інструкції.

Після назви інструкції розміщені її параметри: (n1+), (n2+) та (S+). Перші два параметра визначають адресу модуля (K0...K7) та номер каналу (K0...K1). У випадку застосування клону контролера FX1N обидва ці параметри дорівнюють нулю. Третій параметр – це адреса спеціального регістра, де зберігається код АЦП аналогового входу. Для FX1N адреса цього регістру – це D5101. Для того, щоб сформувати вихідний сигнал необхідно передати значення від звичайного регістра до регістра D5101. Крім того, щоб інструкція виконувалась кожен цикл, потрібно використати спеціальний меркер M8000, який зберігає статус ПЛК на кожен цикл сканування. Це також є особливістю механізму запису значення аналогового виходу.

Таким чином, інструкції передавання значення коду до ЦАП будуть такими:



### 2.3.3 Розроблення програми користувача на другому етапі

Розробимо програму користувача для оброблення аналогових сигналів та взаємодії з індикаторним дисплеєм.

*Навчальне завдання.*

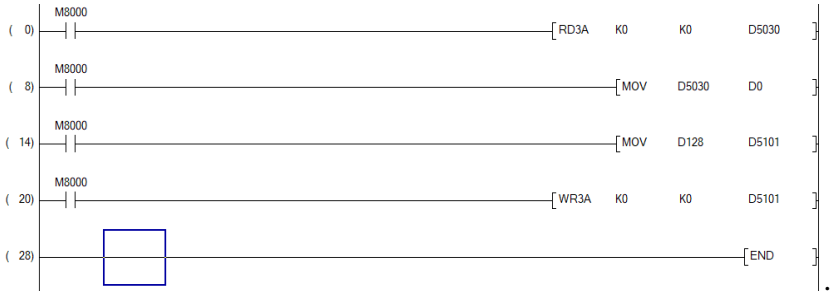
1. Зчитати значення коду в АЦП контролера та передати його до регістру загального зберігання з подальшим відображенням на дисплеї.
2. Записати значення коду на модулі дисплея до регістру загального зберігання з подальшим перетворенням у ЦАП контролера.

*Алгоритм розроблення програми користувача.*

Для вирішення завдань другого етапу можливо використати проєкт з попереднього завдання. Але, його потрібно частково змінити. Для цього спочатку збережіть проєкт з новим ім'ям, наприклад, це ім'я буде ex\_02\_hmi.

Принципи використання дисплейного модуля та адреси його регістрів загального застосування визначені у Дод. А.2.

Програма користувача у цьому випадку буде такою:




Насамкінець, розглянемо процедуру перетворення коду ЦАП, яке відповідає вихідному аналоговому сигналу, на значення фізичного параметру (напруга). Для цього визначимо початкові умови. Виконавчий механізм аналогової дії, наприклад, засувка у системі вентилявання керується аналоговим сигналом 0...3,3 V постійного струму. Це відповідає ступеню відкриття 0...100 %. Наприклад, якщо потрібно відкрити засувку на 35 %, то сигнал керування має бути 1,15 V. Код ЦАП контролера із 12-бітним розрізненням буде дорівнювати значенню 1433. Взагалі, інтервал ЦАП буде розраховуватись так:  $3,3V/4095=0,8mV$ .

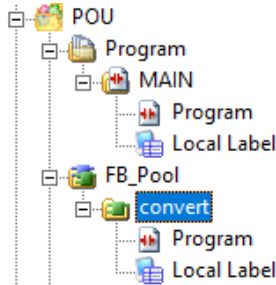
### 2.3.4 Створення функціонального блока для оброблення аналогових сигналів

Розробимо програму користувача з функціональним блоком для оброблення аналогових сигналів та взаємодії з індикаторним дисплеєм. Навчальне завдання буде подібне до завдання у п.2.3.3. на стор. 36.

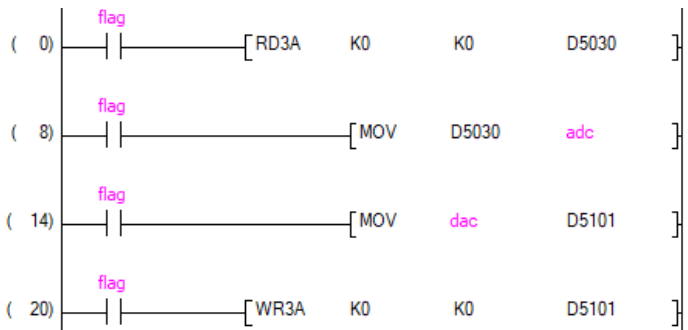
#### Алгоритм розроблення програми користувача.

Для вирішення завдань третього етапу можливо використати проєкт із попереднього завдання. Але, його потрібно частково змінити. Для цього спочатку збережіть проєкт з новим ім'ям, наприклад, це ім'я буде ex\_02\_fb.

Спочатку у менеджері проєкту ЛКМ та піктограми  необхідно додати новий об'єкт – функціональний блок (ФБ). Мову для програмування ФБ оберіть **Ladder**. Нехай ім'я ФБ буде таким – **convert**. Отже, отримаємо у дереві проєкту новий об'єкт, як це зображено нижче:

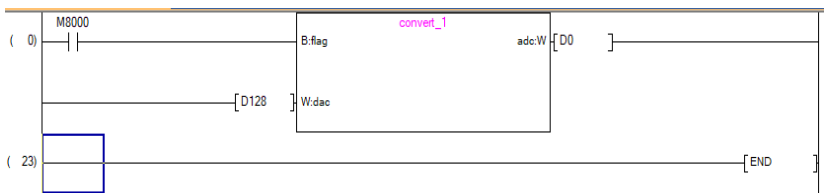


Далі необхідно заповнити ФБ функціями читання та запису аналогових сигналів. Але, на відміну від попереднього прикладу, використайте змінну **flag** типу **Bit** для постійного перерахунку кодів АЦП та ЦАП. Операнди для зберігання значення коду АЦП (вихідне значення) та ЦАП (вхідне значення) призначить відповідно **adc** та **dac**. Тип даних **adc** та **dac** буде **Word[Signed]**. Усі зазначені операнди мають локальний характер. У процесі додавання операндів середовище перемикається у вікно відображення операндів **Local Label**. Нижче зображено склад ФБ для перетворення аналогових сигналів:





Наступним кроком потрібно у основної програмі з ім'ям MAIN додати створений ФБ. Крім того потрібно параметри ФБ з'єднати із потрібними операндами, як це зображено нижче:



Для додавання операндів потрібно ЛКМ вибрати піктограми контакту для з'єднання із флагом M8000 та вставки інструкції для з'єднання із регістрами загального призначення D0 та D128.

На завершення виконання цього завдання потрібно скопіювати проект, зберегти та завантажити до ПЛК. У разі самостійного виконання завдання для налагодження проекту скористайтесь симулятором роботи ПЛК, як це наведено раніше у *Практичному завданні 1* (див. стор. 28).

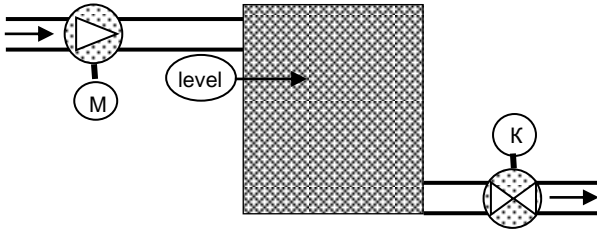
## 2.4 Перевірка роботи навчального проекту

1) Перевірте роботу програми користувача на стенді або за допомогою симулятора. Для цього вмикайте кнопки модуля аналогових сигналів або примусово встановлюйте значення операндів та спостерігайте за всіма змінами у програмі користувача. Зробіть декілька скриншотів робочого простору ПЗ GX\_W2 у режимі моніторингу.

2) Складіть звіт у електронному вигляді відповідно до правил оформлення звітів: відомості про виконавця, назву та мету роботи, схему з'єднань, лістинг (зображення) програми користувача та скриншоти екранів у режимі Online.

## 2.5 Завдання для самостійної роботи

1) Розробіть програму користувача для дискретного керування об'єктом типу «ємність для накопичення та зберігання», яка реалізує алгоритм автоматичного керування установкою водопостачання. Схема установки водопостачання зображена нижче:



*Умови завдання:*

Об'єктом керування є ємність для накопичення води, яка наповнюється із свердловини за допомогою насоса (nasos), який встановлений на вхідному трубопроводі. Вода витрачається для зрошення. В ємності встановлено датчик рівня (level). Датчик рівня формує сигнал напруги  $0 \dots 10 \text{ V}$  постійного струму. У вихідному трубопроводі встановлений електромагнітний клапан (klapan).

Якщо сигнал датчика менше  $3 \text{ V}$ , то вмикається насос та вимикається клапан. Якщо сигнал датчика більше  $7 \text{ V}$ , то вимикається насос та вмикається клапан. Якщо сигнал датчика знаходиться у діапазоні  $3 \dots 7 \text{ V}$  то статус клапану та насосу не змінюється.

2) Для першого завдання для самостійної роботи передбачити зміну діапазону з боку індикаторного модуля.

## 2.6 Контрольні питання

1) Яким чином реалізовано оброблення аналогових сигналів у контролерах MELSEC серії FX різних модифікацій у середовищі GX Works2.

2) Як реалізована взаємодія контролера та індикаторного модуля щодо обміну значеннями регістрів.

### *Практичне завдання 3*

## ОСНОВНІ ПРИНЦИПИ ВИКОРИСТАННЯ ЛІЧИЛЬНИКІВ ТА ТАЙМЕРІВ У КОНТРОЛЕРАХ MELSEC СЕРІЇ FX

### **3.1 Мета завдання**

- ознайомлення з основними принципами використання таймерів та лічильників у контролерах MELSEC серії FX різних модифікацій у середовищі GX Works2;
- закріплення теоретичних знань щодо використання таймерів та лічильників в контролерах MELSEC серій FX різних модифікацій у середовищі GX Works2;
- організація взаємодії з індикаторним модулем.

### **3.2 Порядок виконання завдання**

Виконання завдання розраховано на 2 ак. год. та складається з таких етапів:

- 1) Створення проєкту та розроблення програми користувача із лічильниками.
- 2) Створення проєкту та розроблення програми користувача із таймерами.
- 3) Налаштування відображення поточного значення лічильника та введення значення параметру лічення та інтервалу часу на індикаторному модулі.


### **3.3 Хід виконання завдання**

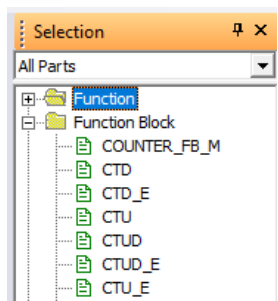
*3.3.1 Розроблення проєкту для ознайомлення з принципами використання лічильників*

Проведіть підготовчі роботи відповідно до дій, які описані у *Практичному завданні 1* щодо створення проєкту у ПЗ GX\_W2. Тип проєкту оберіть **Structured Project**. Ім'я проєкту призначить таке: **ex\_03\_counter.gxw** .

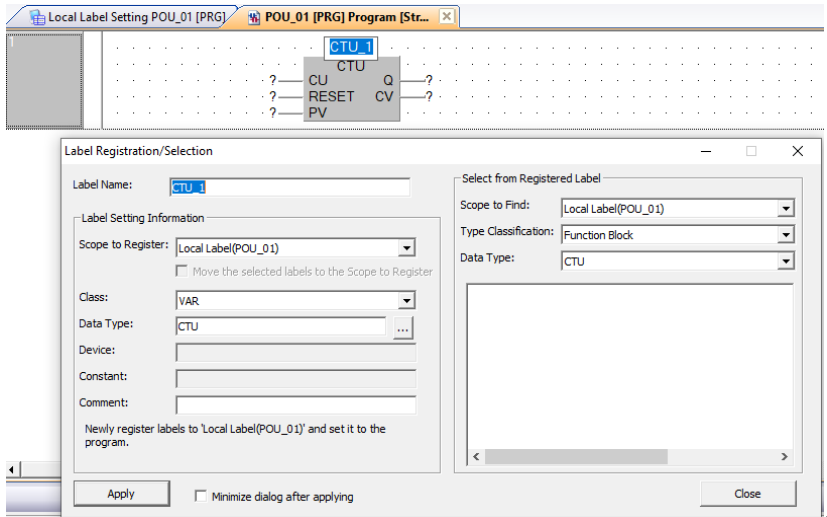
Використання лічильників у ПЗ GX\_W2 відповідає стандарту

*IEC61131*. Розглянемо порядок використання лічильників на прикладі застосування контролера FX1N, який входить до обладнання стенду. Взагалі, у контролерах MELSEC можлива реалізація стандартних лічильників (*IEC61131*) програмним способом або апаратних лічильників. Апаратні лічильники мають обмеження по їхньої кількості та діапазону обчислення в залежності від моделі контролера. Наприклад, контролер FX1N має 16- та 32-бітові апаратні лічильники загального призначення та лічильники із пам'яттю. Апаратні лічильники у контролері FX1N мають такі адреса: інкрементуючі 16-бітні C0...C15 загального призначення та C16...C199 із пам'яттю для лічення та двонаправлені 32-бітні – C200...C219 загального призначення та C220...C234 із пам'яттю для лічення. У контролері FX1N є також швидкісні входи для лічення до 5 кГц.

Розглянемо програмну реалізацію стандартних лічильників програмним способом. Отже, ЛКМ за допомогою піктограми  відкрийте додаткове вікно Function Block Selection Window. Зовнішній вигляд вікна з відкритим переліком функціональних блоків із стандартної бібліотеки зображений нижче:



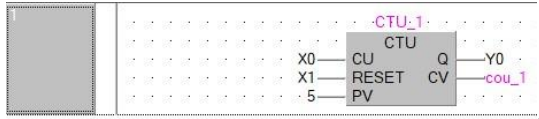
Далі, якщо ЛКМ натиснути на піктограму потрібного функціонального блока, наприклад, блока інкрементного лічення (CTU), та не відпускаючи ЛКМ перенести до робочого простору, отримаємо такий результат:



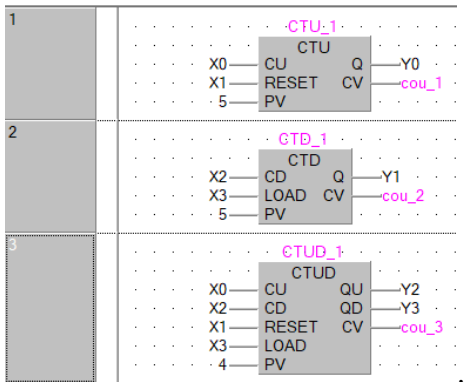
Тобто, одночасно з додаванням блока відкриється вікно **Label Registration/Selection** з його налаштуваннями. За умовчанням, ім'я (Label Name), область зберігання (Scope to Register), клас (Class) та тип (Data Type) блоку надасть середовище, як це зображено вище.

Далі потрібно натиснути клавішу **Enter** для підтвердження запропонованого імені **CTU\_1**. Рамка маркера, яка підсвічена синім кольором, перемкнеться на перший вхід блоку (**CU**). Це лічильний вхід, на який надходять імпульси від давачів. Давачем може бути, наприклад, будь-який безконтактний давач. Для випадку застосування стенда це буде вхід **X0**. Замість символу «?» введіть цей операнд. Підтвердить свій вибір натисканням на клавішу **Enter**. Другий вхід блоку (**RESET**) призначений для скидання лічильника. Нехай це буде кнопка операнду з адресою **X1**. Третій вхід блоку (**PV**) призначений для встановлення параметра лічення. Цей вхід, на відміну від попередніх входів типу **BOOL**, має тип **WORD**. Тут можливо ввести константу або операнд. Нехай це буде константа. Наступне натискання на клавішу **Enter** перемкне рамку на виходи блоку. Перший вихід (**Q**) – це індикатор стану лічильника. Нехай про досягнення параметру лічення (**PV**) сигналізує вихід **Y0**.

Другий вихід (CV) інформує про поточне значення параметру лічення. Цей вихід потрібно зв'язати з операндом типу WORD. Нехай ім'я цього операнду буде таким: cou\_1. Для цього операнда потрібно змінити його тип (WORD). Після усіх проведених дій буде створено перший ланцюг, який зображений нижче:



Подібно до наведеного вище додайте та налаштуйте декрементний лічильник (CTD) та комбінований (двонаправлений) лічильник (CTUD). Для лічення в зворотному напрямку використайте операнд X2 та операнд X3 (для завантаження параметру відліку). Параметр лічення для зворотного відліку зв'яжіть з операндом cou\_2. Комбінований лічильник CTUD\_1 буде використовувати операнди лічильників CTU\_1 та CTD\_1. Нижче зображено фрагмент програми користувача із усіма типами лічильників:




На завершення виконання цього завдання потрібно проект скопіювати, зберегти на жорсткий диск та завантажити до ПЛК. У разі самостійного виконання завдання для налагодження проекту скористайтесь симулятором роботи ПЛК, як це наведено раніше (див. стор. 28).

### 3.3.2 Розроблення проєкту для ознайомлення з принципами використання таймерів

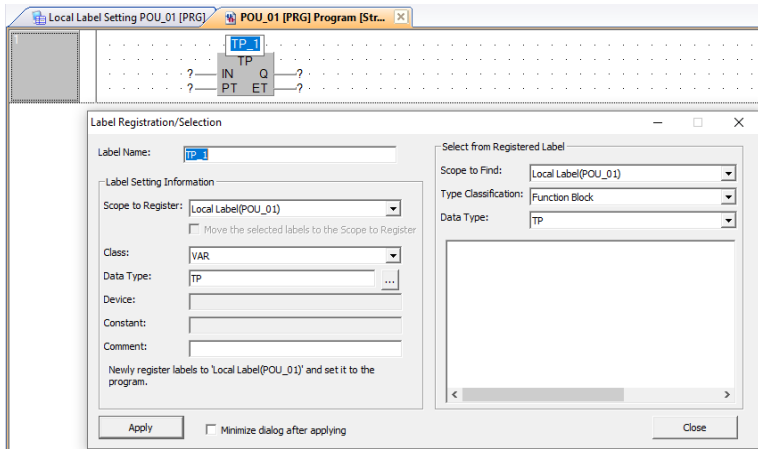
Проведіть підготовчі роботи відповідно до дій, які описані у *Практичному завданні 1* щодо створення проєкту у середовищі GX\_W2. Тип проєкту оберіть **Structured Project**. Ім'я проєкту призначить таке: **ex\_03\_timer.gxw**.

Використання таймерів у ПЗ GX\_W2 відповідає стандарту IEC61131. Розглянемо порядок використання таймерів на прикладі застосування контролера FX1N, який входить до обладнання стенду. Взагалі, у контролерах MELSEC можлива реалізація стандартних таймерів (IEC61131) програмним способом або апаратних таймерів. Апаратні таймери мають обмеження по їхньої кількості та діапазону обчислення в залежності від моделі контролера. Апаратні таймери у контролерах MELSEC розділяють за параметром відліку часового інтервалу. Наприклад, контролер FX1N має такі типи апаратних таймерів із різною часовою базою: 100 мс, 10 мс, 1 мс. Таймери із часовою базою 100 мс у контролері FX1N мають такі адреса: T0...T199. Для часового інтервалу 10 мс таймери мають такі адреса: T200...T245. І нарешті, для часового інтервалу 1 мс таймери мають адреса: T246...T249. Контролери FX1N мають апаратні таймери з пам'яттю із адресами T250...T255 (часова база 100 мс).

Розглянемо реалізацію стандартних таймерів програмним способом. Отже, ЛКМ за допомогою піктограми  відкрийте додаткове вікно **Function Block Selection Window**.

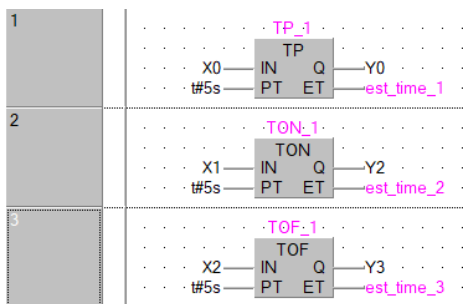
Зауважимо, що усі стандартні таймери (TP, TON та TOF) мають у своєму складі специфічний операнд. Це – операнд типу TIME, який визначає певний проміжок часу. У параметра типу TIME є свій формат представлення. Цей формат для константи має такий вигляд: **T#5s**.

Далі, якщо ЛКМ натиснути на піктограму потрібного функціонального блока, наприклад, блока імпульсного таймера (TP), та не відпускаючи ЛКМ перенести до робочого простору, отримаємо такий результат:



Далі, подібно до попереднього пункту (п.3.3.1), потрібно зв'язати формальні параметри блоку таймера із операндами. Для запуску таймерів встановіть інтервал часу, який дорівнює 5с (Т#5s). Для відображення поточного значення часу використайте змінні типу TIME (est\_time\_1, est\_time\_2 та est\_time\_3). Стан таймерів зв'яжіть із фізичними дискретними виходами ПЛК Y0, Y2, та Y3.

Таким чином, програма користувача із таймерами буде таким:



На завершення виконання цього завдання потрібно скопіювати проект, зберегти та завантажити його до ПЛК. У разі самостійного ви-

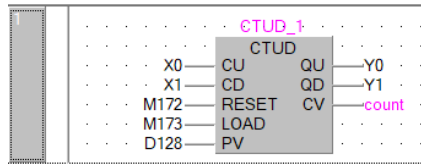


конання завдання для налагодження проекту скористайтесь симулятором роботи ПЛК, як це наведено раніше.

### 3.3.3 Керування лічильниками та таймерами за допомогою індикаторного модуля

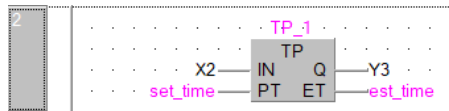
Проведіть підготовчі роботи відповідно до дій, які описані у *Практичному завданні 1* щодо створення проекту у середовищі GX\_W2. Тип проекту оберіть **Structured Project**. Ім'я проекту призначить таке: **ex\_03\_hmi.gxw**.

Додайте до робочого простору програми користувача раніше описаним способом блок комбінованого лічення (CTUD). Підключіть до входів та виходів блока операнди як це зображено нижче.



На зображенні використані фізичні входи (X0, X1) та виходи (Y0, Y1) контролера та операнди для керування станом лічильника (M172, M173) і встановлення параметру лічення (D0). Останні операнди отримують свої значення від індикаторного блока. Поточне значення лічильника зберігається у змінній з ім'ям **count**.

Додайте до програми новий ланцюг, а до нього таймер TP. З'єднайте входи та виходи таймера із операндами, як це зображено нижче:



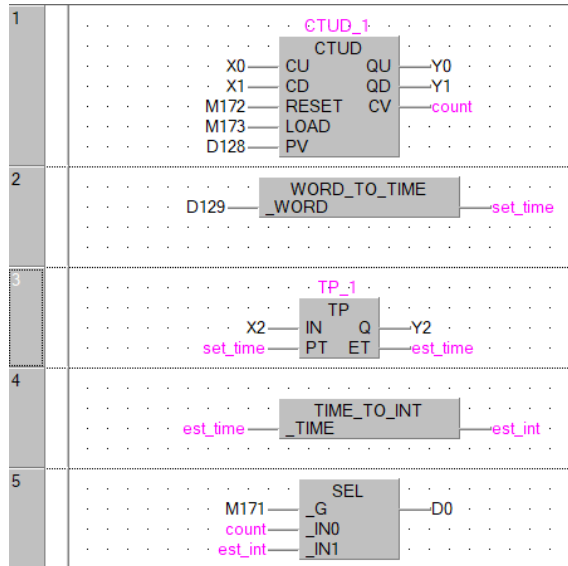
Параметри PT та ET блока таймера TP мають тип даних TIME. Тому змінні **set\_time** та **est\_time** теж повинні бути оголошені з типом TIME.

Але для повноцінної роботи ПЛК та взаємодії з індикаторним модулем цього недостатньо. Потрібно зв'язати локальні змінні із регістрами зберігання даних та зробити перетворення типів даних. Так, змінна з типом даних TIME не може бути передана по інтерфейсу. Можливе передавання лише значень операндів D, тобто цілечисельне значення. Це буде значення, яке відображає певну кількість часових інтервалів. У процесі перетворення з типу WORD до типу TIME буде отримане значення кількості мілісекунд. Наприклад, значення регістру D129 дорівнює числу 2500. Після перетворення це значення буде відповідати часовому інтервалу  $2500 \cdot 1 \text{ мс} = 2,5 \text{ с}$ . Функція перетворення буде такою: WORD\_TO\_TIME. На виході функції перетворення буде змінна set\_time. Ланцюг з цією функцією потрібно додати раніше ланцюга із таймером. Для відображення поточного значення часу у таймері потрібно зробити перетворення TIME\_TO\_INT для змінної est\_time. Вихід функції перетворення типів даних необхідно з'єднати із змінною est\_int. Ланцюг з цією функцією потрібно додати після ланцюга із таймером. Індикатор у робочому режимі відображає значення регістра D0. Для того, щоб відображати поточне значення лічильника або поточного часу таймера на індикаторному модулі можливий такий спосіб. Керувати відображенням можна за допомогою флага M171, станом якого керує кнопка UP на індикаторному модулі. У свою чергу, цей флаг буде керувати функцією вибору SEL. У звичайному режимі роботи індикаторного модуля відображається поточне значення лічильнику. Якщо натиснути на кнопку UP, то відобразиться поточне значення часу.

Нижче наведена таблиця символічних імен програми MAIN:

	Class	Label Name	Data Type
1	VAR	CTUD_1	CTUD
2	VAR	count	Word[Signed]
3	VAR	TP_1	TP
4	VAR	set_time	Time
5	VAR	est_time	Time
6	VAR	est_int	Word[Signed]

У результаті розроблена програма користувача MAIN буде такою:



На завершення виконання цього завдання потрібно скопіювати проект, зберегти та завантажити його до ПЛК. У разі самостійного виконання завдання для налагодження проекту скористайтесь симулятором роботи ПЛК, як це наведено раніше.

### 3.4 Перевірка роботи навчального проекту

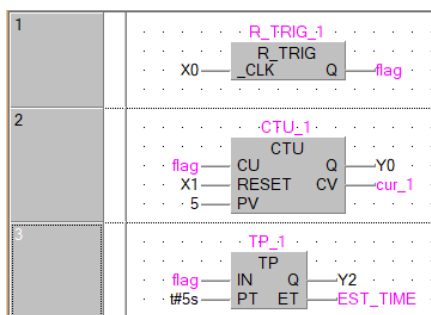
1) Перевірте роботу програми користувача на стенді або за допомогою симулятора. Для цього вмикайте кнопки або примусово встановлюйте значення операндів та спостерігайте за усіма змінами у програмі користувача. Зробіть декілька скріншотів робочого простору із програмою користувача у режимі моніторингу.

2) Складіть звіт у електронному вигляді відповідно до правил оформлення звітів: відомості про виконавця, назву та мету роботи, схему з'єднань, лістинг (зображення) програми користувача та скріншоти екранів у режимі Online. Для документування проекту використайте відповідну утиліту.

### 3.5 Завдання для самостійної роботи

1) Керування насосом у трубопроводі подавання сировини.

Кнопкою PUSK (операнд X0) вмикається електричний двигун насоса. Насос повинен працювати на протязі п'яти секунд. Далі насос автоматично вимикається. Для вмикання насоса потрібно знову натиснути на кнопку PUSK. Одночасно у програмі потрібно рахувати кількість вмикань насоса. Програма керування насосом зображена нижче:



Перевірте правильність роботи програми користувача. У процесі налагодження проекту під час роботи насоса натисніть ЛКМ на кнопку PUSK. Спостерігайте за значенням змінної кількості вмикань. Якщо є застереження, що програма працює неправильно (наприклад, хибне значення кількості вмикань), то виправте цю помилку самостійно. Тобто, змініть програму так, щоб не було хибного рахування вмикань насоса.

2) Взаємодія контролера з індикаторним модулем.

Передбачте для попереднього завдання можливість введення часу роботи насоса та скидання кількості вмикань за допомогою індикаторного модуля.

3) Удосконалення проекту із програмою керування насосом.

Розробіть проект із програмою керування насосом на трубопроводі подавання сировини з можливістю вимірювання часу роботи насоса та захистом роботи насоса від «сухого ходу». Кнопкою PUSK вмикається електричний двигун насоса. Нехай для забезпечення захисту насо-

су від «сухого ходу» в трубопровід вбудований датчик тиску DAT. Якщо на протязі п'яти секунд не спрацює датчик тиску, то насос автоматично вимикається через три секунди. Доопрацюйте попередній проект із програмою керування насосом.

4) Розробіть проект, який реалізує дискретне керування насосом із використанням таймерів, а також датчика тиску аналогової дії.

Схема установки водопостачання подібна до зображеної на стор. 38, а умови роботи описані в першому завданні для самостійної роботи у другому практичному завданні. Для імітування роботи датчика рівня використайте кнопковий модуль аналогового вводу, а поріг для відключення встановіть на рівні 20 % від всього діапазону (0...4096 од.).

### **3.6 Контрольні питання**

- 1) Дайте коротку характеристику таймерів згідно з *IEC61131*.
- 2) Дайте коротку характеристику лічильників згідно з *IEC61131*.
- 3) Як використовують таймери та лічильники в програмах керування зовнішніми пристроями?
- 4) Дайте коротку характеристику компонентів бібліотеки *Standard.lib*.

#### *Практичне завдання 4*

### ПРИНЦИПИ РЕАЛІЗАЦІЇ РЕГУЛЮВАННЯ У КОНТРОЛЕРАХ MELSEC СЕРІЇ FX

#### **4.1 Мета завдання**

- закріплення теоретичних знань щодо принципів двопозиційного керування дискретними виходами ПЛК та принципів аналогового керування дискретними виходами ПЛК (ШІМ);
- розроблення програми користувача для двопозиційного керування дискретними виходами та для ШІМ-регулятора у ПЛК MELSEC серії FX;
- розроблення програми користувача для ПІД-регулятора у ПЛК MELSEC серії FX.

#### **4.2 Порядок виконання завдання**

Виконання практичного завдання розраховано на 4 ак. год. та складається з таких етапів:

1) Створення проєкту та розроблення програми двопозиційного сигналізатора та регулятора.

2) Створення проєкту та розроблення програми ШІМ-регулятора на прикладі використання функції PWM.

3) Створення проєкту та розроблення програми користувача із ПІД-регулятором з використанням ШІМ-регулювання.

#### **4.3 Хід виконання завдання**

*4.3.1 Створення проєкту з програмою двопозиційного сигналізатора про параметр*

Необхідно реалізувати в проєкті такі функції:

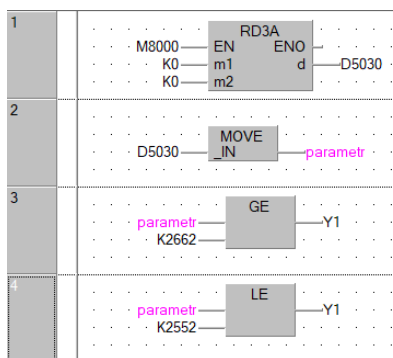
– вимірювання температури за допомогою датчика, який підключений до аналогового входу;

– вмикання сигналізації (індикатор, Y1), якщо температура буде вища від заданого максимального значення (за умовчанням 80°C);

– вимикання сигналізації, якщо температура буде нижча від заданого мінімального значення (за умовчанням 60°C).

Нехай аналоговий сигнал імітується кнопковим модулем, який підключений до входу ПЛК FX1N (див. опис стенда). Порядок оброблення аналогового сигналу розглянуто під час виконання *Практичного завдання 2*. Кодове значення вхідного аналогового сигналу порівнюється із константою. Умови перетворення будуть такими. Сигнал давача, наприклад термометра опору із перетворювачем у сигнал напруги постійного струму 0...10 V. Це відповідає температурі всередині ОУ у діапазоні -50...150 °С. Тобто, якщо температура дорівнює 80 °С, то аналоговий сигнал буде розраховуватись за такою формулою:  $10V / (150^{\circ}C - (-50^{\circ}C)) \cdot (80^{\circ}C - (-50^{\circ}C)) = 6,5 \text{ VDC}$ . З іншого боку перетворення фізичного параметру (напруга) у код АЦП (12-бітний) буде розраховуватись так:  $6,5V \cdot 4096 \text{ од.} / 10V = 2662 \text{ од.}$  Подібні розрахунки щодо температури 60°C дають такі результати: сигнал давача –  $10V / (150^{\circ}C - (-50^{\circ}C)) \cdot (60^{\circ}C - (-50^{\circ}C)) = 5,5 \text{ VDC}$ , код АЦП –  $5,5V \cdot 4096 \text{ од.} / 10V = 2522 \text{ од.}$  Таким чином було отримано коди значень завдань для 2-х позиційного сигналізатора.

Проведіть підготовчі роботи відповідно до дій, які описані у *Практичному завданні 1* щодо створення проєкту у середовищі GX\_W2. Тип проєкту оберіть **Structured Project**, а мову програмування – **Structured Ladder/FBD**. Ім'я проєкту призначить таке: `ex_04_sig.gxw`. Програму користувача розробіть подібно до наведених раніше прикладів. Фрагмент програми зображений нижче:



Перші два ланцюги призначені для отримання поточного значення параметру на аналоговому вході. Операнд `parameter` є локальною змінною та оголошується під час розроблення програми за допомогою асистента оголошення змінних. Третій та четвертий ланцюги є компараторами. Вони порівнюють значення вхідного параметру із завданням. В залежності від результатів порівняння вмикається або вимикається індикатор. У програмі операнди завдання мають постійне значення (збережані раніше). У подальшому можливе встановлення значень завдань за допомогою індикаторного модуля. Для цього достатньо замість констант застосувати для передавання значень операнди `D128` та `D129`.

На завершення виконання цього завдання потрібно скопіювати проєкт, зберегти та завантажити його до ПЛК. У разі самостійного виконання завдання для налагодження проєкту скористайтесь симулятором роботи ПЛК, як це наведено раніше.

#### *4.3.2. Загальні відомості про принципи 2-х позиційного регулювання*

Регулятор – це пристрій, який призначений для підтримки контрольованої величини на заданому рівні. Для цього можна використовувати вихідний пристрій з двома станами: включено або виключено. Причому для підтримки заданого значення регульованого параметра можна використовувати різні типи виконавчих пристроїв, але всі вони можуть бути умовно розділені на дві групи: *нагрівачі* та *охолоджувачі*.

*Нагрівачем* називають пристрій, включення якого повинне приводити до збільшення значення вимірюваного параметра. У цьому випадку процес регулювання має прямий характер.

*Охолоджувачем* називають пристрій, включення якого повинне приводити до зменшення значення вимірюваного параметра. У цьому випадку процес регулювання має зворотний характер.

Алгоритм сигналізатора із попереднього прикладу можна перетворити у алгоритм 2-х позиційного регулятора.

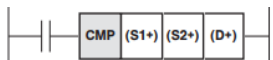
Отже, знову проведіть підготовчі роботи відповідно до дій, які



описані у *Практичному завданні 1* щодо створення проєкту у середовищі GX\_W2. Тип проєкту оберіть **Simple Project**, а мову програмування – **Ladder**. Ім'я проєкту призначить таке: **ex\_04\_reg2.gxw** .

Нижче наведено приклад програми терморегулятора у режимі прямого гістерезиса, тобто нагрівача (операнд Y3) з індикатором (операнд Y1). Завдання встановить на рівні 21°C. Відповідно розрахункові значення аналогового сигналу та коду АЦП будуть дорівнювати відповідно 3,55 VDC та 1450 од. У програмі передбачено блокування вимкнення нагрівача за допомогою термоперемикача типу **NO** (див. опис стенда), який має фізичну адресу у ПЛК X4. Додатково у програмі передбачена сигналізація про вимкнений стан нагрівача за допомогою червоного індикатора на виході Y1. Програму користувача розробіть подібно до наведених раніше прикладів.

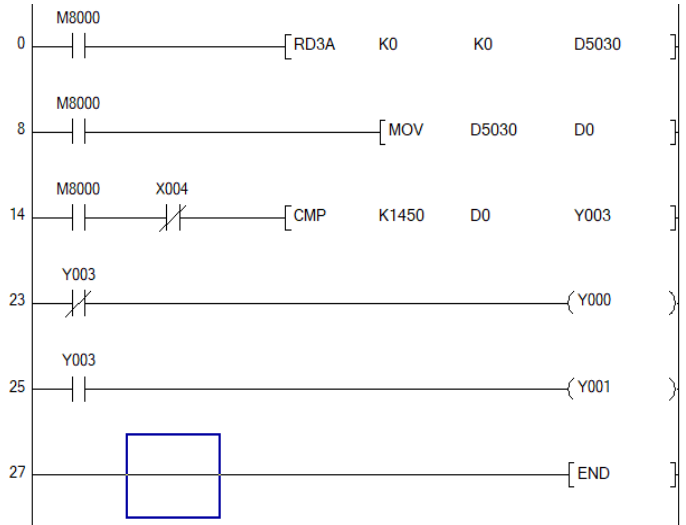
У програмі використана інструкція порівняння **CMP** (**FNC10**). Ця інструкція призначена для порівняння двох значень у двійковому вигляді: більше, рівно, менше. Інструкція має такий формат:



. Надамо пояснення щодо цієї інструкції. Після назви інструкції розміщені її параметри: **(S1+)**, **(S2+)** та **(D+)**. Перші два параметра визначають значення, які порівнюються. Третій параметр зберігає результат порівняння. Причому, якщо здійснюється порівняння «більше», то третій параметр буде **(D+)**. Якщо здійснюється порівняння «рівно», то третій параметр буде **((D+)+1)**. Якщо здійснюється порівняння «менше», то третій параметр буде **((D+)+2)**.

На завершення виконання цього завдання потрібно скопіювати проєкт, зберегти та завантажити його до ПЛК. У разі самостійного виконання завдання для налагодження проєкту скористайтесь симулятором роботи ПЛК, як це наведено раніше.

Фрагмент програми зображений нижче:



#### 4.3.3. Загальні відомості про принципи ШІМ-регулювання

Сигнал ШІМ – це імпульсні сигнали постійної частоти (періоду) та змінної шпаруватості (тривалість імпульсів). За допомогою завдання шпаруватості можна змінювати потужність на дискретному виході ПЛК. Наприклад, це дозволяє регулювати температуру у термошафі з дискретними нагрівальними елементами – інфрачервоними лампами.

Шпаруватість – одна з класифікаційних ознак імпульсних систем, яка визначає відношення періоду слідування сигналу до тривалості імпульсу. Величина, яка обернена шпаруватості, в англійській літературі називається коефіцієнтом заповнення (від англ. *Duty Cycle*).

Таким чином, для імпульсного сигналу справедливі такі співвідношення:  $S = T / \tau = 1 / D$ , де  $S$  – шпаруватість,  $D$  – коефіцієнт заповнення,  $T$  – період слідування імпульсів,  $\tau$  – тривалість імпульсу.

Найчастіше застосування в практиці знаходить сигнал з шпаруватістю, яка дорівнює двом. Це так званий сигнал типу «меандр».

В режимі аналогового регулювання логічний пристрій розраховує відхилення  $E$  (тобто, розузгодження) поточного значення контролюва-

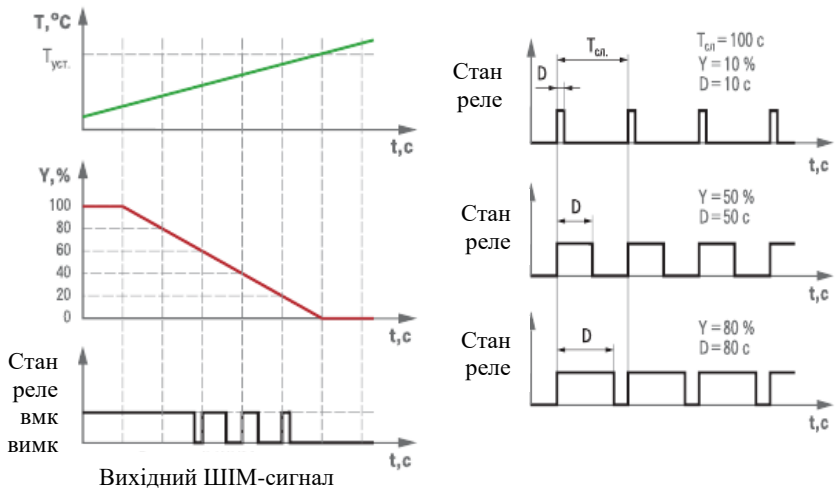
ної величини  $T$  від заданого завдання  $T_{завд.}$ . В результаті на виході регулятора виробляється аналоговий сигнал  $Y$ , який направлений на зменшення розузгодження  $E$ . Цей сигнал подається на виконавчий пристрій регулятора у вигляді послідовності імпульсів (ШІМ).

Якщо вихідний пристрій регулятора ключового типу (реле, транзисторна або симісторна оптопара, вихід для управління твердотільним реле), вихідний сигнал перетвориться в послідовність управляючих імпульсів з тривалістю  $D$ :

$$D = Y \frac{T_{сл}}{100\%},$$

де  $D$  – тривалість імпульса, с;  $T_{сл}$  – період слідування імпульсів, с (задається користувачем у програмі);  $Y$  – вихідний сигнал регулятора.

Принципи формування ШІМ-сигналу для «нагрівача» при різних значеннях вихідного сигналу  $Y$  та діаграми ШІМ-сигналів для різних значень шпаруватості зображені нижче:



Отже, знову проведіть підготовчі роботи відповідно до дій, які описані у *Практичному завданні 1* щодо створення проєкту у середовищі GX\_W2. Тип проєкту оберіть **Structured Project**, а мову програ-

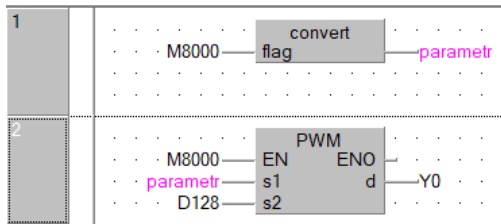
мування – Structured Ladder/FBD. Ім'я проєкту призначить таке: `ex_04_pwm.gxw`. Сигнал заповнення формується функцією користувача `convert`, яка складається із фрагменту програми на стор. 50. Порядок створення функції користувача розглянуто у *Практичному завданні 3*.

У програмі використана інструкція PWM (FNC58). Ця інструкція призначена для формування послідовності імпульсів із модуляцією їхньої ширини при фіксованому значенні періоду. Інструкція має такий формат:

формат: . Надамо пояснення щодо цієї інструкції.

Після назви інструкції розміщені її параметри: (S1+), (S2+) та (D+). Перший параметр визначає довжину імпульсу у діапазоні 1...32767 мс. Другий параметр визначає період слідування імпульсів у діапазоні 1...32767 мс. Довжина імпульсу (S1+) має бути меншою ніж період слідування імпульсів (S2+). Третій параметр вказує на адресу виходу. У контролерах FX1N можливе використання лише виходів Y0 та Y1. Причому виходи ПЛК мають бути транзисторними.

Нижче зображено фрагмент програми формування ШІМ-сигналу за допомогою функції PWM. Причому довжину імпульсу визначає змінна `parameter`, яка є результатом перетворення аналогового сигналу за допомогою функції `convert`:

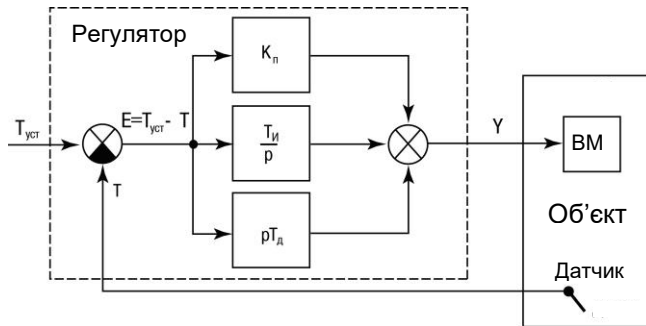


На завершення виконання цього завдання потрібно скопіювати проєкт, зберегти та завантажити його до ПЛК. У разі самостійного виконання завдання для налагодження проєкту скористайтесь симулятором роботи ПЛК, як це наведено раніше.

#### 4.3.4. Програмування ПІД-регулятора

Проведіть підготовчі роботи відповідно до дій, які описані у *Практичному завданні 1* щодо створення проєкту у середовищі GX\_W2. Тип проєкту оберіть **Structured Project** а мову програмування – **Structured Ladder/FBD**. Ім'я проєкту призначить таке: `ex_04_pid.gxw`.

Нагадаємо, що основне завдання ПІД-регулятора – це підтримання поточного параметра на заданому рівні шляхом впливу на виконавчі механізми. Загальна схема ПІД-регулятора зображена нижче:



Пропорційно-інтегрально-диференційний (ПІД) регулятор використовується в системах автоматики для підтримки з високою точністю потрібних параметрів. Він видає вихідний сигнал, направлений на зменшення відхилення поточного значення регульованого параметру від завдання. У загальному випадку робота універсального ПІД-регулятора для вихідного сигналу ( $Y_i$ ) може бути описана таким рівнянням:

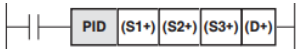
$$Y_i = K_p \cdot \left[ E_i + K_d T_d \frac{\Delta E_i}{\Delta t_{вим}} + \frac{1}{\tau_i} \sum_{i=0}^n E_i \Delta t_{вим} \right],$$

де  $K_p$  – смуга пропорційності;  $E_i$  – різниця між завданням та поточним значенням контрольованої величини або відхилення;  $\Delta E_i$  – різниця між двома послідовними вимірами  $E_i$  та  $E_{i-1}$ ;  $\Delta t_{вим}$  – час між двома послідовними вимірами;  $K_d$  – коефіцієнт диференціювання;  $T_d$  – постійна

часу диференціювання;  $\tau_i$  – постійна інтегрування;  $\sum_{i=0}^n E_i$  – накопичене

сумарне відхилення.

У програмі використана інструкція PID (FNC88). Ця інструкція призначена для програмування контуру регулювання та має 25 параметрів для його налаштувань. Інструкція має такий формат:



. Інструкція використовує операнди типу 16-бітові регістри загального призначення (D). Призначення параметрів інструкції таке:

- параметр (S1+) визначає значення завдання для стеження;
- параметр (S2+) визначає поточне значення технологічного параметра;
- параметр (S3+) визначає значення початкового регістру для зберігання параметрів регулювання. Усього резервується 25 регістрів даних, які слідують один за одним;
- параметр (D+) призначений для зберігання значення управляючої дії.

В таблиці наведені відомості про усі 25 параметрів регулятора.

(S3+)	Призначення	Опис	Значення
+0	Інтервал часу вибірки	Інтервал часу перевірки значення поточного параметру	1...32767мс
+1	Напрямок руху/ контроль аварії	Біт 0: 0 – уперед, 1 – назад	
		Біт 1: 0/1 – уперед, 1 – назад	
		Біт 2: 0/1 – уперед, 1 – назад	
		Біт 3: резерв	
		Біт 4: автоналаштування: 1 – старт, 0 – вимкнено	

		Біт 5: обмеження вихідного значення: 1 – так, 0 – ні	
		Біт 6: резерв	
+2	Вхідний фільтр	Встановлення часу вхідного фільтрування	0...99%
+3	Коефіцієнт пропорційності $K_p$	Коефіцієнт множення для пропорційного регулювання	1...32767%
+4	Постійна часу інтегрування $\tau_i$	Коефіцієнт множення для інтегрального регулювання	1...32767• 100мс 0 – немає регулювання
+5	Диференційний коефіцієнт $K_d$	Коефіцієнт множення для диференційного регулювання	0...100%
+6	Постійна часу диференціювання $T_d$	Коефіцієнт множення для диференційного регулювання	1...32767• 10мс 0 – немає регулювання
+7...19	Резерв	–	–
+20	Аварійне значення поточного параметру (зростаючого)	Формування сигналу аварії, якщо параметр перебільшив вказаний параметр	0...32767
+21	Аварійне значення поточного параметру (падаючого)	Формування сигналу аварії, якщо параметр менше вказаного параметру	0...32767
+22	Аварійне значення вихідного параметру (зростаючого)	Формування сигналу аварії, якщо вихідний параметр перебільшив вказане значення	0...32767
+23	Аварійне значення вихідного параметру	Формування сигналу аварії, якщо вихідний	0...32767

	(падаючого)	параметр менше за вказане значення	
+24	Формування сигналу аварії	Біт 0: аварія поточного значення (перебільшено)	–
		Біт 1: аварія поточного значення (знижено)	
		Біт 2: аварія вихідного значення (перебільшено)	
		Біт 3: аварія вихідного значення (знижено)	

Контролери MELSEC серії FX мають функцію авто налаштування ПІД-регулятора. Ця функція визначає початкові параметри регулювання. Це такі параметри:

- коефіцієнт пропорційності  $K_p$  ((S3+)+3);
- постійна часу інтегрування  $\tau_i$  ((S3+)+4);
- постійна часу диференціювання  $T_d$  ((S3+)+6);
- напрям регулювання ((S3+)+1): біт 0.

Інші параметри повинен вказати користувач. Функція автоналаштування активується встановленням четвертого біта у ((S3+)+1)). Вихідне значення параметру утримується у межах встановленого початкового значення. Реакція ОУ у вигляді поточного параметра контролюється. Якщо значення поточного параметра досягає 1/3 від завдання, функція авто налаштування вимикається, а біт 4 у ((S3+)+1)) скидається.

Під час автоналаштування потрібно слідкувати за такими параметрами:

- відхилення поточного параметра від завдання має бути 150 %;
- інтервал часу перевірки поточного значення параметру має бути кратним часу сканування та більшим ніж 1 с;

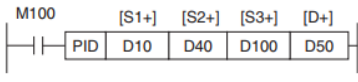


– перед ввімкненням функції авто налаштування ОУ має бути у стабільному стані.

Усі значення параметрів можна зберегти у регістрах даних. Якщо потрібно зберегти ці значення під час зникнення живлення має бути застосовуватись регістри із пам'яттю.

Кількість ПІД-інструкцій у програмі користувача необмежена. Але потрібно слідкувати за областю даних (S3+) , щоб не було перетину даних всередині блоку даних.

Приклад програмування ПІД-інструкцій має такий вигляд



Виконання ПІД-регулювання

активується, якщо встановлений флаг M100. Завдання параметру збережено у регістрі D10, поточне значення параметру – у регістрі D40, управляюча дія – у регістрі D50. Параметри налаштування збережені у регістрах загального призначення D100...D124.

Таким чином, для розроблення ПІД-регулятора потрібно у попередньому прикладі (див. п. 4.3.3) додати інструкцію PID (FNC88) із встановленням потрібних значень параметрів регулятора між першим та другим ланцюгом. Тобто, поточний параметр визначається за допомогою функції convert , а керуюча дія інструкції PID перетворюється у сигнал ШІМ за допомогою функції PWM. Таким чином управляюча дія перетворюється у довжину імпульсу у режимі ШІМ.

У таблиці, яка зображена нижче наведені параметри налаштування ПІД-регулятора для різноманітних варіантів законів регулювання.

Тип регулятора	Вибір регістра зберігання даних			Опис
	$K_p$ ((S3+)+3)	$\tau_i$ ((S3+)+4)	$T_d$ ((S3+)+6)	
P	значення користувача	встановлення 0	встановлення 0	пропорційне регулювання
PI	значення	встановлення	встановлення	пропорційне

	користувача	0	0	інтегральне регулювання
PD	значення користувача	встановлення 0	значення користувача	пропорційне диференційне регулювання
PID	значення користувача	значення користувача	значення користувача	повне регулювання

Реалізація регулювання вперед/назад ((S3+)+1), b0. Регулювання вперед/назад визначається за ім'ям. Тобто, так визначається рух точки регулювання відносно попереднього стану точки. Можливі два випадки:

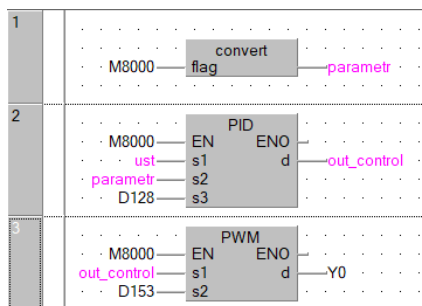
- поточне значення (S2+) більше ніж завдання (S1+);
- поточне значення (S2+) менше ніж завдання (S1+);

Нижче зображено обидва випадки у вигляді графіку зміни параметру у часі на системі координат. На графіку горизонтальна вісь означає параметр, вертикальна – напрям руху та швидкість. Пунктиром на графіку позначено рух уперед, суцільна лінія – рух назад.



Надамо деякі пояснення щодо наведеного графіку. Рух уперед означає, що у випадку збільшеного значення означає регулювання униз та назад поточного значення параметру або у випадку зменшеного поточного значення у регулюванні уверх та уперед поточного значення. Рух назад означає, що поточне значення занадто велике, регулювання рухається униз та уперед, або у випадку регулювання має бути уверх та назад. На поданому графіку не враховуються коефіцієнти регулювання для пропорційного, диференційного та інтегрального законів регулювання або їхніх комбінацій.

Отже, розробіть програму користувача для регулювання температури всередині ОУ із використанням дискретного вихідного пристрою із ШІМ-регулюванням. Нижче зображено фрагмент програми користувача:



Параметри ПІД-регулятора встановить згідно наведеному у наступній таблиці.

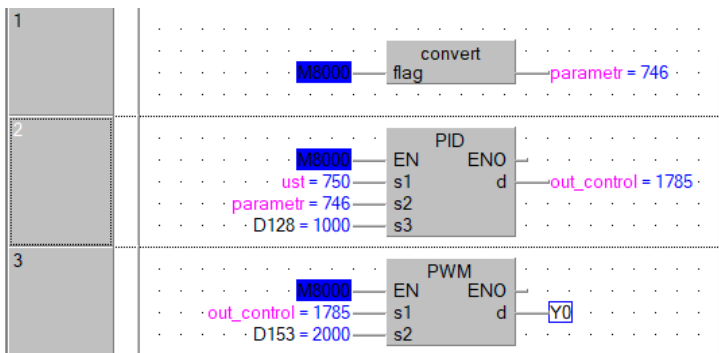
Адрес операнда	Призначення	Значення
D128	Інтервал часу вибірки	1000 мс
D129	Напрямок руху/контроль аварії	6 5 4 3 2 1 0 0 1 1 0 0 0 0 DEC 48 HEX 0030
D130	Вхідний фільтр	10 %
D131	Коефіцієнт пропорційності $K_p$	10 %
D132	Постійна часу інтегрування $\tau_i$	1000 мс
D133	Диференційний коефіцієнт $K_d$	5 %
D134	Постійна часу диференціювання $T_d$	500 мс
D135... D147	Резерв	—
D148	Аварійне значення поточного параметру (зростаючого)	4096
D149	Аварійне значення поточного параметру (падаючого)	4096

D150	Аварійне значення вихідного параметру (зростаючого)	1999
D151	Аварійне значення вихідного параметру (падаючого)	1999
D152	Формування сигналу аварії	—
D153	Період PWM	2000 ms

Для зручності налаштування параметрів ПД-регулятора потрібно заповнити перелік Global1 у гілці Global Label менеджера проекту. Приклад заповнення переліку Global1 із параметрами налаштувань зображений нижче:

	Class	Label Name	Data Type	Constant	Device	Address
1	VAR_GLOBAL	time_zikl	Word[Unsigned]/Bit String[16-bit]	...	D128	%MW0.128
2	VAR_GLOBAL	ar_al	Word[Unsigned]/Bit String[16-bit]	...	D129	%MW0.129
3	VAR_GLOBAL	inp_filter	Word[Unsigned]/Bit String[16-bit]	...	D130	%MW0.130
4	VAR_GLOBAL	k_p	Word[Unsigned]/Bit String[16-bit]	...	D131	%MW0.131
5	VAR_GLOBAL	t_i	Word[Unsigned]/Bit String[16-bit]	...	D132	%MW0.132
6	VAR_GLOBAL	k_d	Word[Unsigned]/Bit String[16-bit]	...	D133	%MW0.133
7	VAR_GLOBAL	t_d	Word[Unsigned]/Bit String[16-bit]	...	D134	%MW0.134
8	VAR_GLOBAL	pv_alam_up	Word[Unsigned]/Bit String[16-bit]	...	D148	%MW0.148
9	VAR_GLOBAL	pv_alam_down	Word[Unsigned]/Bit String[16-bit]	...	D149	%MW0.149
10	VAR_GLOBAL	upr_alam_up	Word[Unsigned]/Bit String[16-bit]	...	D150	%MW0.150
11	VAR_GLOBAL	upr_alam_down	Word[Unsigned]/Bit String[16-bit]	...	D151	%MW0.151
12	VAR_GLOBAL	alam	Word[Unsigned]/Bit String[16-bit]	...	D152	%MW0.152
13	VAR_GLOBAL	t_pwm	Word[Unsigned]/Bit String[16-bit]	...	D153	%MW0.153

Нижче зображено програму користувача у режимі моніторингу із встановленими параметрами:



На завершення виконання цього завдання потрібно скопіювати проєкт, зберегти та завантажити його до ПЛК. У разі самостійного виконання завдання для налагодження проєкту скористайтесь симулятором роботи ПЛК, як це наведено раніше.

#### **4.4 Завдання для самостійної роботи**

1) Розробіть проєкт двопозиційного сигналізатора із можливістю встановлення меж за допомогою індикаторного модуля.

2) Розробіть проєкт двопозиційного сигналізатора із використанням інструкції LIMITATION.

3) Розробіть проєкт двопозиційного регулятора типу «нагрівач» із затримкою вмикання та вимикання виходу.

4) Розробіть проєкт двопозиційного регулятора типу «охолоджувач».

5) Розробіть проєкт ПД-регулятора із аналоговим керуванням.

#### **4.5 Контрольні питання**

1) Поясніть принцип дії 2-позиційного регулятора параметра.

2) Дайте коротку характеристику принципу дії ШІМ-регулятора.

3) Як визначається параметр шпаруватості у ШІМ-регуляторі?

4) Дайте коротку характеристику принципів ПД-регулювання.

5) Яким чином можна реалізувати ПД-регулювання за допомогою дискретних виходів ПЛК?

## ДОДАТОК А

### ОПИС ТА ПРИНЦИПИ ФУНКЦІОНУВАННЯ СТЕНДА

На рис. А.1 зображено зовнішній вигляд стенда з контролером FX1N. Для практикуму використовують два стенда, до складу яких входять контролери однакової модифікації FX1N, але із різним виконанням «green» або «red». Контролери відрізняються кольором індикатора роботи програми користувача – «зелений» та «червоний», та різними адресами аналогових входів. Стенд складається із двох частин: блок із фізичною моделлю теплового об'єкту і блоком живлення (об'єкт керування – ОК) та блок із контролером та органами керування (система керування – СК). Система керування, крім органів керування, має зовнішній адаптер інтерфейсу TTL/232 (або TTL/USB). Цей адаптер призначений для підключення середовища програмування, тобто персонального комп'ютера, до контролера для перевірки роботи програм керування або моніторингу технологічних параметрів всередині ОК.

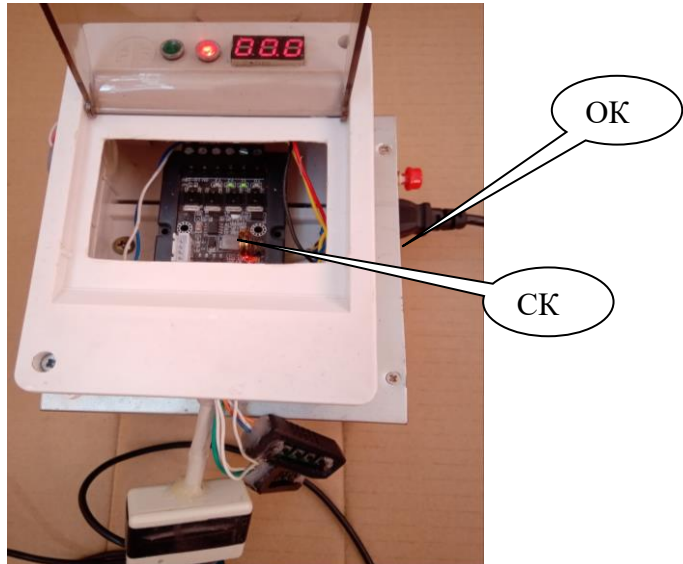


Рис. А.1 – Зовнішній вигляд стенда

Електрична схема з'єднань обох частин стенда та обладнання зображена на рис. А.2. Схеми для моделей «*green*» та «*red*» подібні. Відмінність полягає лише у логічних адресах аналогових входів. У табл. А.1 та А.2 наведена інформація з адресами дискретних та аналогових входів та виходів (підключення вхідних та вихідних елементів стенда).

Рис. А.2 – Електрична схема з'єднань стенда

Табл. А.1 – Адреса входів та виходів контролера (FX1N-10MT, «green»)

DI		DO	
X0	Button 1	Y0	Lamp (GREEN)
X1	Button 2	Y1	Lamp (RED)
X2	Button 3	Y2	COOLER
X3	Button 4	Y3	HEATER
X4	TermoSwitch (NC)	D5030	ADC 0...10V
X5	TermoSwitch (NO)	D5101	DAC 0...3,3V

Табл. А.2 – Адреса входів та виходів контролера (FX1N-10MT, «red»)

DI		DO	
X0	Button 1	Y0	Lamp (GREEN)
X1	Button 2	Y1	Lamp (RED)
X2	Button 3	Y2	COOLER
X3	Button 4	Y3	HEATER
X4	TermoSwitch (NC)	D5000	ADC 0...10V
X5	TermoSwitch (NO)	D5101	DAC 0...3,3V

Виходячи із конструктивних особливостей стенда та роботи АЦП (12 біт) та ЦАП (12 біт) контролера за допомогою модуля дискретного введення аналогових сигналів можливе формування аналогових вхідних сигналів у діапазоні 0...5 VDC із відповідним кодом АЦП 0...4096.

Приблизно розподіл сигналів буде таким:

- кнопка AI1 – код 750 – 1,85 VDC;
- кнопка AI2 – код 1125 – 2,7 VDC;
- кнопка AI3 – код 1500 – 3,7 VDC;
- кнопка AI4 – код 2020 – 4,9 VDC.

Формування аналогових сигналів теж має свої особливості. Розподіл сигналів буде таким: для ЦАП із кодом 0...4095 вихідний сигнал буде 0...3,3 V.



Такі результати програмного оброблення аналогових сигналів пов'язані з використанням вбудованого джерела живлення та недосконалістю використаних дротів, а також, з відсутністю електричної розв'язки між аналоговим входом та виходом у контролері. Але, на загальне знайомство з програмним обробленням аналогових сигналів це не впливає. Крім того, у складі стенда використана модель контролера, яка є лише клоном оригінального контролера Mitsubishi Melsec FX1N.

Згідно із схемою електричних з'єднань (див. рис. А.2) та табл. А.1 кнопки (типу «сухий контакт») дискретного модуля підключені до дискретних входів контролера із замиканням відповідного входу на нульову шину контролера без електричної розв'язки, за схемою із внутрішнім живленням. До входів X4 та X5 підключені термоперемикачі, які налаштовані на температуру перемикання 40 °С, які є нормально замкнутими (NC) та розімкнутими (NO) відповідно.

Для керування дискретними виходами в контролері використані транзистори, які підключені по схемі із відкритим колектором. Тому реалізована схема з використанням схеми загального живлення (+12 VDC) від блока живлення стенда. Усі дискретні виходи контролера, крім виходу Y3 (HEATER), підключені безпосередньо. Для підключення нагрівача, тобто лампи розжарювання, використано проміжне електромеханічне реле.

Отже, використані в стенді засоби керування дозволяють створювати невеликі системи керування технологічними об'єктами із різною функціональністю.

Насамкінець, зазначимо, що до складу стенду входить спеціальний модуль з індикатором та кнопками керування. Якщо використати функціонал цього модуля, то можливе створення СК із зовнішнім модулем керування та відображення потрібних параметрів. Для того, щоб його використати у процесі керування технологічним процесом, потрібно до програми керування додати потрібні інструкції із прив'язуванням потрібних регістрів та кнопок.

Нижче наведено короткий опис функціональних можливостей 4-х розрядного індикатора з 4-ма кнопками керування. Модуль керування призначений для відображення та завдання параметрів роботи програми спільно з контролерами Mitsubishi Melsec FX1N. Для обміну даними дисплей використовує послідовний інтерфейс *TTL/232* у дуплексному режимі. Зовнішній вигляд індикаторного модуля зображений на рис. А.3.



Рис. А.3 – Зовнішній вигляд індикаторного модуля

Наявність кнопок на платі модуля дає можливість керувати значеннями операндів у програмі ПЛК. На платі модуля розміщено чотири кнопки із комбінованим функціоналом. Це кнопки **MODE**, **UP**, **DOWN** та **RST**.

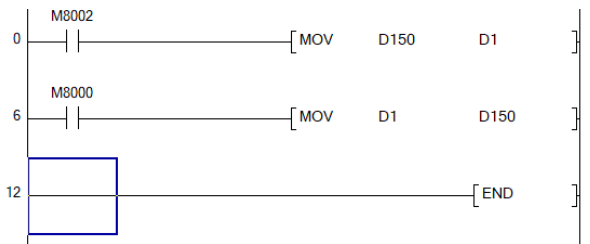
В основному режимі кнопки **UP**, **DOWN** та **RST** керують статусом трьох операндів – **M171**, **M172** та **M173**. Якщо натиснуто одну з цих трьох кнопок, то активується відповідний операнд (переходить у стан **TRUE**). У режимі налаштувань кнопки є засобом для введення даних. Кнопка **MODE** призначена для переведення дисплея в режим конфігурування, кнопки **UP** та **DOWN** – для зміни значення способом інкрементування та декрементування, а кнопка **RST** – для виходу модуля з режиму налаштувань.

Модуль в основному режимі відображає стан регістру **D0** у контролері. За допомогою модуля можливе звернення до інших регістрів після переходу до режиму налаштування. Перехід до режиму налаштування відбувається після натискання на кнопку **MODE** тривалістю більше 1 сек. Перехід до режиму налаштувань сигналізує про наявність

активної точки праворуч від символу на індикаторі. Після переходу до режиму налаштувань модуль відображає адресу поточного регістру (за замовчуванням це регістр D0). За допомогою кнопок UP та DOWN можна змінити адресу регістру на потрібну. Далі, повторне натискання на кнопку MODE відобразить поточне значення обраного регістру зберігання даних, яке також можна змінити за допомогою кнопок UP та DOWN. Режим зміни значення сигналізується миготливою точкою праворуч від символу на індикаторі. Регістри зберігання даних знаходяться у діапазоні D128...D5029. Вихід із режиму налаштувань – це натискання на кнопку RST тривалістю 1 сек. Модуль перемкнеться до основного режиму, тобто, до відображення значення регістру D0.

Функціонал модуля дозволяє керувати розрядністю відображення даних. Так, для визначення розрядності відображення регістру D0 використовується значення регістру D1. Якщо значення регістру D1 дорівнює «0», то значення регістру D0 відображається без десяткової точки. Якщо значення регістру D1 дорівнює «1», то значення регістру D0 відображається з одним знаком після точки, і так далі.

Для того, щоб налаштування формат відображення зберігався після перезавантаження ПЛК, значення регістру D1 необхідно перезаписати у програмі користувача з використанням регістрів з адресами, які більше за адресу D128 із використанням спеціальних флагів M8002 та M8000, як це показано стосовно регістру D1:



В регістрі D150 зберігається значення, яке вказує на формат відображення значення регістру D0 на індикаторному модулі.

**ДОДАТОК Б**  
**ОСНОВНІ ВІДОМОСТІ ПРО МОЖЛИВОСТІ ТА РЕСУРСИ**  
**ПЛК MITSUBISHI MELSEC FX1N**

Таблиця В.1 – Загальна характеристика ПЛК FX1N

Ознака	Технічні дані
Оброблення програми	Циклічне оброблення програми користувача
Оброблення входів/виходів	Оброблення відображення процесу. Є безпосередня обробка інструкції. Вхідний фільтр налаштовується від 0 до 15 мс.
Мова програмування	Лістинг інструкцій та релейно-контактна схема по DIN19239. Крокові інструкції
Час виконання інструкції	Базові команди: від 0,55 до 0,7 мкс Застосовувані інструкції: див. додаток В
Об'єм програми	8000 кроків, EEPROM-пам'ять
Кількість інструкцій	Набір базових команд: 29 Інструкцій керуючих кроків: 2 Застосовуваних інструкцій: 89

Таблиця В.2 – Операнди ПЛК MITSUBISHI MELSEC FX1N

Параметр		Технічні дані	
Входи/ виходи	FX1N-MR	Максимальна конфігурація технічних засобів складає 128 входів/виходів. Програмно можна адресувати 128 вх. та 128 вих.	
	FX1N-MT		
Маркери	Маркери	M0-M383	384 адреси
	Маркери з пам'яттю	M384- M1535	Дійсне значення занесено до EEPROM 1152 адреси
	Спеціальні маркери	M8000-M8256	256 адрес

Стан кроків	Ініціалізація	S0-S9		10 адрес	
	Загальні значення	S10-S999		990 адрес	
Таймер	100 мс	0,1-3276,7 с	T0-T199	200 адрес	
	10 мс	0,1-327,67	T200-T245	46 адрес	
	1 мс	0,001-32,767 с	T246-T249	4 адреси	
	100 мс, з пам'яттю	0-25,5 с	T250-T255	6 адрес	
Лічильник	Лічення вгору	16 біт +1...32767	Загальне значення	C0-C15	16 адрес
			Дійсне значення занесене до EEPROM	C16-C199	184 адреси
	Лічення вгору/вниз	32 біта	Загальне значення	C200-C219	16 адрес
			Дійсне значення занесене до EEPROM	C220-C234	15 адрес

Таблиця В.3 – Операнди ПЛК MITSUBISHI MELSEC FX1N

Параметр		Технічні дані			
Високошвидкісний лічильник	1-фазний лічильник без пуску та повернення, лічення вгору та вниз	32 біти	Дійсне значення занесено до EEPROM	C235-C240	6 адрес

	1-фазний лічильник з пуском та поверненням, лічення вгору і вниз	32 біти		C241- C245	5 адрес
	2-фазний лічильник, лічення вгору і вниз	32 біта		C246- C250	5 адрес
	A/B фазний лічильник	32 біта		C251- C255	5 адрес
Регістр	Регістр даних	16 біт	Загальні значення	D0- D127 D1000 - D7999	7128 адрес
			Дійсне значення занесено до EEPROM	D128- D255	128 адрес
	Файловий регістр	16 біт	Визначається параметром у 14-ти блоках по 500 кроків програми	D1000 - D7999	7000 адрес
	Спеціальний регістр	16 біт		D8000 -	256 адрес

			D8255	
	Зовні змінюваний регістр	16 біт, значення від 0 до 255, завдання по VR1 та VR2	D8030, D8031	дві адреси
	Індексний регістр	16 біт	V, Z	2 адреси
Точка	Точка інструкції переходу		P0- P127	128 адрес
	Точка переривання	Входи переривання: X0...X3	I00*- I130*	6 адрес
Розгалуження	Розгалуження програми, головний контакт		N0-N7	8 адрес
Константа	Десяткова	16 біт	-32768...+32767	
		32 біта	2 147 483648...+ 2 147 483 647	
	Шіснадцятирічна	16 біт	0...FFFF <sub>H</sub>	
		32 біта	0...FFFFFFFF <sub>H</sub>	

Таблиця В.4 – Інструкції MELESEC FX<sub>1S</sub> /FX<sub>1N</sub>

Класифікація	Інструкція	FNC	Призначення
Інструкція обробки програми	CJ	00	Перехід усередині програми
	CALL	01	Виклик підпрограми
	SRET	02	Кінець підпрограми
	IRET	03	Закінчити переривання програми
	EI	04	Активізувати переривання програми

	DI	05	Деактивізувати переривання програми
	FEND	06	Кінець області програмування
	WDT	07	Оновити значення таймера контролю часу сканування
	FOR	08	Початок повторення програми
	NEXT	09	Кінець повторення програми
Інструкція порівняння та перенесення	CMP	10	Порівняння числових даних
	ZCP	11	Порівняння числових областей даних
	MOV	12	Передача даних
	BMOV	15	Передача блоків
	BCD	18	BCD-конвертування
	BIN	19	BIN-конвертування
Арифметичні інструкції	ADD	20	Складання числових даних
	SUB	21	Віднімання числових даних
	MUL	22	Помноження числових даних
	DIV	23	Поділ числових даних
	INC	24	Інкремент (прирощення)
	DEC	25	Декремент (зменшення)
	WAND	26	Логічний зв'язок «І»
	WOR	27	Логічний зв'язок «АБО»
Інструкція зсуву	SFTR	34	Зсув бітів двійкових даних праворуч
	SFTL	35	Зсув бітів двійкових даних вліво
	SFWR	38	Запис у накопичувач типу FIFO
	SFRD	39	Читання з накопичувача FIFO
Операція з даними	ZRST	40	Вимкнути область операндів
	DECO	41	Декодування даних



	ENKO	42	Кодування даних
Високошвидкісні інструкції	REF	50	Обновлення входів та виходів
	MTR	52	Читання матриці
	DHSCS	53	Включення по високошвидкісному лічильнику
	DHSCR	54	Виключення по високошвидкісному лічильнику
	SPD	56	Визначення швидкості
	PLSY	57	Видача заданої кількості імпульсів
	PWM	58	Видача імпульсів із модуляцією їх ширини
	PLSR	59	Видача певної кількості імпульсів
Інструкції, орієнтовані на користувача	IST	60	Ініціалізація стану кроків
	ABSD	62	Порівняння абсолютного лічильника
	INCD	63	Порівняння інкрементального лічильника
	ALT	66	Функція пульс-пари (Flip-Flop)
	RAMP	67	Функція рампи (нахилу)

## ЗМІСТ

Вступ . . . . .	3
<i>Практичне завдання 1.</i> Загальні принципи програмування контролерів MELSEC серії FX у середовищі GX WORKS 2 . . . . .	7
<i>Практичне завдання 2.</i> Оброблення аналогових сигналів в контролерах MELSEC серії FX . . . . .	33
<i>Практичне завдання 3.</i> Основні принципи використання лічильників та таймерів у контролерах MELSEC серії FX . . . . .	41
<i>Практичне завдання 4.</i> Принципи реалізації регулювання у контролерах MELSEC серії FX . . . . .	52
Список літератури . . . . .	68
Додатки . . . . .	69

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ

для проведення комп'ютерного практикуму

«Програмування промислових контролерів Mitsubishi MELSEC серії FX»

з курсу «Програмне забезпечення промислових контролерів»

для студентів спеціальності

151 – «Автоматизація та комп'ютерно-інтегровані технології»

усіх форм навчання

Укладачі: ЛИСАЧЕНКО Ігор Григорович

ДЗЕВОЧКО Альона Ігорівна

Відповідальний за випуск проф. ПОДУСТОВ Михайло

Роботу до видання рекомендував ДУДНИК Олексій

В авторській редакції

План 2022 р., поз. 63

Підп. до друку \_\_.\_\_.2022. Гарнітура Times New Roman.

---

Видавничий центр НТУ «ХП»

Свідоцтво суб'єкта видавничої справи ДК № 5478 від 21.08.2017 р.

НТУ «ХП», вул. Кирпичова, 2, м. Харків-2, 61002

---

Самостійне електронне видання