

Кучерук В. Ю., Севастьянов В. М., Дудатьєв І. А.

**Мікропроцесори в ІВТ. Лабораторний
практикум у середовищі WinPLC7 з
використанням ПЛК VIPA**



Vipa System
200V



WinPLC7

Міністерство освіти і науки України
Вінницький національний технічний університет

Кучерук В. Ю., Севастьянов В. М., Дудатьєв І. А.

Мікропроцесори в ІВТ. Лабораторний практикум у
середовищі WinPLC7 з використанням ПЛК VIPA

Лабораторний практикум

Вінниця
ВНТУ
2013

УДК 621.3
ББК 30.2-5-05
К58

Рекомендовано до друку Вченою радою Вінницького національного технічного університету Міністерства освіти і науки, молоді та спорту України (протокол № 7 від 23. 02. 2012 р.)

Рецензенти:

Л. М. Заміховський, доктор технічних наук, професор

Й. Й. Білинський, доктор технічних наук, професор

В. В. Кухарчук, доктор технічних наук, професор

Кучерук, В. Ю.

К58 Мікропроцесори в ІВТ. Лабораторний практикум у середовищі WinPLC7 з використанням ПЛК VIPA : Лабораторний практикум / В. Ю. Кучерук, В. М. Севастьянов, І. А. Дудатьєв. – Вінниця : ВНТУ, 2013. – 82 с.

Лабораторний практикум призначений для організації лабораторних робіт з дисциплін "Мікропроцесори в ІВТ", "Основи мікропроцесорної техніки", "Мікроконтролери та мікро-ЕВМ", "Цифрова та мікропроцесорна техніка" для студентів спеціальності 6.0510001 «Метрологія та інформаційно-вимірювальні технології». Розглянуто принципи роботи у програмному пакеті WinPLC7 для програмування та діагностики програмованих логічних контролерів VIPA всіх серій.

УДК 621.3
ББК 30.2-5-05

© В. Кучерук, В. Севастьянов, І. Дудатьєв, 2013

Вступ

Програмований логічний контролер (англ. Programmable Logic Controller, PLC) або програмований контролер – спеціалізований цифровий комп'ютер, що використовується для автоматизації технологічних процесів. На відміну від комп'ютерів загального призначення ПЛК мають розвинені пристрої введення-виведення сигналів датчиків та виконавчих механізмів, які пристосовані для тривалої роботи без серйозного обслуговування, а також для роботи в несприятливих умовах довкілля.

ПЛК, як правило, не мають розвинених засобів інтерфейсу, типу клавіатури і дисплея, встановлюються в шафах, їх програмування, діагностика і обслуговування виконується програматорами – спеціалізованими пристроями, що підключаються для цієї мети або пристроями на базі РС із спеціальним програмним забезпеченням, а можливо із спеціальними інтерфейсними платами. У системах управління технологічними процесами ПЛК взаємодіють із системами людино-машинного інтерфейсу: операторськими панелями або робочими місцями операторів на базі РС. Датчики та виконавчі пристрої підключаються до ПЛК або централізовано: в “рейку” ПЛК встановлюються модулі введення-виведення, підключені до датчиків та виконавчих пристроїв окремими провідниками, або за методом розподіленої периферії, коли віддалені від ПЛК датчики та виконавчі пристрої пов'язані з ПЛК через промислову мережу. ПЛК є пристроями реального часу. На відміну від:

- мікроконтролера (однокристального комп'ютера), мікросхеми, призначеної для управління електронними пристроями, сферою застосування ПЛК зазвичай є автоматизовані процеси промислового виробництва;
- комп'ютерів, ПЛК орієнтовані на роботу з машинами і мають розвинене “машинне” введення-виведення сигналів датчиків та виконавчих механізмів на противагу можливостям комп'ютера, орієнтованого на людину (клавіатура, миша, монітор тощо);
- вбудовуваних систем – ПЛК виготовляється як самостійний виріб, окремо від керованого за його допомогою устаткування.

Перші логічні контролери з'явилися у вигляді шаф з набором з'єднаних між собою реле і контактів. Ця схема задавалася жорстко на етапі проектування і не могла бути змінена у подальшому. Перший у світі ПЛК - MOdular DIgital CONtroller (Modicon), що має пам'ять 4 кБ, виготовлений у 1968 році.

В перших ПЛК, що прийшли на заміну звичайним логічним контролерам, логіка з'єднань програмувалася схемою з'єднань LD (Ladder logic Diagram). Пристрій мав той самий принцип роботи, але реле і контакти (окрім вхідних і вихідних) були віртуальними, тобто існували у вигляді програми, що виконується ПЛК.

У системах управління технологічними об'єктами логічні команди переважають над числовими операціями, що дозволяє при порівняльній простоті мікроконтролера (шини шириною 8 або 16 біт), отримати потужні системи, які діють в режимі реального часу. У сучасних ПЛК числові операції реалізуються нарівні з логічними. В той же час, на відміну від більшості процесорів комп'ютерів, в ПЛК забезпечується доступ до окремих біт пам'яті.

Німецька компанія VIPA спеціалізується на розробці і виробництві ПЛК сумісних з контролерами фірми Siemens за системою команд. Компанія була заснована в 1985 році і пройшла шлях від невеликої інжинірингової фірми до корпорації світового масштабу з представництвами від США до Китаю. Штаб-квартира і склад готової продукції фірми VIPA знаходяться в місті Герцогенаурах (Herzogenaurach).

Сьогодні VIPA пропонує такі види продукції :

- мікро ПЛК серії System 100V;
- компактний ПЛК серії System 200V;
- ПЛК серії System 300V (програмно і апаратно сумісний з S7-300 фірми Siemens);
- процесори Speed_7 (надшвидкий процесор);
- ПЛК серії System 500V (SlotPLC, для установлення в PCI-слот розширення PC-сумісного комп'ютера);
- пристрої видаленого введення – виведення;
- різні види операторських панелей;
- програмне забезпечення.

Найбільші переваги від використання контролерів VIPA отримують ті підприємства, які застосовують системи автоматизації на базі ПЛК Siemens. У них з'явиться можливість розширювати наявні системи і створювати нові з меншими фінансовими витратами. Інженерам, знайомим з SIMATIC S7, не буде потрібно багато часу для вивчення нового устаткування.

Програмування контролерів здійснюється як за допомогою ПЗ WinPLC7 (VIPA), так і за допомогою STEP7 (Siemens). Контролери VIPA можуть працювати без карти пам'яті – її використовують, коли недостатньо об'єму пам'яті, вбудованої в процесорний модуль, або для дублювання програми на іншій ПЛК.

Крім того, однією з важливих особливостей продукції VIPA є підтримка відкритих інтерфейсів, широкоживаних в промисловості. Це створює можливість для підключення додаткових апаратних пристроїв і полегшує інтеграцію окремих виробничих ділянок в інформаційну мережу підприємства. За допомогою стандартних промислових інтерфейсів, таких як Ethernet, PROFIBUS, CANOpen, DeviceNet, INTERBUS стає можливим використовувати контролери VIPA (які підтримують ці інтерфейси) спільно з устаткуванням інших виробників.

Мета фірми – найближчим часом увійти до п'ятірки видатних світових виробників засобів управління технологічного рівня, і VIPA упевнено до цієї мети рухається. У списку замовників фірми – BMW, Volkswagen, Audi, BASF, Bayer, Continental, Thyssen і багато інших найбільших підприємств Німеччини практично в усіх галузях промисловості.

Різке зростання виробництва почалося останніми роками, коли VIPA прийняла рішення про вихід на світовий ринок. На експорт продукція VIPA йде як під своїм ім'ям, так і у рамках OEM – співробітництва під логотипом замовника. Зокрема, свій логотип ставлять на контролери VIPA такі відомі марки як німецька фірма Lenze – відомий виробник електроприводів, італійська фірма Gefran – відомий виробник устаткування для обробки пластмас, і найбільший китайський системний інтегратор в області промислової автоматизації HollySys.

В.1 Загальні відомості та технічні характеристик ПЛК VIPA

Завдяки своєму компактному дизайну та оптимальному співвідношенню ціна/продуктивність контролери серії System 100V добре підходять для систем з невеликою кількістю точок введення-виведення. Однак сумісність із SIMATIC S7_300 за набором інструкцій і розвиненням комунікаційних можливостей дозволяють використати їх й у досить складних завданнях, що потребують розподіленого керування, у тому числі в комбінації з іншими контролерами фірми VIPA і третіх виробників.

До складу сімейства входять кілька моделей контролерів з вбудованими каналами введення-виведення і з підтримкою функцій формування сигналів переривання, швидких лічильників та імпульсних виходів. Кількість каналів введення-виведення може бути збільшене за допомогою модулів розширення.

У це ж сімейство входять модулі розподіленого введення-виведення для мереж PROFIBUS й CANOpen (рисунок 1). Процесорні модулі й модулі розширення монтуються безпосередньо на 35-міліметрову DIN-рейку.



Рисунок 1 – Серія System 100V: контролер CPU 115 (ліворуч) і модуль розподіленого введення- виведення SM123 з інтерфейсом PROFIBUS (праворуч)

Модулі System 200V можна застосовувати для розширення систем автоматизації, виконаних на базі даних контролерів фірми Siemens і промислової шини PROFIBUS, користуючись при цьому єдиним інструментом розробки. За допомогою System 200V можна також створювати нові високопродуктивні системи керування, що відповідають сучасним вимогам. При цьому функціональні можливості модулів VIPA не поступаються «оригіналу», а конструктивно вони набагато компактніші. Програмне забезпечення для контролерів серії System 200V може створюватися як за допомогою стандартного середовища розробки STEP7, так і за допомогою застосування недорогих пакетів з обмеженою функціональністю WinPLC7 й WinNCS.

Однією з головних особливостей контролерів цієї серії є універсальність. Добре відомо, що використання розподіленої архітектури АСУ на базі цифрових інтерфейсів передачі даних має цілий ряд переваг перед рішеннями на основі класичної централізованої архітектури. Серед них можна згадати зниження витрат на розгортання й обслуговування кабельної мережі, підвищення надійності за рахунок зменшення кількості з'єднань, поліпшення завадостійкості, переваги при розширенні системи. Однак централізована архітектура й зараз знаходить своє застосування там, де кількість каналів не дуже велика і всю систему керування можна зосередити на невеликій площі. За допомогою модулів System 200V можна створювати системи збору даних і керування як із централізованою, так і з розподіленою архітектурою.

Іншою важливою особливістю є підтримка відкритих інтерфейсів, широкозастосовуваних у промисловості. Наявність комунікаційних модулів для Ethernet створює можливість для підключення додаткових апаратних засобів, наприклад, панельних комп'ютерів для побудови людино-машинного інтерфейсу, і полегшує інтеграцію окремих виробничих ділянок в інформаційну мережу підприємства. За допомогою стандартних промислових інтерфейсів, таких як PROFIBUS, DeviceNet й CANOpen, стає можливим використати пристрої серії System 200V разом з устаткуванням інших виробників.

Серія System 200V побудована за модульним принципом. Це значить, що користувач має можливість оптимально підбирати склад модулів для рішення свого завдання й гнучко модифікувати його при розширенні або при зміні вимог до системи. Всі модулі введення-виведення й інтерфейсні модулі мають властивість універсальності, тобто, їх можна застосовувати разом з будь-яким CPU даної серії. При цьому є можливість вибору процесорного модуля з оптимальною продуктивністю для рішення конкретного завдання.

Таким чином, контролери VIPA серії System 200V можуть бути використані скрізь, де застосовуються класичні ПЛК і підсистеми розподіленого введення-виведення. Вони мають швидкий час реакції й підходять для керування виробництвами дискретного, безперервного й

дискретно-безперервного типу. Застосування пристроїв цієї серії дозволить легко розширити систему керування, додаючи до неї окремі модулі введення-виведення, станції розподіленої периферії і нові програмовані контролери, що мають можливості масштабування. Зовнішній вигляд контролера VIPA серії System 200V подано на рисунку 2.



Рисунок 2 – Контролер VIPA серії System 200V

У сімейство VIPA System 200V входить кілька моделей процесорних модулів з різними комунікаційними можливостями. Кожна модель існує в трьох версіях, що відрізняються одна від одної обсягом пам'яті: CPU 214x, CPU 215x й CPU 216x. Всі моделі забезпечують підключення до своєї локальної магістралі до 32 модулів введення-виведення.

Процесорні модулі й модулі розширення монтується безпосередньо на 35-міліметровій DIN-рейці. Їх підключення до системної шини реалізується за допомогою складальних з'єднувачів на одне, два, чотири або вісім посадкових місць, що містяться в поглибленні рейки. Це дозволяє позбутися використання спеціальної об'єднувальної плати, забезпечуючи при цьому необхідну гнучкість конфігурування системи, коли користувач може підібрати стільки з'єднувачів, скільки йому необхідно.

Всі моделі мають годинники реального часу, а також літєвий акумулятор, що забезпечує зберігання даних в енергонезалежному ОЗУ й роботу протягом 30 днів. Як зовнішня пам'ять для зберігання програм, даних і вихідних текстів, а також архівів використовуються звичайні карти флеш-пам'яті у форматі Multimedia Card (MMC).

Як засіб зовнішньої пам'яті можна використати звичайну карту флеш-пам'яті у форматі Multimedia Card (до 64MB) від VIPA (VIPA 953-0KX10), що збільшує простір пам'яті для зберігання програм, даних і вихідних текстів, а також архівів. Доступ до MMC завжди можливий після повного скидання. Також в VIPA доступно зовнішній MMC пристрій читання та запису (VIPA 950-0AD00).

CPU мають вбудоване електроживлення. Підключення до трьох клем на фронтальній стороні. Напруга живлення повинна бути в межах DC 24V (20...30V). Електроживленням забезпечуються як електронна частина CPU, так і всі приєднані модулі через шинний з'єднувач.

Додатково до захищеного RAM, CPU 11x мають вбудовану Flash-ROM, що збільшує обсяг оперативної пам'яті. Вміст Flash-ROM не

видаляється при повному скиданні (OVERALL RESET). Flash-ROM може бути очищений за допомогою запиту повного скидання (OVERALL RESET) і перенесене в порожню оперативну пам'ять із Flash-ROM через PLC функцію Copy RAM to ROM. Повідомлення про помилку відбувається тоді, коли ви ініціюєте команду запису в MMC.

Процесорний модуль CPU 214, зовнішній вигляд якого подано на рисунку 3, а основні технічні характеристики наведено у таблиці 1.



Рисунок 3 – Процесорний модуль CPU 214

Серія VIPA System 200V включає у свій склад модульні контролери, які з однаковим успіхом підходять для створення як централізованих, так і розподілених систем керування. Вона відрізняється більшою різноманітністю процесорних модулів, орієнтованих на застосування в системах керування середньої величини (від 100 до 1000 сигналів введення-виведення), і модулів розширення для них. CPU 214...216 сумісні за набором інструкцій з популярними контролерами SIMATIC S7-300 і можуть програмуватися як за допомогою ПЗ WinPLC7 (VIPA), так і за допомогою STEP 7 (Siemens), а модулі CPU 241...244 програмуються за допомогою пакета STEP 5. Крім процесорних модулів, до складу серії входять модулі введення-виведення, інтерфейсні модулі для PROFIBUS_DP, CANOpen, DeviceNet, Interbus_S й RS-485, а також пристрої розподіленої периферії для цих промислових мереж. І, на кінець, у цю же серію входить CPU 288, що являє собою компактний IBM PC сумісний контролер і може програмуватися мовами високого рівня.

Процесорні й периферійні модулі System 300V (рисунок 4) апаратно й програмно сумісні з S7-300 і підходять для створення систем керування з більшою кількістю сигналів. Всі процесорні модулі цієї серії мають вбудований інтерфейс PROFIBUS_DP, а деякі з них оснащені також інтегрованим контролером Ethernet.



Рисунок 4 – Контролер VIPA серії System 300V

Контролери серії System 500V являють собою так званий Slot-PLC-карту у форматі PCI, що встановлюється в корпус промислового комп'ютера й оснащена інтерфейсом PROFIBUS-DP (рисунок 5).



Рисунок 5 – Slot-PLC CPU 517

Завдяки великому обсягу пам'яті й високій продуктивності ці контролери здатні вирішувати складні завдання, а шина PCI дозволяє при цьому з високою швидкістю вести обмін даними з комп'ютером. На платі є роз'єми для підключення зовнішнього джерела живлення, що дозволяє контролеру вирішувати свої завдання навіть тоді, коли комп'ютер виключений. Програмування такого контролера можна здійснювати за допомогою STEP 7.

Заслуговує на особливу увагу остання розробка фірми - серія високошвидкісних CPU, сумісних з STEP 7, які побудовані на базі мікросхеми Speed 7 фірми ProfiChip GmbH. Ця фірма була створена в 1997 році як підрозділ VIPA із розробки інтегральних схем, а потім стала самостійною.

Такі CPU мають найшвидший на сьогоднішній день процесор, що підтримує систему інструкцій S7-300/S7-400: він виконує операцію із плаваючою комою всього за 0,084 мікросекунди, а операцію над бітом або словом - за 0,014 мікросекунди. Завдяки його використанню час циклу ПЛК може скласти всього 100 мікросекунд. У серію високошвидкісних CPU входять процесорні модулі CPU 315S, 314S й 317S, а також Slot_PLС CPU 517S; причому модулі CPU 314S й 317S оснащені високошвидкісною паралельною шиною SpeedBus для обміну з модулями розширення.

Крім усього перерахованого, фірмою VIPA пропонуються процесорні і периферійні модулі для SIMATIC S5, широкий набір текстових дисплеїв і графічних операторських панелей, призначених для створення людино-машинного інтерфейсу, а також системи розробки програмного забезпечення для контролерів VIPA та Siemens.

Характеристики для процесорних модулів різних версій, наведені в таблиці 1 та 2.

Таблиця 1- Загальні характеристики процесорних модулів серії System 200V

	CPU 214	CPU 215	CPU 216
Системні дані			
Постійна пам'ять	48 кБайт	96 кБайт	128 кБайт
Оперативна пам'ять	80 кБайт	144 кБайт	192 кБайт
Розширювана пам'ять	ММС		
Батарейка (акумулятор)	так		
Дата й час	Годинники реального часу		
Програмування	STEP7 або WinPLC7		
Час виконання:			
- логічної операції	0.18 мкс		
- операцій зі словами	0.78 мкс		
Типи блоків, OB – організаційні, FB – функціональні, FC – функції, DB – даних	OB1/10/20/35/40/80/82/85/87/100/121/1 22 FB: до 1024; FC: до 1024; DB: до 2047		
Бітова пам'ять	2048 біт		
Таймери	128		
Лічильники	256		
Інтерфейси	MP ² I		
Адресний простір	Входи: 1024 байти, Виходи: 1024 байти		
Відображення процесу	Входи: 128 байтів, Виходи: 128 байтів		
Кількість входів/виходів			
- дискретних каналів	1024		
- аналогових каналів	128		

Таблиця 2 – Комунікаційні можливості різних процесорних модулів серії System200V

	Обсяг ОЗУ/ЭППЗУ, кбайт			
	32/40	48/80	96/144	128/192
Тільки MP2I	CPU	CPU	CPU 215	CPU 216
MP21 й PROFIBUS-DP (Slave)	—	CPU 214DP	CPU 215DP	CPU 216DP
MP21 й PROFIBUS-DP (Master)	—	CPU 214DPM	CPU 215DPM	CPU 216DPM
MP2I й Industrial Ethernet	—	CPU 214Ne	CPU 215Ne	CPU 216Ne
MP2I й CANOpen (Master)	—	CPU 214Ca	CPU 215Ca	CPU 216Ca

Модулі оснащені знімними пружинними клемми WAGO, що допускають підключення проводів перерізом від 0,08 до 2,5 мм². Для полегшення їх розведення до складу системи входять термінальні модулі CM 201. Кожен процесорний модуль оснащений роз'ємами MP2I, що поєднує в собі два інтерфейси: MPI й RS232. MPI являє собою багатоточковий інтерфейс, реалізований на базі специфікацій стандарту EIA RS-485 і застосовуваний для обміну даними між контролерами S7-300 й S7-400, операторськими панелями, програматорами й персональними комп'ютерами, у тому числі при завантаженні програм. Для підключення комп'ютера до контролера за допомогою MPI необхідний спеціальний адаптер. Підтримка інтерфейсу RS232, реалізована на незадіяних контактах роз'єма MPI, дозволяє встановити з'єднання «точка – точка» з комп'ютером без використання MPI адаптера, за допомогою особливого «зеленого кабелю» (green cable). Він являє собою кабель зелених кольорів, призначений спеціально для контролерів VIPA.

За допомогою такого кабелю здійснюються завантаження й налагодження проектів в CPU 11x, 21x, 31x, 51x й обмін даними з ними, наприклад для задач візуалізації. Ще одне його призначення - завантаження конфігурації в інтерфейсні модулі IM 208, а також відновлення системного ПЗ “прошивань” комунікаційних і процесорних модулів.

Використання розподілених систем введення-виведення дозволяє одержати переваги на етапах розгортання, експлуатації й модернізації системи. Для підключення периферійних пристроїв можуть бути використані як спеціальні інтерфейсні модулі, так і процесорні модулі, які можуть підтримуватися стандартними промисловими інтерфейсами. Процесорні модулі CPU 21xDPM, крім інтерфейсу MP2I, містять також

вбудований комунікаційний модуль PROFIBUS_DP (Master) і допускають підключення до 125 ведених вузлів. Нові модулі CPU 21xCAN мають вбудований інтерфейс CANOpen. Веденими вузлами можуть бути модулі розподіленої периферії серій 100V, 200V й 300V, а також інших виробників, таких як WAGO й Siemens. Крім того, це можуть бути інші контролери, операторські панелі, інтелектуальні датчики, пристрої керування частотними приводами, абсолютні енкодери.

Моделі CPU 21xDP оснащені вбудованим інтерфейсом PROFIBUS_DP (Slave) і застосовуються для створення інтелектуальних ведених вузлів. У порівнянні зі звичайними вузлами розподіленої периферії інтелектуальні ведені вузли дозволяють одержати ряд переваг. В першу чергу, можна з малим часом реакції відпрацьовувати на них локальні алгоритми керування й знижувати в такий спосіб навантаження на основний процесорний модуль, скорочуючи його час циклу. Крім того, за рахунок попередньої обробки сигналів, наприклад фільтрації, логічних операцій або підрахунку кількості імпульсів, з'являється можливість скоротити обсяг даних, переданих по мережі. Це дозволяє збільшити швидкість обміну й підвищити його надійність. Нарешті, у випадку виходу з ладу головного процесорного модуля можна за допомогою введеного контролера автоматично перевести керуючі сигнали в безпечний стан і у такий спосіб уникнути аварії.

В останні кілька років у промисловості все більше використовують мережу Ethernet. Такі рішення мають цілий ряд переваг. Зокрема, вони легко вводяться в експлуатацію й обходяться порівняно недорого, оскільки Ethernet широко застосовується повсюдно. Підтримка цього інтерфейсу давно стала обов'язковою для персональних комп'ютерів, і підключення сумісного з ним устаткування практично не потребує додаткових витрат. Крім того, Ethernet має високу пропускну здатність і базується на відкритих стандартах, які добре знайомі багатьом фахівцям. Нарешті використання Ethernet полегшує інтеграцію систем автоматизованого керування в існуючі системи масштабу підприємства й відкриває дорогу об'єднанню окремих підприємств за допомогою глобальної мережі Internet.

На даному етапі розвитку саме Ethernet може стати основою для вертикальної інтеграції всіх систем підприємства, починаючи від рівня низової автоматики, контролерів, робочих місць операторів і закінчуючи системами керування виробництвами планування ресурсів підприємства. Якщо раніше Ethernet використалася для зв'язку операторських станцій і серверів масштабу підприємства, у той час як на низовому рівні панували порівняно повільні мережі на базі RS485, то тепер існує можливість об'єднати всі ці пристрої в рамках єдиної мережі. Таким чином, дані від технологічного устаткування можуть по єдиній інформаційній магістралі надходити саме туди, де вони потрібні в даний момент, минаючи проміжні ланки.

У зв'язку із цим інтерес становлять процесорні модулі CPU 21xNET з вбудованим інтерфейсом Industrial Ethernet, що підтримують протоколи TCP/IP й SINEC H1.

Протокол SINEC H1 базується на стандарті Ethernet і призначений для забезпечення надійного обміну даними між контролерами. Інформація передається між станціями за допомогою фреймів H1. Передача кожного фрейму обов'язково завершується підтвердженням, що гарантує вірогідність прийнятої інформації, але негативно позначається на швидкості обміну.

До того ж цей протокол не можна використати для передачі даних через маршрутизатори. На відміну від SINEC H1, протокол TCP/IP є загальноприйнятим стандартом і найбільше підходить для зв'язку з персональними комп'ютерами. З його допомогою зручно передавати дані на верхній рівень для візуалізації й архівування, а також вести обмін між контролерами, що перебувають у різних сегментах мережі. Для створення конфігурації мережного інтерфейсу при використанні обох протоколів обміну застосовується програма WinNCS.

В.2 Інтерфейсні модулі

Серія інтерфейсних модулів ІМ 253 призначена для створення децентралізованих систем керування з використанням вузлів розподіленої периферії. У неї входять ведені пристрої для Ethernet PROFIBUS_DP, CANOpen, DeviceNet, SERCOS й Interbus. Кожен ведений модуль (coupler) підтримує підключення до 32 модулів розширення.

За допомогою інтерфейсного модуля ІМ 253NET стає можливим підключати з використанням Ethernet вилучені периферійні пристрої. Він підтримує протоколи ModBus TCP, Siemens S5 Header, HTTP і може використовуватись як вилучений УСО для контролерів SIMATIC S7_400 з комунікаційним процесором CP 443, SIMATIC S5, з комунікаційним процесором CP 143, а також VIPA CPU 21xNET. Крім того, доступ до нього можна одержати за допомогою OPC-сервера, прикладної програми або навіть звичайного веб-браузера. Допускається одночасне підключення до 8 клієнтів.

Мережа Ethernet реалізована на коаксіальному кабелі основана на шинній топології, та крученій парі (вита пара), яка основана на двоточковому з'єднанні. Вита пара - це кабель із чотирма провідниками скрученими попарно. Кожен провідник може мати діаметр від 0,4 до 0,6 мм, максимальна кількість з'єднаних елементів на сегмент – 2, максимальна довжина сегмента – 100 м.

Вита пара не дуже піддається зовнішньому впливу. З'єднувач ІМ 253NET для мережі Ethernet має роз'єм під виту пару.

Мережа реалізована за допомогою цього кабелю має топологію «зірка». Кожна станція з'єднується з хабом/свічем через окремий кабель. Хаб/свіч забезпечує Ethernet інтерфейс.

Хаб – це центральний елемент, що необхідний для створення мережі Ethernet на витій парі. Він працює для прийому, посилення й передачі сигналів у всіх напрямках. Хаб не доступний за допомогою окремої мережної адреси, тому що він не видимий станціям у мережі. Хаб має додатковий інтерфейс для Ethernet або іншого хаба.

Свіч – це центральний елемент для здійснення мережі Ethernet на витій парі. Кілька станцій приєднуються разом до хабу через свіч. При цьому вони можуть контактувати один з одним не завантажуючи всю мережу.

Ethernet підтримує принцип прямого доступу до шини: будь-яка станція мережі звертається до шини в будь-який час, коли потрібно. Цей доступ координується CSMA/CD схемою: кожна станція "слухає" на шині й одержує повідомлення з'єднання, які адресовані для неї.

Станції ініціалізують передачу тільки тоді, коли не зайнята лінія. У випадку, коли два учасники повинні запустити передачу одночасно, вони виявляють це, зупиняються, і відновляють її після того як пройде випадково вибраний час затримки.

Ethernet-з'єднувач по'єднується з модулями через шинний з'єднувач. При цьому він збирає дані підключених модулів і розміщає їх як "сервер".

Для конфігурування інтерфейсного модуля IM 253NET необхідна програма WinNCS. Цікавою можливістю є застосування модуля IM 253DN для створення шлюзу між мережами DeviceNet й PROFIBUS. Необхідно як модуль розширення для нього встановити PROFIBUS Master IM 208DP. Через таке з'єднання може бути передано до 256 байт вхідних й 256 байт вихідних даних.

Істотно полегшити розгортання мережі Ethernet дозволяє чотирипортовий мінікомутатор Ethernet CM 240, встановлюваний на DIN-рейку разом з модулями серії 200V або окремо.

Крім ведених інтерфейсних модулів, у серії подані також майстер-модулі для інтерфейсів PROFIBUS_DP й CANOpen. З їхньою допомогою можна нарощувати кількість сегментів з розподіленими периферійними пристроями, підключених до контролера. Модулі IM 208CAN, IM 208DP й IM 208DPO підтримують підключення до 126 периферійних пристроїв, інші їх характеристики аналогічні відповідним веденим модулям.

Ще одна група модулів – комунікаційні процесори – призначена для підключення до контролерів або станцій розподіленого введення – виведення пристроїв з послідовними інтерфейсами RS232, RS485 або «струмова петля».

Таблиця 3 – Загальні характеристики інтерфейсного модуля для Ethernet

Модуль	IM 253NET
Підтримувані протоколи	ModBus TCP, Siemens S5
З'єднувач	RJ-45
Мережна топологія	"Зірка"
Середовище передачі	Кручена пара
Швидкість передачі	10/100 Мбіт/з
Максимальна довжина лінії	100 м на сегмент
Максимальне число вузлів мережі	1024
Максимальне число модулів	32, у тому числі до 16 аналогових
Вхідних даних	256 байт
Вихідних даних	256 байт
Електричні дані	
Напруга живлення	DC 24V (20.4...28.8V)
Поточна сила струму	120m
Вихідний струм споживаний шиною	3.5A
Поділ потенціалів	> AC 500V
Розміри	25.4·76·76 мм
Вага	70 г

Комунікаційні процесори CP 240 існують у двох основних варіантах, один із яких забезпечує підключення пристроїв з інтерфейсами 3964R, RK512, STX/ETX й ASCII, а інший – підтримує протоколи ModBus ASCII й ModBus RTU і може працювати в режимах Master (ведучий) і Slave (ведений).

В.3 Модулі введення – виведення

Серед переваг модулів дискретного введення й виведення VIPA System 200V варто назвати те, що всі вони забезпечують гальванічну ізоляцію до 500 В між вхідними сигналами й внутрішньою шиною контролера, а також оснащені світлодіодними індикаторами стану входів. Крім того, навантажувальна здатність дискретних виходів більшості модулів вища, ніж в аналогічних модулях інших фірм, наприклад Siemens.

Більшість модулів аналогового введення мають можливість налаштування й конфігурування, дозволяючи підключати різноманітні датчики та інші джерела сигналів. SM 231_1BD52 – чотириканальний модуль із роздільною конфігурацією каналів має можливість введення сигналів термоопорів Pt 100, Pt 1000, Ni 100, Ni 1000 як із двопровідним, так і із чотирипровідним підключенням. Він також дозволяє вимірювати опори в діапазонах 0...60 Ом, 0...600 Ом. Використання SM 231_1BD52

дозволяє здійснювати введення інформації з таких датчиків шляхом програмного перерахування значення опору в температуру за градуовальною таблицею. Крім того, можливе підключення термопар типів J, K, N, R, T й S як із зовнішньою, так і із внутрішньою компенсацією температури холодного спаю, а також вимір напруг у діапазонах 0...50 мВ, ±10 В, ±4 В, ±400 мВ і струмів у діапазонах ±20 мА й 4...20 мА. Модуль дозволяє гнучко набудувувати час перетворення й проводити згладжування вхідного сигналу. Таким чином, один і той же модуль можна використати для рішення різних завдань.

Модулі розширення System 200V. Дискретні модулі

Компактні дискретні вхідні й вихідні або вхідні/вихідні модулі забезпечують можливості розширення System 200V.

Серед переваг модулів дискретного введення й виведення VIPA System 200V варто назвати те, що всі вони забезпечують гальванічну ізоляцію до 500 В між вхідними сигналами й внутрішньою шиною контролера, а також оснащені світлодіодними індикаторами стану входів. Крім того, навантажувальна здатність дискретних виходів більшості модулів вища, ніж в аналогічних модулях інших фірм, наприклад Siemens.

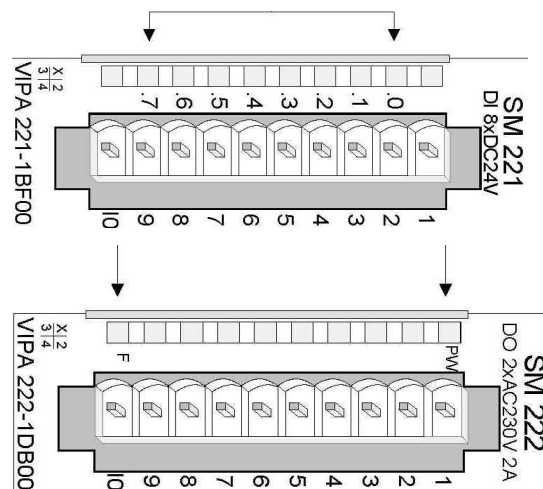


Рисунок 8 – Позначення клем SM 222

Аналогові модулі розширення

Більшість модулів аналогового введення мають багаті можливості налаштування й конфігурування, дозволяють підключати різноманітні датчики й інші джерела сигналів.

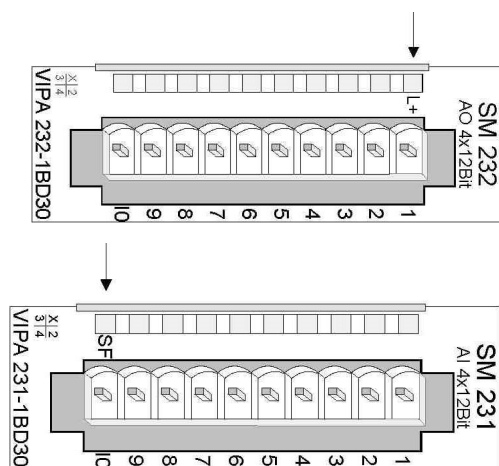


Рисунок 9 – Позначення клем SM 222

В.4 Джерело живлення

Для забезпечення високої стабільності та безперебійної роботи системи живлення елементів має бути стабільним. Враховуючи особливості конструкції лабораторного станда, зупинимо свій вибір на блоці живлення SPD 24181 (рисунок 6).



Рисунок 6 – Блок живлення SPD 24181

Джерела живлення серії SPD монтуються на DIN-рейці або монтуються прямо на панелі без зменшення вихідної потужності у великому температурному діапазоні від -10 до $+60$ °C. Вихідну напругу можна точно відрегулювати. Вихід захищений від перенапруги й короткого замикання. Відповідає стандартам ULus й VDE. Технічні характеристики джерела живлення SPD 24181 наведено у таблиці 4.

Таблиця 4 – Технічні характеристики джерела живлення SPD 24181

Вхідна напруга	85–264 В
Потужність	18 В
Вихідна напруга	24 В
Підстроювання вихідної напруги	від -10 до $+15$ %
Пульсації	Максимум 2 %
Підстроювання навантаження	± 2 %
Підстроювання лінійності	± 1 %
Захист від радіоперешкод	EN61204-2, EN55011 клас B
Промисловий стандарт	EN/UL60950
Розміри	90 · 22.5 · 115 мм

На рисунку 7 зображена структурна схема блока живлення SPD 24181.

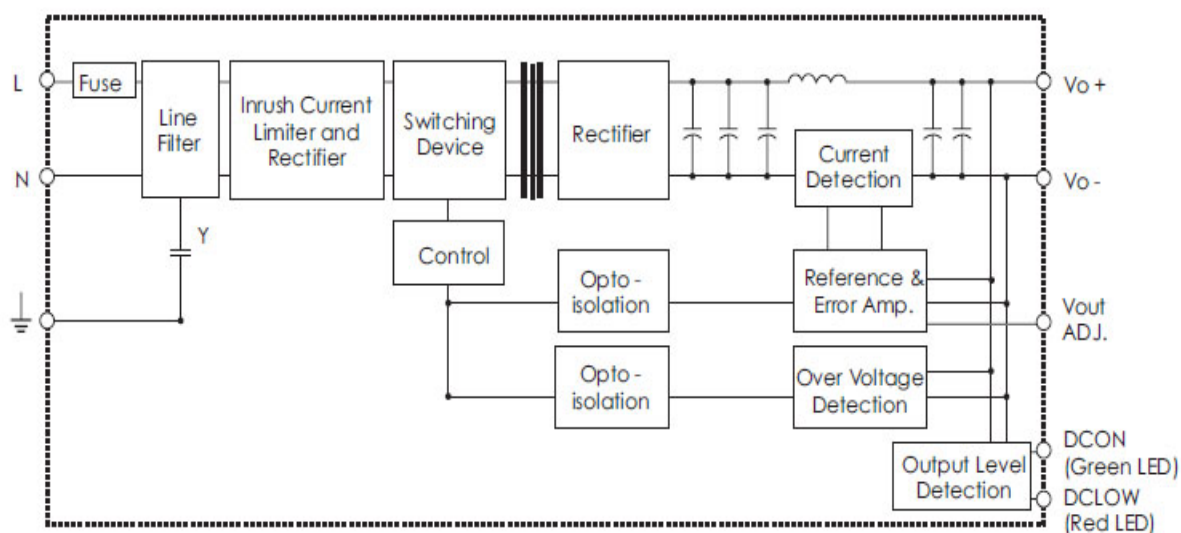


Рисунок 7 – Структурна схема блока живлення SPD 24181

В.5 Блок підключення зовнішніх сигналів

Для того щоб отримати змогу задавати й зчитувати дані з ПЛК було розроблено блок підключення зовнішніх сигналів, що зображений на рисунку 10. Взагалі це панель на якій розміщені всі необхідні для роботи елементи.



Рисунок 10 – Блок для підключення зовнішніх сигналів

Блок являє собою модульну структуру, що складається зі структурних модулів введення, виведення. Завдяки клемам до цих модулів під'єднуються зовнішні пристрої і відбувається подальша обробка сигналів, що надходять з датчиків. Також на панелі окрім вищезгаданих клем розміщені два вольтметри, елементи індикації та перемикачі станів з логічного нуля в стан логічної одиниці.

Лабораторна робота № 1

ОСНОВИ РОБОТИ У СЕРЕДОВИЩІ WinPLC7 З ВИКОРИСТАННЯМ ПЛК VIPA СЕРІЇ 200 V

Мета роботи: ознайомитися з особливостями програмування контролерів фірми VIPA у середовищі розробки WinPLC7.

1.1 Основні теоретичні відомості

Програмний пакет WinPLC7 для конфігурування, програмування, налагодження програм та діагностики контролерів VIPA всіх серій. Пакет має “дружній інтерфейс” для всіх етапів роботи.

WinPLC7 містить всі необхідні інструменти для створення проекту: конфігуратор апаратури, яка використовується, символний редактор, конфігуратор мережі PROFIBUS, редактор програм, емулятор контролера.

Для програмування систем автоматизації в рамках пакета WINPLC7 можуть бути використані три мови [1] : Statement List (STL) – список інструкцій, Ladder Diagram (LAD) – мова релейно-контактних схем, Function Block Diagram (FBD) – мова функціональних блоків.

WINPLC7 [6] дозволяє імпортувати/експортувати проекти для контролерів фірми VIPA (Siemens), зберігати резервну копію програми та даних на MMC-карту, а також здійснювати програмну симуляцію роботи контролера. Для контролерів System 200V ліцензія на пакет WINPLC7 безкоштовна.

System 200V [7] – сімейство ПЛК від VIPA, що призначене для розв’язування задач керування та регулювання в системах автоматизації нижнього і середнього рівнів складності. Завдяки своєму компактному дизайну і вигідному співвідношенню ціна – продуктивність контролери серії System 200V особливо добре підходять для застосування з невеликою кількістю точок введення – виведення.

Для параметризації і керування мережними конфігураціями використовується оболонка VIPA WinNCS. Вона дозволяє створювати проект мережі Profibus-DP з використанням контролерів VIPA та Siemens і здійснює параметризацію обладнання та контролерів, що підтримують TCP/IP для мережі Ethernet.

МікроПЛК серії System 200V має такі особливості:

- блоково-модульна конструкція;
- системи до 160 каналів введення – виведення;
- час виконання логічної операції 0,25 мкс;
- час виконання операції зі словами 1,2 мкс;
- годинник реального часу;

- підключення до 4 модулів розширення серій System 100V і System 200V;
- напруга живлення: 24 В постійного струму;
- від'ємні клемні з'єднувачі з пружинним контактом у комплекті постачання;
- гнучка конфігурація каналів введення – виведення;
- монтаж на 35 мм DIN рейку;
- діапазон робочих температур: від 0 до +60 градусів °С.

1.1.1 Світлодіоди стану

ПЛК VIPA 200 має 7 світлодіодів на передній панелі, що слугують для діагностики та моніторингу стану (таблиця 1.1).

Таблиця 1.1 – Ідентифікація світлодіодів стану

Мітка	Колір	Опис
PW	Жовтий	Ввімкнений ПЛК
R	Зелений	ПЛК знаходиться в режимі RUN
D	Жовтий	*Лише в ПЛК 11· DP D (Date exchange) показує активність з'єднання по ширині PROFIBUS.
S	Червоний	ПЛК знаходиться в режимі STOP
SF	Червоний	Миготить, коли в системі виявлена помилка (апаратна поломка)
FC	Жовтий	Миготить, якщо змінні завантажені
MC	Жовтий	Миготить, коли використовується карта пам'яті

1.1.2 MPI інтерфейс

Роз'єм MP21, що встановлений на ПЛК 115, об'єднує в собі два інтерфейси: MPI та RS-232. MPI являє собою багатоточковий інтерфейс, реалізований на базі стандарту RS-485 і застосовується для обміну даними між контролерами, операторськими панелями і програматорами.

Для під'єднання комп'ютера до контролера за допомогою MPI необхідний спеціальний адаптер. Підтримка інтерфейсу RS-232, що реалізована на незадіяних контактах роз'єму MPI, дозволяє встановити з'єднання "точка – точка" з комп'ютером без використання MPI-адаптера, за допомогою "Green Cable".

1.1.3 Потенціометр та MMC слот

На передній панелі ПЛК є два потенціометри, які застосовуються для прямого введення аналогового значення, і можуть бути використані як датчики або з іншою метою.

Застосовується для карт пам'яті MMC, які слугують для зберігання програм. Звернення до карти пам'яті іде після перезавантаження ПЛК-MRST.

Позначення, які використовуються при програмуванні ПЛК у середовищі WinPLC7 з використанням мови релейно-контактних схем (таблиця 1.2).

Таблиця 1.2 – Основні позначення мови релейно-контактних схем

-- --	Нормально відкритий контакт
-- / --	Нормально замкнений контакт
-- NOT --	Інверсія результату логічної операції
--()	Вихідна котушка
--(#)	Конвектор
--(R)	Скидання біту
--(S)	Встановлення біту
RS	RS тригер
SR	SR тригер
--(N)--	Виділення від'ємного фронту RLO
--(P)--	Виділення додатного фронту RLO
--(SAVE)	Збереження RLO в біті BR
NEG	Виділення від'ємного фронту сигналу
POS	Виділення додатного фронту сигналу

1.2 Порядок виконання роботи

Щоб почати роботу з програмою WinPLC7 необхідно запустити її за допомогою ярлика "WinPLC7", що знаходиться на робочому столі.

Після того, як програма відкриється (рисунок 1.1), необхідно створити новий проект. Для цього необхідно вибрати меню File >> Open\create a project, або це можна зробити, натиснувши поєднання клавіш Ctrl+0. У вікні, що відкрилося, буде показаний список вже існуючих проектів, також можна створити свій власний проект, натиснувши на кнопку "New

project” . Після чого буде запропоновано ввести назву проекту, наприклад, назвемо його MPA_1_Lab_1. Після цих дій проект з’явиться у списку вже існуючих проектів, Вам залишилося лише вибрати ваш проект зі списку та натиснути кнопку ”Open” .

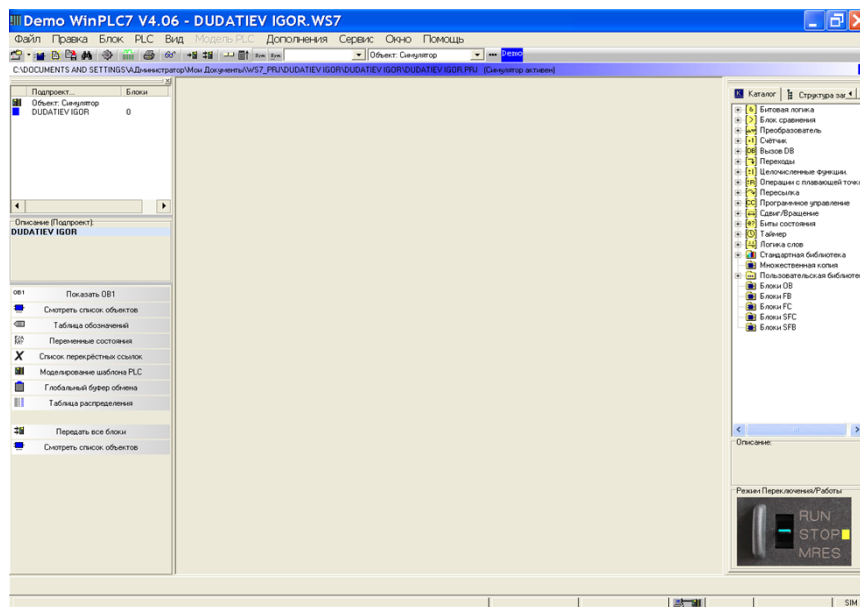




Рисунок 1.1 – Головне вікно WinPLC7

Основні функціональні команди WinPLC7 подано у таблиці 1.3.

Таблиця 1.3 – Функціональні команди WinPLC7

Команда	Опис
OB1 Показать OB1	Переключення на організаційний блок OB 1. Якщо блока не існує, він буде створений
 Смотреть список объектов	Менеджер блоків. Дозволяє відкрити існуючий блок, створити новий блок, видалити блок, порівняти два блоки, копіювати блок. Також дозволяє змінити версію блока, автора, для блоків даних можна заборонити зміни блока
PRJ	Менеджер керування підпроектами. Дозволяє створювати, змінювати ім'я, копіювати та видаляти підпроекти
 Таблица обозначений	Редагування таблиці символів. Призначення імен користувачів змінним блокам
 Моделирование шаблона PLC	Емулятор зовнішнього вигляду ПЛК
 Таблица распределения	Показує входи і виходи ПЛК, які використовуються

WinPLC7 – це просте та зручне програмне забезпечення, з добре узгодженим інтерфейсом, яке призначено для розробки програм і обслуговування програмованих контролерів VIPA. Це програмне забезпечення може використовуватися для рішення широкого кола задач автоматизації.



Рисунок 1.2 – Програмне забезпечення WinPLC7V5

Hardware configuration – порграма для зв'язку між контролером та комп'ютером. Для того, щоб запустити Hardware configuration, необхідно у меню WinPLC7 натиснути: PLC → Hardware configuration (рисунок 1.3). Після завантаження програми необхідно вибрати вигляд вашого PLC.

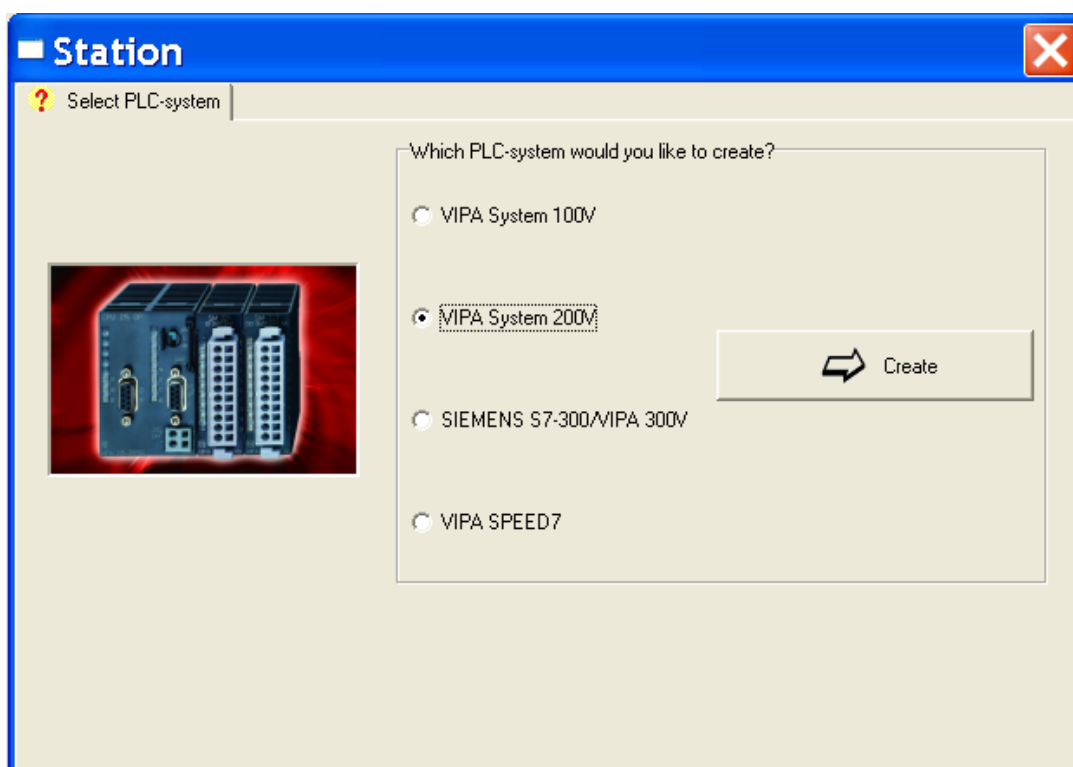
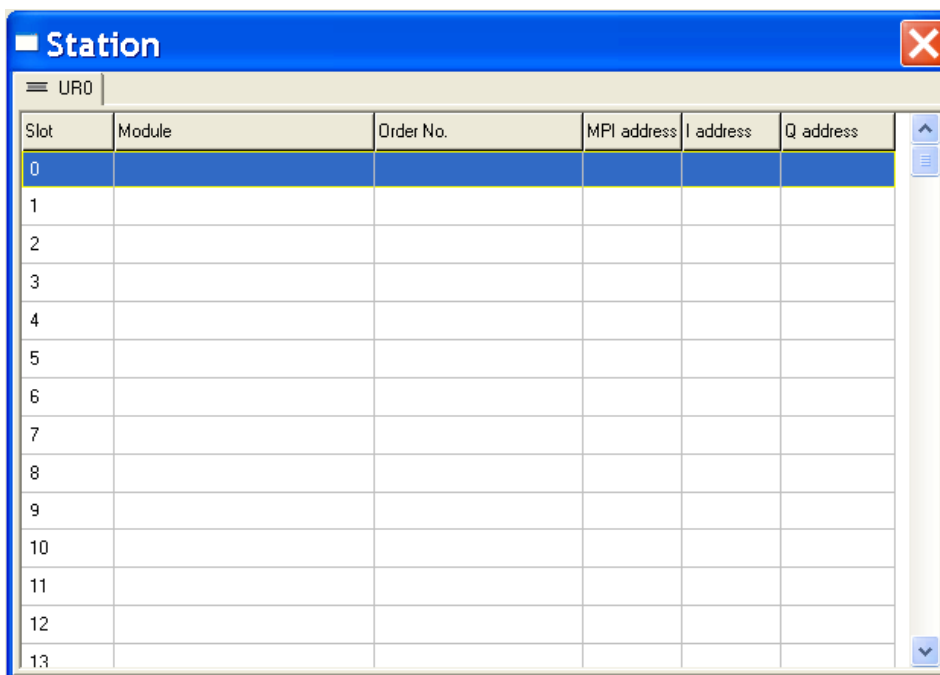


Рисунок 1.3 – PrtSc Hardware configuration

Після вибору серії ПЛК необхідно заповнити таблицю (рисунок 1.4):



The screenshot shows a software window titled 'Station' with a sub-header 'UR0'. It contains a table with the following columns: Slot, Module, Order No., MPI address, I address, and Q address. The table has 14 rows, numbered 0 to 13. Row 0 is highlighted in blue. The rest of the table is empty.

Slot	Module	Order No.	MPI address	I address	Q address
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					

Рисунок 1.4 – Таблиця конфігурації модулів ПЛК

У таблиці необхідно заповнити назви модулів, які позначені безпосередньо на блоках контролера, який Ви використовуєте. Для зручного та швидкого заповнення конфігурації необхідно використовувати таку таблицю (рисунок 1.5):

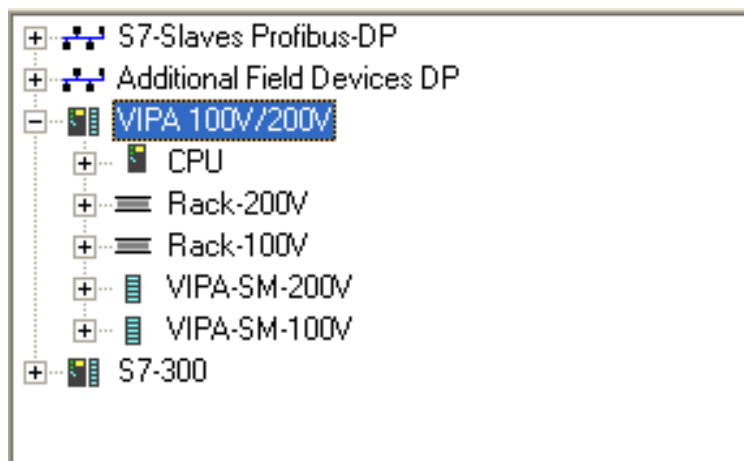


Рисунок 1.5 – Швидка конфігурація модулів ПЛК

де легко і зручно знайти назви модулів контролера. Після введення назв модулів необхідно зберегти конфігурацію (рисунок 1.6):

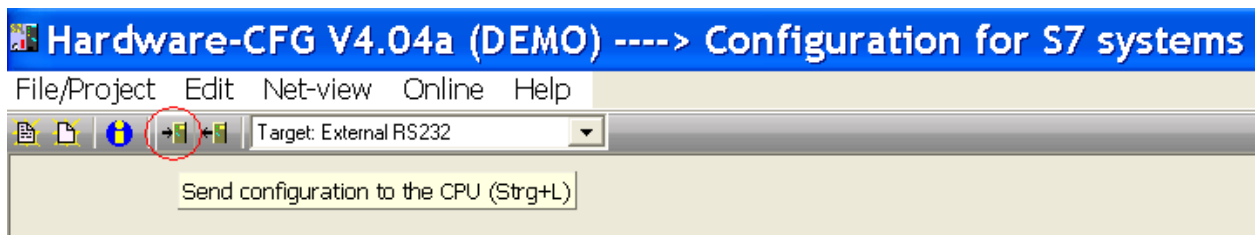


Рисунок 1.6 – Збереження конфігурації у CPU

Після чого з'явиться зв'язок між контролером та комп'ютером. Контролер можна програмувати ☺.

1.3 Завдання

Варіант 1, 2, 3, 4, 5: створити новий проект ім'я якого відповідає номеру ПК з яким встановлюється зв'язок ПЛК (Наприклад МРА_1_Lab_1). З допомогою "Hardware configuration" сконфігурувати модулі для зв'язку між ПЛК та комп'ютером. Використовуючи команду ОС Microsoft Windows Print Screen SysRq, зберегти отриману таблицю конфігурації модулів ПЛК.

1.4 Зміст звіту

1. Мета роботи.
2. Створений новий проект.
3. Вибрана серія ПЛК.
4. Таблиця конфігурації модулів вибраної серії ПЛК.
5. Висновки.

Контрольні запитання

1. Для чого призначений пакет WinPLC7?
2. Які мови програмування включає в себе пакет WinPLC7? Охарактеризуйте особливості кожної з них.
3. ПЛК System 200V: структура, особливості, модулі, інтерфейси, ММС слот, потенціометр, світлодіоди стану.
4. Основні функціональні команди WinPLC7 та позначення релейно-контактних схем, що використовуються у пакеті WinPLC7.
5. Призначення Hardware configuration у пакеті WinPLC7.
6. MPI інтерфейс.
7. Потенціометр.
8. ММС слот.
9. Основні функціональні команди WinPLC7. Менеджер керування підпроектами (PRJ).

Лабораторна робота № 2

ПРОГРАМУВАННЯ ТАЙМЕРІВ

Мета роботи: проектування логічних ланцюгів з використанням програмного пакета WinPLC7, ознайомитися із завантаженням програми в ПЛК, її тестуванням та навчитися програмувати таймери (Timers).

2.1 Основні теоретичні відомості

Таймери дозволяють програмно реалізувати послідовності синхронізації, такі як інтервали очікування та спостереження, вимірювання інтервалів або генерування імпульсів.

Існують такі типи таймерів: імпульсні таймери (Pulse timers); розширені імпульсні таймери (Extended pulse timers); таймери затримки ввімкнення (On-delay timers); таймери затримки ввімкнення із запам'ятовуванням (Retentive on-delay timers); таймери затримки вимкнення (Off-delay timers).

Ви можете запрограмувати таймер повністю як блоковий елемент або за допомогою окремих програмних елементів. При запуску таймера ви визначаєте необхідний тип таймера, і як довго він повинен працювати; також можна скинути таймер. Таймер контролюється шляхом запиту його стану ("Timer running" → "Таймер працює") або поточного значення таймера, яке можна отримати із таймера у двійковому або двійково-десятьковому (BCD) коді.

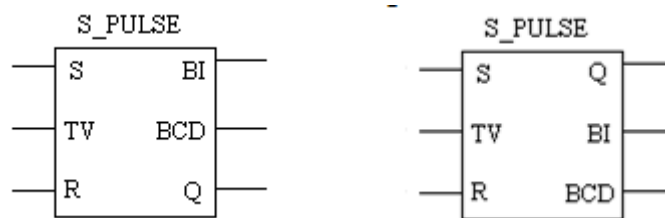


Рисунок 2.1 – Таймер у вигляді блокового елемента (timer box)

Таблиця 2.1 – Основні програмні елементи таймерів

Найменування	Тип даних	Опис
S	BOOL	Вхід запуску
TV	S5TIME	Специфікація тривалості часу
R	BOOL	Вхід скидання
BI	WORD	Поточне значення часу в двійковому поданні
Q	BOOL	Стан таймера

2.1.1 Програмування таймера

Можна виконати такі операції з таймером: запустити таймер, задавши значення часу; обнулити (скинути) таймер; перевірити (двійковий) стан таймера; прочитати (цифрове) значення таймера в бінарному вигляді; прочитати (цифрове) значення таймера в BCD вигляді.

Блоковий елемент для таймера містить всі ці операції таймера в формі функціональних входів та функціональних виходів (рисунок 2.1). Над блочним елементом розташована абсолютна або символічна адреса таймера. В самому блоковому елементі як заголовок вказаний режим таймера (S_PULSE означає "Start pulse timer" → "Запуск імпульсного таймера"). Призначення (адресація) для входів S та TV обов'язкові, у той час як для інших – ні.

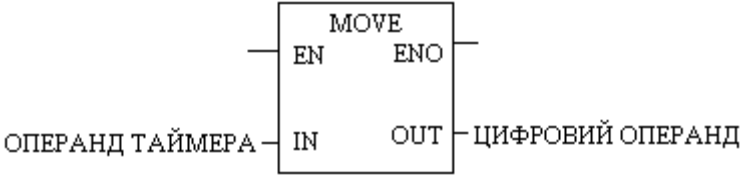
2.1.2 Окремі програмні елементи в LAD

Також можна запрограмувати таймер з використанням окремих програмних елементів (таблиця 2.3). В цьому випадку запуск таймера здійснюється за допомогою котушки. Режим таймера вказаний в котушці (SP = start pulse timer або запуск імпульсного таймера), а під котушкою розташоване значення в форматі S5TIME, що визначає тривалість (duration). Щоб обнулити таймер, використовуйте котушку скидання. NO- і NC-контакти можна застосувати для перевірки стану таймера. Нарешті, можна зберегти поточне значення часу в двійковій формі в операнді, розміром у слово за допомогою блокового елемента MOVE.



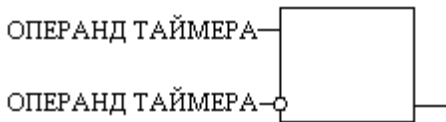
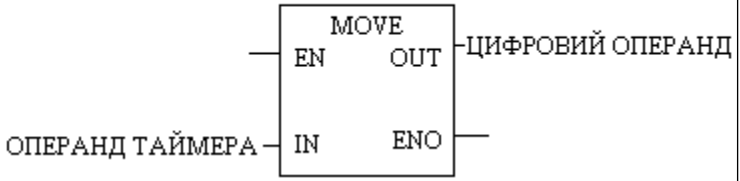
2.1.3 Окремі програмні елементи в FBD

Ви також можете запрограмувати таймер за допомогою окремих програмних елементів (таблиця 2.3). У даному випадку таймер стартує за допомогою простого блокового елемента, що містить режим таймера (SP = start pulse timer або запуск імпульсного таймера). Під елементом знаходиться значення в форматі S5TIME, що визначає тривалість. Щоб обнулити таймер, використовуйте блоковий елемент скидання. Сканувати стан таймера можна безпосередньо або в інвертованому вигляді з використанням будь-якого бінарного входу. Ви можете зберегти поточне значення часу в двійковій формі в операнді розміром у слово за допомогою блокового елемента MOVE.

Таблиця 2.2 – Окремі елементи таймера (LAD)

<p>Запуск таймера з визначеним значенням часу (котушка запуску з робочими характеристиками)</p>	<p>Операнд таймера ------(SP)----- Тривалість</p>
<p>Скидання таймера (котушка скидання)</p>	<p>Операнд таймера ------(R)----- </p>
<p>Перевірка стану таймера (NO-контакт і NC - контакт)</p>	<p>Операнд таймера Операнд таймера ----- ----- ----- / -----</p>
<p>Читання значення часу в двійковій формі (блоковий елемент MOVE)</p>	

Таблиця 2.3 – Окремі елементи таймера (FBD)

<p>Запуск таймера з визначеним значенням часу (блоковий елемент запуску з робочими характеристиками)</p>	
<p>Скидання таймера (блоковий елемент скидання)</p>	<p>ОПЕРАНД ТАЙМЕРА</p> 
<p>Перевірка стану таймера (прямий або інвертований двійковий вхід)</p>	
<p>Читання значення часу в двійковій формі (блоковий елемент MOVE)</p>	

Для створення програм за допомогою покрокового програмування елементи-таймери необхідно знайти в каталозі програмних елементів (Program Element Catalog), що викликається командою меню View → View → Catalog (Вигляд → Каталог) [Ctrl + K].

2.1.4 Запуск таймера

Таймер переходить в режим “START”, коли результат логічної операції (RLO) змінюється на вході запуску (start input) або перед котушкою \ блоковим елементом запуску. Така зміна сигналу завжди необхідна, щоб запустити таймер. У випадку програмування таймера на затримку вимкнення RLO повинен змінитися з ”1” на ”0”; всі інші таймери стартують при зміні RLO з ”0” на ”1”. Можна працювати в одному з п’яти різних режимів. Але немає сенсу використовувати будь-який даний таймер більше, ніж в одному режимі.

2.1.5 Визначення тривалості часу

Таймер як тривалість бере значення, що вказане під котушкою \ блоковим елементом, або значення на вході TV. Ви можете задати тривалість як константу, як операнд розміром в слово або як змінну типу S5TIME.

Таблиця 2.4 – Визначення тривалості для константи

S5TIME#10s	Тривалість 10 с
S5T#1m10ms	Тривалість 1м + 10 мс

Тривалість задається в годинах, хвилинах, секундах і мілісекундах. Область значень простягається від S5TIME#10s до S5TIME#2h46min30s (що відповідає 9990 с).

Проміжні значення округляються до 10 мс. Для визначення константи ви можете використовувати S5TIME# або S5T#.

Таблиця 2.5 – Визначення тривалості як операнда або змінної

MW 20	Операнд розміром у слово, що містить тривалість
”Time 1”	Змінна типу S5TIME

Значення в 16-бітовому операнді повинно відповідати типу даних S5TIME.

2.1.6 Скидання таймера

LAD: таймер скидається, коли електричний струм протікає на вході скидання (reset input) або в котушці скидання (reset coil) (коли "RLO" дорівнює "1"). Поки таймер залишається скинутим, сканування за допомогою NO-контакту поверне "0", а за допомогою NC-контакту "1".

FBD: таймер онуляється, коли на вході скидання присутня "1". Поки таймер залишається скинутим, пряме сканування стану таймера поверне "0", а інвертоване сканування поверне "1".

Скидання таймера встановляє даний таймер та тимчасову базу в нуль. Вхід R блокового елемента таймера повинен бути призначений (тобто, на цьому вході повинно бути присутнє фактичне значення).

2.2 Порядок виконання роботи

Щоб розпочати роботу з програмою **WinPLC7** програмного пункту **VIPA** необхідно натиснути кнопку **Пуск**, потім вибрати пункт **Програми** → **VipaGmbH** → **WinPLC7** → **WinPLC7 starten**.

Після відкриття **WinPLC7** натискаємо **Continue**. Після завантаження програми необхідно створити новий проект. Це можна зробити таким чином **File** → **Open\Create a project (Ctrl+O)**. У вікні, яке з'явилося, натискаємо **New project** і називаємо його, наприклад, **MPA_1_Lab_2**. Далі виділяємо наш проект та натискаємо кнопку **Open** (рисунок 2.1).

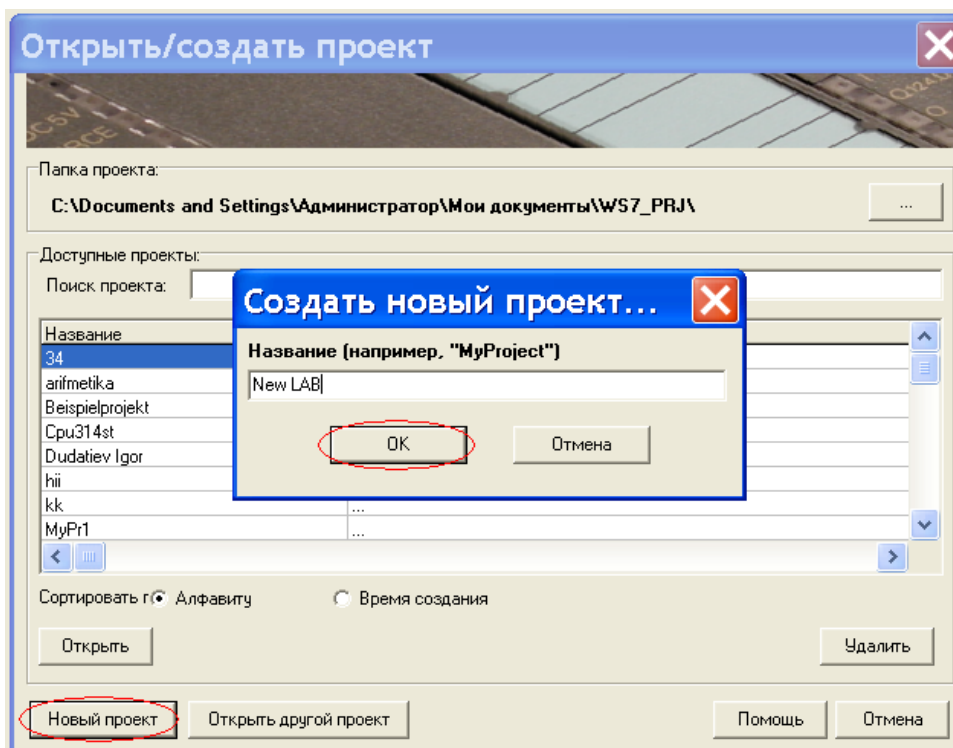


Рисунок 2.1 – Створення нового проекту у WinPLC7

Для подальшої роботи необхідно створити організаційний блок, який назвемо **OB1**. Виберемо в головній панелі **File** → **Create New Block (Ctrl+N)**, введіть назву блока **OB1** та натисніть **OK**. Другий варіант створення організаційного блока: натисніть на іконку **OB1** та зробіть підтвердження створення блока.

Налагодження програми. Програмний пакет WinPLC7 має засоби, які дозволяють перевірити правильність роботи програми або її частини. Перевірка проводиться у режимі налагодження. Для налагодження необхідно вибрати режим роботи – симуляція (рисунок 2.2):

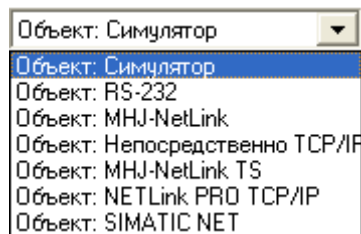


Рисунок 2.2 – Вибір об'єкта у WinPLC7

Після цього можна завантажувати програму в емулятор ПЛК. Для цього виконайте такі дії **PLC** → **Send all block**. В результаті виконання з'явиться таке вікно (рисунок 2.3):

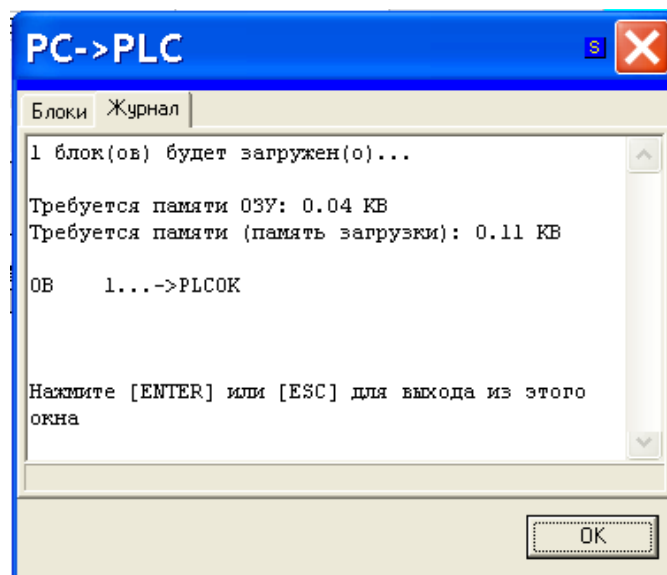


Рисунок 2.3 – Завантаження робочих блоків у ПЛУ в режимі симулятор

Потім необхідно запустити емулятор роботи ПЛК. Для цього наведіть курсор на панель **Switch\Operating mode** та натисніть ліву кнопку миші або **PLC** → **Operating mode**.

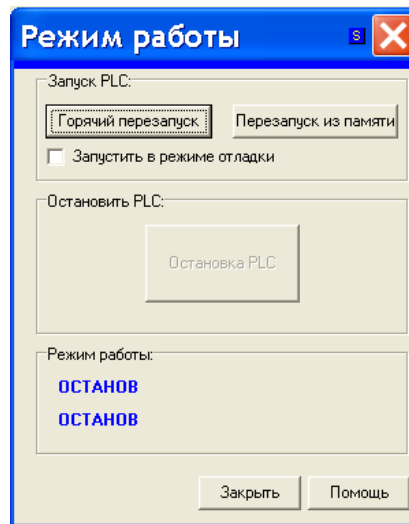


Рисунок 2.4 – Емуляція роботи ПЛК

Для запуску емуляції роботи ПЛК натисніть **Hot Restart** (рисунок 2.4). Ви можете завершити діалог, натиснувши на Close. Панель Switch\Operating mode набуде такого вигляду (рисунок 2.5):

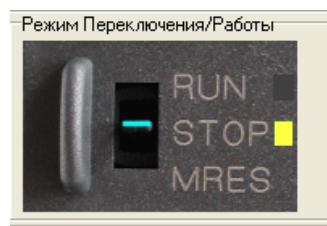


Рисунок 2.5 – Робоча панель режимів роботи

При налагодженні програми вам необхідно буде відстежувати стани входів та виходів ПЛК, лічильників та ін. Зручно з цією метою використовувати process-images-window (рисунок 2.6). Це вікно можна викликати так: **View->Display process-images-window**.

PIB0 76543210	PIB1 76543210	PIB2 76543210	PIB3 76543210	PIB4 76543210
PQB0 76543210	PQB1 76543210	PQB2 76543210	PQB3 76543210	PQB4 76543210

Рисунок 2.6 – Таблиця станів входів\виходів ПЛК

Вікно "PIBO" байт входу IBO, тобто біти I0.0 – I0.7. Ви можете змінювати стани бітів, натискаючи на відповідний номер мишкою. Наприклад, якщо Ви натиснете на номер 1 в вікні "PIBO" зміниться стан входу I0.1. Якщо стан біту дорівнює 1, то він виділяється червоним

кольором. Стан виходів таким чином не може бути змінений, оскільки вони контролюються програмою в ПЛК.

Таким способом можна відстежувати стан не лише входів – виходів ПЛК, а також таймерів, лічильників та інших компонентів розробки. Для того, щоб його додати в process-images-window, необхідно навести курсор на дане вікно, натиснути праву кнопку миші та вибрати **Configure**.

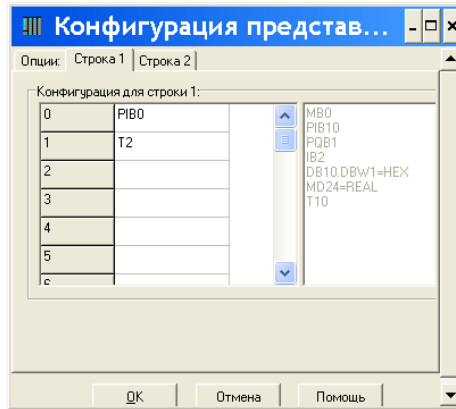


Рисунок 2.7 – Конфігурування ліній

Дане діалогове вікно має такі особливості. Кожна лінія в вікні process-images-window асоціюється з окремою таблицею, тому якщо ви хочете, щоб таймер відображався у верхньому рядку, то вам необхідно перейти в таблицю Line 1, видалити PIB1 – PIB4 (якщо їх не використовувати) і у рядку з номером 1 записати T2 – це таймер, який нами використовується. Натискаємо ОК.

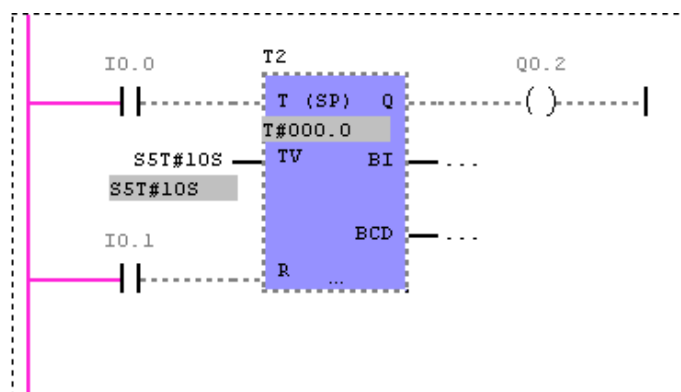


Рисунок 2.8 – Налаштування програми

Для того, щоб розпочати налаштування необхідно виконати таке **Block → Monitoring On\Off**. У графічних мовах програмування з допомогою функцій “Контактний план” та “Функціональний план” можна простежити результати тестування, переглядаючи зміни кольору в сегментах програми. Ця зміна кольору показує, що результат логічної операції виконаний для цієї точки.

На рисунку 2.8 струмова шина відображається суцільною лінією аж до I0.0 і I0.1 це означає, що у коло вже поступає живлення. Якщо ми замкнемо контакт I0.0, то через 10 секунд встановиться в "1" вихід Q0.2. Process-images-window буде виглядати таким чином.

PIB0 7 6 5 4 3 2 1 0		T2 S5T#00H 00M 08S 500MS	
QV0 7 6 5 4 3 2 1 0	QV1 7 6 5 4 3 2 1 0	QV2 7 6 5 4 3 2 1 0	

Рисунок 2.9 – Таблиця станів входів\виходів ПЛК

2.3 Завдання

Варіант 1. Після активації входу I0.3, через 5 секунд активізується вихід Q0.1.

Варіант 2. Після активації входу I0.1, активізується вихід Q0.2. Після відключення входу I0.1, через 3 секунди вихід Q0.2 відключається.

Варіант 3. Після активації входу I0.4, через 5,5 секунд активізується вихід Q0.1. Після відключення входу I0.4, вихід Q0.1 відключається.

Варіант 4. Після активації входу I0.1 активізується вихід Q0.2, а через 10 секунд відключається.

Варіант 5. Після активації входу I0.4, через 7 секунд активізується вихід Q0.1, а потім через 12,8 секунд відключається.

2.4 Зміст звіту

1. Мета роботи.
2. Налаштування програми в режимі симулятор.
3. Лістинг програми мовою LAD.
4. Приклад роботи програми.
5. Таблиця станів входів\виходів ПЛК
6. Висновки.

Контрольні запитання

1. Таймери. Їх призначення.
2. Які існують типи таймерів? Охарактеризуйте кожен з них.
3. Подання імпульсного таймера за допомогою LAD. Позначення входів та виходів (input and output) у імпульсному таймері.
4. Основні операції, що можна реалізувати за допомогою таймера.
5. Керування таймером у режимі симуляції.
6. Основні режими роботи ПЛК.

Лабораторна робота № 3

ПРОЕКТУВАННЯ ЛОГІЧНИХ ЛАНЦЮГІВ ЛІЧИЛЬНИКИ. КОМПАРАТОРИ

Мета роботи: проектування логічних ланцюгів з використанням програмного пакета **WinPLC7**, навчитися програмувати лічильники (**Counters**) та компаратори (**Comparators**).

3.1 Основні теоретичні відомості

Лічильники (counters) дозволяють використовувати CPU в обчислювальних задачах. Лічильники можуть вести підрахунок по зростаючій (прямий підрахунок) і по спадній (зворотний підрахунок). Область підрахунку охоплює три розряди (від 000 до 999). Лічильники знаходяться в системній пам'яті CPU; кількість лічильників визначається версією CPU.

Ви можете запрограмувати лічильник повністю у вигляді блокового елемента або з використанням окремих програмних елементів. Можна задати для лічильника конкретне початкове значення або скинути його, вести прямий або зворотний підрахунок. Лічильник сканується шляхом зчитування його стану (нульове або ненульове значення рахунку) або поточного значення лічильника (значення підрахунку, count value), яке можна отримати або в двійковому вигляді, або в двійково-десятковому коді.

3.1.1 Програмування лічильника

З лічильниками реалізуються такі операції:

- встановлення лічильника, задання значення лічильника;
- прямий рахунок;
- зворотний рахунок;
- скидання лічильника;
- зчитування стану (двійкового) лічильника;
- зчитування (числового) значення лічильника в двійковій формі;
- зчитування (числового) значення лічильника в двійково-десятковій формі.

3.1.1 Подання лічильника у вигляді блокового елемента

Блоковий елемент лічильника містить всі операції підрахунку у формі функціональних входів і функціональних виходів (рисунок 3.1). Над блоковим елементом розташована абсолютна або символічна адреса

лічильника. В блоковому елементі як заголовок зазначається тип лічильника (S_CUD означає "up-down counter", "лічильник прямого і зворотного підрахунку"). Призначення для першого входу (на прикладі CU) обов'язково; для інших входів і виходів призначення можна не робити.

Маємо три виконання блокових елементів лічильника: лічильник прямого і зворотного підрахунків (up-down counter, при програмуванні мовою FBD називається S_CUD, а мовою LAD - ZAEHLER), лічильник лише прямого підрахунку (up counter, мовою FBD називається S_CUD, а мовою LAD – Z_VORW) і лічильник зворотного рахунку (down counter, мовою FBD називається S_CD, а мовою LAD – Z_RUECK).

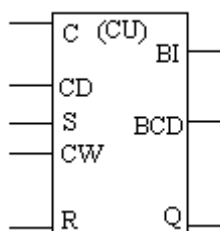


Рисунок 3.1 – Лічильник

Таблиця 3.1 – Входи – виходи блокового елемента лічильника

Найменування	Тип даних	Опис
CU	BOOL	Вхід прямого підрахунку
CD	BOOL	Вхід зворотного підрахунку
S	BOOL	Вхід установа
CW	WORD	Вхід передустановка
R	BOOL	Вхід скидання
BI	WORD	Поточне значення в двійковому вигляді
BCD	WORD	Поточне значення в двійково-десятковому вигляді
Q	BOOL	Стан лічильника

3.1.2 Подання лічильника з використанням елементів (LAD)

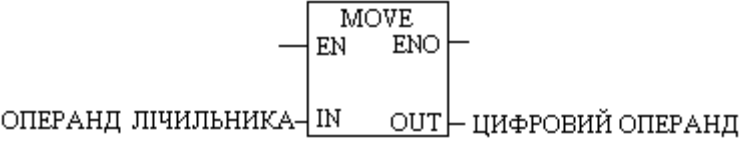
Запрограмувати лічильник можна використовуючи окремі елементи. Встановлення і підрахунок в цьому випадку відбувається за допомогою катушок. Катушка встановлює лічильник, який містить операцію обрахування (підрахунку) (SC = Set Counter, встановлення лічильника); під катушкою знаходиться значення лічильника в форматі WORD, яке використовується для встановлення лічильника.

В катушках для обрахування CU (count up) означає прямий підрахунок, CD (count down) – зворотний підрахунок. Щоб скинути

лічильник, використовуйте котушку для скидання, а NO- або NC-контакт – для перевірки стану лічильника.

Нарешті, ви можете переслати поточне значення лічильника в двійковому вигляді за допомогою блокового елемента MOVE.

Таблиця 3.2 – Окремі елементи лічильника (LAD)

Лічильник прямого підрахунку (котушка прямого підрахунку)	Операнд лічильника ------(CU)-----
Скидання зворотного підрахунку (котушка зворотного підрахунку)	Операнд лічильника ------(CD)-----
Установлення лічильника, визначення значення лічильника (котушка установлення лічильника зі значенням підрахунку)	Операнд лічильника ------(SC)----- Значення підрахунку
Скидання лічильника (котушка скидання)	Операнд лічильника ------(R)-----
Перевірка стану лічильника NO-контакт, NC-контакт	Операнд лічильника Операнд лічильника ----- ----- ----- / -----
Читання значення рахунку в двійковій формі (блоковий елемент MOVE)	

3.1.3 Подання лічильника з використанням елементів (FBD)

Запрограмувати лічильник можна, використовуючи окремі елементи. Встановлення та підрахунок в цьому випадку відбувається за допомогою простих блочних елементів. Блоковий елемент установлення лічильника містить операцію підрахунку (SC = Set Counter, встановлення лічильника); на вході CW знаходиться значення лічильника в форматі WORD, яке використовується для встановлення лічильника.

В блокових елементах для підрахунків CU (count up) – прямий підрахунок, CD (count down) – зворотний підрахунок. Щоб скинути лічильник, використовуйте блоковий елемент для скидання, а прямий або інвертований двійковий вхід функції – для перевірки стану лічильника.

Нарешті ви можете переслати поточне значення лічильника в двійковому вигляді за допомогою блокового елемента MOVE.

3.1.4 Встановлення і скидання лічильників

Встановлення лічильників. Лічильник встановлюється, коли RLO змінюється з "0" на "1" на вході встановлення S або перед котушкою встановлення, або блоковим елементом встановлення. Для встановлення лічильника завжди потрібен додатний фронт. "Встановлення лічильника" ("Set counter") означає, що лічильник встановлюється в початкове значення. Область значення – від 0 до 999.

Таблиця 3.3 – Окремі елементи лічильника (FBD)

Лічильник прямого підрахунку (блоковий елемент прямого підрахунку)	ОПЕРАНД ЛІЧИЛЬНИКА 
Лічильник зворотного підрахунку (блоковий елемент зворотного підрахунку)	ОПЕРАНД ЛІЧИЛЬНИКА 
Встановлення лічильника, визначення значення лічильника (блоковий елемент встановлення лічильника зі значенням підрахунку)	ОПЕРАНД ЛІЧИЛЬНИКА  ЗНАЧЕННЯ РАХУНКУ
Скидання лічильника (блоковий елемент скидання)	ОПЕРАНД ЛІЧИЛЬНИКА 
Перевірка стану лічильника (прямий або інвертований двійковий вхід)	ОПЕРАНД ЛІЧИЛЬНИКА 
Читання значення підрахунку в двійковій формі (блоковий елемент MOVE)	 ЦИФРОВИЙ ОПЕРАНД ОПЕРАНД ЛІЧИЛЬНИКА

3.1.5 Визначення значення підрахунку лічильника

Коли лічильник автоматично встановлюється, він бере як значення рахунку значення на вході CW (значення під котушкою установки). Ви можете визначити значення лічильника як константи, операнда довжиною в слово або змінною типу WORD.

Таблиця 3.4 – Визначення значення рахунку як константи

C#100	Значення підрахунку 100
W#16#0100	Значення підрахунку 100

Значення рахунку містить три розряди з області 000...999. Припустимі лише додатні BCD-значення; лічильник не обробляє від'ємні значення. Для ідентифікації константи ви можете використовувати C# або W#16# (лише разом з десятковими числами).

3.1.6.1 Скидання лічильників (LAD)

Лічильник скидається, коли струм протікає на вході скидання або в котушці скидання (коли присутнє RLO "1"). В цьому випадку перевірка лічильника NO-контактом повертає результат зчитування "1". Скидання лічильника встановлює його значення рахунку в "нуль". Вхід R блокового елемента лічильника може бути вільним (непід'єднаним).

3.1.6.2 Скидання лічильників (FBD)

Лічильник скидається, коли "1" присутня на вході скидання. Пряме сканування стану лічильника поверне "0", а інвертоване сканування поверне "1". Скидання лічильника встановлює його значення рахунку в "нуль". Вхід R блокового елемента лічильника може бути вільним (непід'єднаним).

3.1.7 Частота підрахунку лічильника

Частота підрахунку лічильника визначається часом виконання вашої програми. Для того, щоб вести підрахунок, CPU повинен виявити зміни в стані вхідного імпульсу, тобто вхідний імпульс (або його відсутність) повинен бути присутнім протягом, як мінімум, одного програмного циклу. Таким чином, чим довший час виконання програми, тим менша частота ведення підрахунку.

Підрахунок по зростаючій. Лічильник приступає до прямого підрахунку, коли RLO змінюється з "0" → "1" на вході прямого підрахунку CU (up count input) або перед котушкою прямого підрахунку,

або блоковим елементом. Для зворотного підрахунку завжди необхідний додатний фронт.

При прямому підрахунку кожний додатний фронт збільшує рахунок на одиницю до досягнення верхньої межі інтервалу 999. Кожен наступний додатний фронт впливу на ведення прямого підрахунку не дає. Перенос не відбувається.

Підрахунок по спадній. Лічильник починає вести підрахунок по спадній при зміні RLO з "0" → "1" на вході зворотного підрахунку CD (down count input) або перед котушкою зворотного підрахунку, або блоковим елементом. Для зворотного підрахунку завжди необхідний додатний фронт.

При зворотному підрахунку кожний додатний фронт зменшує рахунок на одиницю до досягнення нижньої межі інтервалу 0. Кожний наступний додатний фронт для ведення зворотного підрахунку не використовується. Від'ємних значень підрахунку не виникає.

3.1.8 Опитування лічильника

Перевірка стану лічильника (LAD). Стан лічильника подається на вихід Q блокового елемента лічильника. Стан лічильника також можна перевірити з використанням NO-контакту (відповідає виходу Q) або NC-контакту. Вихід Q містить "1" (струм тече з виходу), коли поточне значення рахунку дорівнює нулю. Вихід Q в блоковому елементі лічильника може бути не під'єднаний.

Перевірка стану лічильника (FBD). Стан лічильника подається на вихід Q блокового елемента лічильника. Стан лічильника також можна опитати безпосередньо (відповідає виходу Q) за допомогою двійкової функції або в інвертованому вигляді. Вихід Q містить "1", коли поточне значення рахунку більше нуля. На виході Q міститься "0", коли поточне значення рахунку дорівнює нулю. Вихід 0 блокового елемента лічильника може бути не під'єднаний.

3.2 Завдання

Варіант 1. Лічильник запускається при активації входу I1, до 4-х і активізує котушку Q2. Лічильник обнуляється кожного разу при активації входу I2.

Варіант 2. Лічильник запускається при активації входу I1, до 3-х і відмикає котушку Q1. Лічильник обнуляється кожного разу при активації входу I2.

Варіант 3. Лічильник запускається при активації входу I1, від 5 до 0 і відключає котушку Q1. Лічильник обнуляється кожного разу при активації входу I2.

Варіант 4. Лічильник запускається при активації входу I3, від 4 до 0 і активізує котушку Q2. Лічильник обнуляється кожного разу при активації входу I2.

Варіант 5. Лічильник запускається при активації входу I2, до 5 і активізує котушку Q1. Лічильник обнуляється кожного разу при активації входу I1.

3.3 Функції порівняння (Comparator). Основні теоретичні відомості

Функції порівняння порівнюють дві числові змінні, що належать до типів даних INT, DINT, REAL, на предмет рівності, нерівності, більше, більше чи дорівнює, менше чи дорівнює. Після операції порівняння видається її результат у вигляді двійкового значення (таблиця 3.5).

Таблиця 3.5 – Огляд функцій порівняння

Функції порівняння	Порівняння відповідно до типу даних		
	INT	DINT	REAL
Порівняння "дорівнює"	CMP==I	CMP==D	CMP==R
Порівняння "не дорівнює"	CMP<>I	CMP<>D	CMP<>R
Порівняння "більше"	CMP>I	CMP>D	CMP>R
Порівняння "більше чи дорівнює"	CMP>=I	CMP>=D	CMP>=R
Порівняння "менше"	CMP<I	CMP<D	CMP<R
Порівняння "менше чи дорівнює"	CMP<=I	CMP<=D	CMP<=R

3.3.1 Обробка функцій порівняння

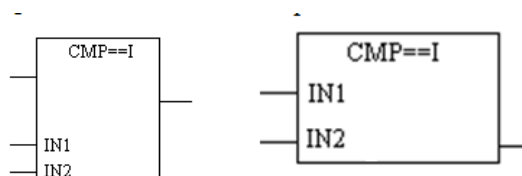


Рисунок 3.2 – Блоковий елемент порівняння (операція порівняння "дорівнює" між числами INT)

Подання (LAD). Окрім (немаркованого) двійкового входу блоковий елемент функції порівняння має два входи IN1 і IN2 і (немаркований) двійковий вихід. "Заголовок" в блоковому елементі ідентифікує операцію порівняння (CMP означає compare, порівняння) і тип виконуваного порівняння (наприклад, під CMP ==1 маємо на увазі порівняння двох чисел типу INT на предмет рівності).

У мережі можна компаратор (блок порівняння) поставити замість контакту. Немарковані вхід і вихід слугують для з'єднання інших (двійкових) програмних елементів.

Значення, що порівнюються, подаються на входи IN1 та IN2, результат порівняння – на виході. Успішне порівняння еквівалентне замкненому контакту ("струм" протікає через компаратор). Якщо порівняння не успішне, то контакт розімкнено. Вихід компаратора завжди повинен бути підключений.

Подання (FBD). Блоковий елемент порівняння має два входи IN1 та IN2 і немаркований двійковий вихідом. "Заголовок" в блоковому елементі ідентифікує операцію виконуваного порівняння (наприклад, під CMP==1 маємо на увазі порівняння двох чисел типу INT на предмет рівності).

Значення, що порівнюються, подаються на входи IN1 та IN2, результат порівняння – на виході. Якщо порівняння позитивне, то вихід компаратора показує сигнальний стан "1", у протилежному випадку на виході "0". Він завжди повинен бути підключений.

3.3.2 Типи даних

Тип даних входів в функції порівняння залежить від цієї функції. Наприклад, входи типу REAL в функції порівняння CMP>R (порівняння "більше" чисел REAL). Змінні повинні відповідати типу даних на вході компаратора. При використанні операндів з абсолютними адресами розміри операндів повинні відповідати типам даних. Наприклад, можна використовувати операнд розміром в слово для типу даних INT.

Порівняння між числами REAL не істинне, якщо один або обидва числа REAL недійсні (не є числами типу REAL). Також встановлюються біти стану OS та OV.

Приклад. Порівняння відповідно до REAL. Якщо змінна #Act_value більша чи дорівнює змінній #Calibrat, то #NewCali встановлюється; інакше #NewCali залишається без змін.

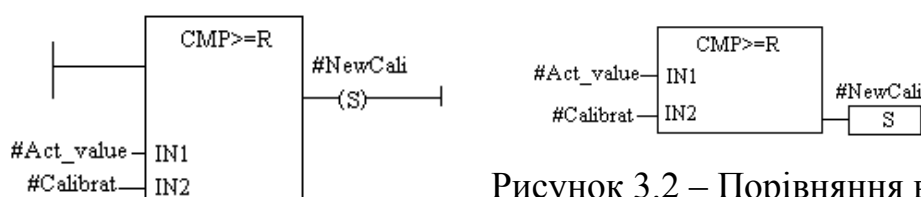


Рисунок 3.2 – Порівняння відповідно до REAL

3.4 Завдання

Варіант 1. Якщо змінна Val_1 \geq Val_2 (тип змінних **REAL**) і змінна Val_3 \geq Val_4 (тип змінних **REAL**), то активується вихід Q 0.3.

Варіант 2. Якщо змінна Val_1 \leq Val_2 (тип змінних **REAL**) і змінна Val_3 \geq Val_4 (тип змінних **INT**), то активується вихід Q 1.2.

Варіант 3. Якщо змінна Val_1 $<$ Val_2 (тип змінних **INT**) або змінна Val_3 = Val_4 (тип змінних **INT**), то активується вихід Q 0.2.

Варіант 4. Якщо змінна Val_1 \leq Val_2 (тип змінних **REAL**) і змінна Val_3 = Val_4 (тип змінних **INT**), то активується вихід Q 0.5.

Варіант 5. Якщо змінна Val_1 = Val_2 (тип змінних **REAL**) і змінна Val_3 \leq Val_4 (тип змінних **REAL**), то активується вихід Q 1.3.

3.5 Зміст звіту

1. Мета роботи.
2. Лістинг програми реалізації лічильника мовою LAD (відповідно до варіанта завдання).
3. Приклад роботи програми.
4. Лістинг програми реалізації компаратора мовою LAD (відповідно до варіанта завдання).
5. Приклад роботи програми.
6. Висновки.

Контрольні запитання

1. Лічильники. Їх роль.
2. Основні операції, що реалізуються за допомогою лічильника.
3. Подання лічильника з використанням елементів LAD.
4. Компаратори. Їх роль.
5. Подання компаратора за допомогою LAD.
6. Які типи даних можна порівнювати з допомогою компаратора? Яка різниця між ними?
7. Основні функції порівняння, що можна реалізувати з допомогою компаратора.

Лабораторна робота № 4

АРИФМЕТИЧНІ ТА МАТЕМАТИЧНІ ФУНКЦІЇ ПРОГРАМНОГО ПАКЕТА WinPLC7

Мета роботи: набути навичок проектування логічних ланцюгів з використанням арифметичних та математичних функцій програмного пакета WinPLC7.

4.1 Основні теоретичні відомості

Арифметичні функції. Арифметичні функції комбінують два значення відповідно до основних арифметичних операцій додавання, віднімання, множення та ділення. Ви можете застосовувати арифметичні функції до змінних типів INT, DINT та REAL (Таблиця 4.1).

Таблиця 4.1 – Огляд арифметичних функцій

Арифметична функція	З типом даних		
	INT	DINT	REAL
Додавання	ADD_I	ADD_DI	ADD_R
Віднімання	SUB_I	SUB_DI	SUB_R
Множення	MUL_I	MUL_DI	MUL_R
Ділення із часткою як результат	DIV_I	DIV_DI	DIV_R
Ділення із залишком як результат	-	MOD_DI	-

4.1.1 Обробка арифметичної функції. Арифметичний блоковий елемент (додавання відповідно до INT)

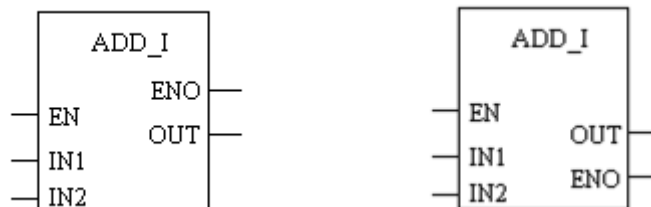


Рисунок 4.1 – Додавання відповідно до REAL

Окрім входу дозволу (enable input) EN і виходу дозволу (enable output) ENO блокувий елемент арифметичної функції має два входи IN1 та IN2 і вихід OUT. "Заголовок" в блокувому елементі ідентифікує арифметичну дію, яка виконується (ADD_1, наприклад, означає додавання чисел типу INT).

Значення, що комбінуються, подаються на входи IN1 та IN2, результат розрахунку знаходиться на виході OUT. Входи та вихід можуть мати різні типи даних в залежності від арифметичної функції. Наприклад, у випадку арифметичної функції ADD_R (додавання чисел типу REAL) входи та вихід віднесені до типу REAL. Змінні, які використовуються, повинні, бути того ж типу даних, що і входи чи виходи. Якщо використовувати для операндів абсолютні адреси, то розміри операндів повинні відповідати типам даних. Так, можна застосувати операнд розміром у слово для типу даних INT.

4.1.2 Функція

Арифметична функція виконується, якщо на вході дозволу присутня "1" (у вході EN протікає струм). Якщо під час розрахунку виникає помилка, то вихід дозволу встановлюється в "0"; у протилежному випадку він встановлюється в "1". Якщо виконання функції не дозволене (EN = "0"), то розрахунок не відбувається і ENO також обнуляється.

Якщо головне реле керування (MCR) активовано, то вихід OUT встановлюється в нуль, коли арифметична функція у процесі обробки (EN = "1"). MCR не впливає на вихід ENO.

Таблиця 4.2 – Стани головного реле керування

Якщо EN == "1" або не використовується ТО		ІНАКШЕ
OUT := IN1 Cfcf IN2		ENO := "0"
Якщо виникла помилка ТО	ІНАКШЕ	
ENO := "0"	ENO := "1"	

Під час виконання арифметичної функції можуть виникнути такі помилки: вихід за межі діапазону (переповнення) в розрахунках з типами INT і DINT; зникання значущих розрядів та переповнення в розрахунках з

типом REAL; недійсне (недопустиме) число REAL в розрахунках з типом REAL.

4.2 Приклади

Нижче наведені приклади для кожного типу даних. Арифметична функція виконує розрахунок відповідно до певних параметрів, навіть якщо не були об'явлені типи даних при використанні операндів з абсолютними адресами.

Розрахунок відповідно до INT. Значення слова пам'яті (маркерів) MW 100 ділиться на 250; цілий чисельний результат зберігається в слові пам'яті MW 102.

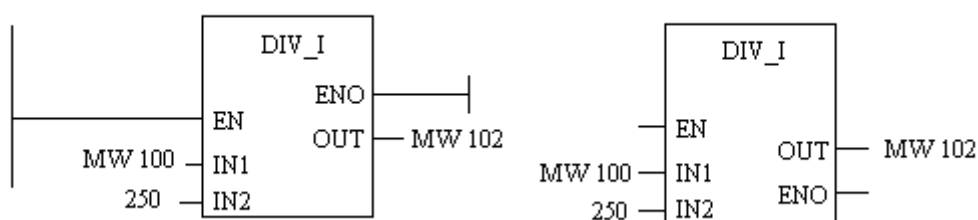


Рисунок 4.2 – Розрахунок відповідно до INT

Розрахунок відповідно до DINT. Значення змінних "CalcVal1" і "CalcVal2" додаються, результат зберігається в змінній "CalcRes". Всі змінні знаходяться у блоці даних "Global_DB".

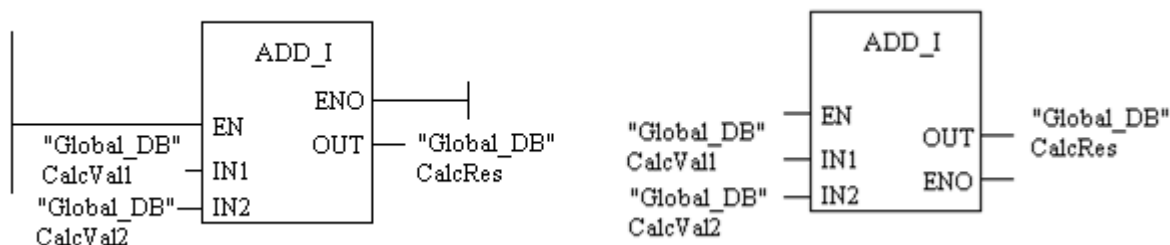


Рисунок 4.3 – Розрахунок відповідно до DINT

Розрахунок відповідно до REAL. Змінна #Act_value множить на змінну #Factor, результат пересилається в змінну #Indicator.

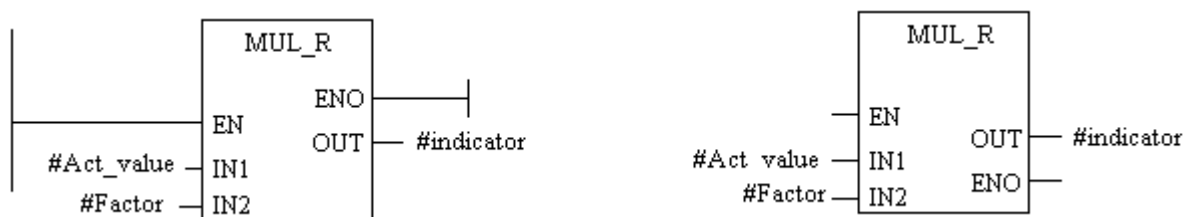


Рисунок 4.4 – Розрахунок відповідно до REAL

Арифметичні функції в логічному ланцюгу (LAD). Ви можете поставити контакти, з'єднавши їх послідовно чи паралельно, перед входом EN і після виходу ENO.

Сам арифметичний блоковий елемент може бути розміщений після T-гілки і у гілці, що веде безпосередньо до лівої живильної напрямної (шини живлення). Ця гілка також може мати контакти перед входом EN та не може бути верхньою гілкою.

Пряме з'єднання з лівою живильною напрямною означає, що ви можете підключити арифметичні блокові елементи паралельно. Коли блокові елементи підключаються паралельно, потрібна котушка для завершення ланцюга. Якщо не робити перевірку на наявність помилки, призначте котушці "пустий" операнд, наприклад, біт часових локальних даних.

Можна з'єднати арифметичні блокові елементи послідовно. Якщо вихід ENO попереднього блокового елемента з'єднаний з входом EN наступного, то останній спрацьовує лише тоді, якщо попередній елемент був оброблений без помилок. Якщо використовувати результат з попереднього блокового елемента як вхідне значення для наступного блока, то змінні з області часових локальних даних надають зручні проміжні буфери.

Якщо компоувати декілька арифметичних блокових елементів в одному ланцюгу (паралельно лівій живильній напрямній і в подальшому послідовно), блокові елементи верхньої гілки треба обробляти зліва направо, потім елементи другої гілки зліва направо і так далі.

Арифметичні функції в логічному ланцюгу (FBD). EN і ENO можуть бути не призначені. Якщо потрібно обробити арифметичний блоковий елемент згідно з певними умовами, то можна скомпонувати логічні операції перед входом EN. Можна з'єднати вихід ENO з бінарними входами інших функцій. Наприклад, можна побудувати арифметичні блокові елементи послідовно, таким чином, вихід ENO попереднього блокового елемента буде підключений до входу EN наступного. Якщо використовувати результат розрахунків з попереднього блокового елемента як вхідне значення для наступного елемента, то змінні з області часових локальних даних дають зручні проміжні буфери.

4.3 Математичні функції

В LAD і FBD подані такі математичні функції:

- синус, косинус, тангенс;
- арксинус, арккосинус, арктангенс;
- піднесення до квадрату, добування квадратного кореня;
- експоненційна функція за основою e ;
- натуральний логарифм.

Всі математичні функції працюють з числами типу даних REAL.

4.3.1 Обробка математичної функції

Блоковий елемент математичної функції має вхід IN та вихід OUT, а також вхід дозволу EN та вихід дозволу ENO. "Заголовок" в блоковому елементі ідентифікує математичну функцію, що виконується (наприклад, SIN означає синус). **Математичний блоковий елемент** (на прикладі: sin)

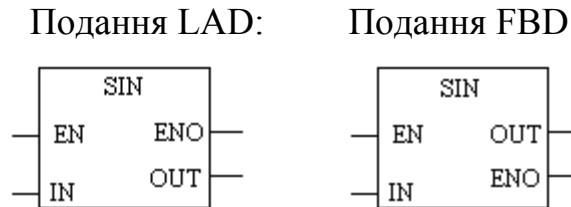


Рисунок 4.5 – Математичний блоковий елемент (sin)

Вхідне значення подається на вхід IN, і результат математичної функції виводиться на вихід OUT. Вхід та вихід відносяться до типу даних REAL. Операнди, що адресуються з використанням абсолютних адрес, повинні мати розмір подвійного слова.

4.3.2 Функція

Математична функція виконується, якщо на дозволяючому вході (EN) присутня "1", або якщо через вхід EN проходить "електричний струм". Якщо при розрахунку виникає помилка, то вихід дозволу встановлюється в "0"; інакше він встановлюється в "1". Якщо виконання функції не дозволено (EN = "0"), то розрахунок не відбувається, і ENO також стає рівним "0".

Таблиця 4.3 – Стани головного реле керування

Якщо EN == "1" або не використовується TO		ІНАКШЕ
OUT := Mfct (IN)		ENO := "0"
Якщо виникла помилка TO	ІНАКШЕ	
ENO := "0"	ENO := "1"	

Якщо головне реле керування (MCR) є активним, то коли математична функція у процесі обробки (EN = '1'), вихід OUT встановлений в нуль. MCR не впливає на ENO.

В математичній функції можуть виникати такі помилки: вихід за межі діапазону (зникнення значущих розрядів і переповнення); недійсне (недопустиме) число типу REAL як вихідне значення.

4.4 Тригонометричні функції

SIN (Синус), COS (Косинус), TAN (Тангенс), приймають на вході значення кута в радіанах як число типу REAL.

Для розміру кута за згодою використовуються дві величини: градуси від 0 градусів до 360 градусів та радіани від 0 до 2π (де $\pi = +3.141593e = 00$). Вони обидва можуть бути пропорційно конвертовані. Наприклад, радіанна міра для кута 90 градусів (прямого) дорівнює $\frac{\pi}{2}$ або $+1.570796e+00$. У випадку значень, більших 2π ($+6.283185e+00$), 2π або декілька 2π віднімається від нього, поки вхідне значення тригонометричної функції не стане менше 2π .

Зворотні тригонометричні функції. ASIN (Арксинус), ACOS (Арккосинус), ATAN (Арктангенс). Аркфункції є зворотними функціями відповідних тригонометричних функцій. Вони допускають на вході IN числа типу REAL певного діапазону і повертають кут в радіанній мірі (таблиця 4.4).

Якщо на вході IN присутнє значення не з допустимого діапазону, то аркфункція повертає недійсне число типу REAL, і ENO = '0', а також вона встановлює біти стану CC0, CC1, OV і OS в '1'.

Таблиця 4.4 – Діапазон аркфункцій

Функція	Допустимий діапазон	Значення, що повертається
ASIN (арксинус)	від -1 до +1	від $-\pi/2$ до $+\pi/2$
ACOS (арккосинус)	від -1 до +1	від 0 до π
ATAN (арктангенс)	весь діапазон	від $-\pi/2$ до $+\pi/2$

4.5 Різні математичні функції

Застосовувати також можна такі математичні функції: SQR (розрахунок квадрата числа), SQRT (розрахунок квадратного кореня

числа), EXP (розрахунок експоненти за основою e), LN (знаходження натурального логарифму (логарифму за основою e)).

Розрахунок квадрата. Функція SQR возводить в квадрат значення на вході IN і зберігає результати на виході OUT.

Розрахунок квадратного кореня. Функція SQRT добуває квадратний корінь значення на вході IN і зберігає результат на виході OUT. Якщо значення на вході IN менше нуля, то SQRT встановлює біти стану CC0, CC1, OV і OS в "1" і повертає недійсне число REAL. Якщо значення на вході IN -0 (мінус нуль), повертається в нуль.

Якщо ви зазначили локальну змінну, ви повинні попередити її знаком #, щоб редактор розпізнав її як локальну змінну, якщо змінна є глобальною змінною, вона повинна бути заключена у лапки.

Розрахунок експоненти за основою e . Функція EXP розраховує експоненційне значення за основою e (рівне 2.718282e+00) і значення на вході IN (e^{IN}) і зберігає результат на виході OUT.

Ви можете розраховувати будь-яке експоненційне значення, використовуючи формулу $a^b = e^{b \ln a}$.

Знаходження натурального логарифму. Функція LN знаходить натуральний логарифм за основою e (=2.718282e+00) числа на вході IN і зберігає його на виході OUT. Якщо значення на вході менше чи дорівнює нулю, то LN встановлює біти стану CC0, CC1, OV і OS в "1" і повертає недійсне число REAL.

Натуральний логарифм є зворотною функцією експоненційної функції: якщо $y = e^x$, то $x = \ln y$.

Щоб знайти будь-який логарифм, скористайтеся формулою

$$\log_b a = \frac{\log_n a}{\log_n b}.$$

Тут b чи n – будь-яка основа. Якщо взяти $n = e$, то можна знайти логарифм за будь-якою основою, використовуючи натуральний логарифм

$$\log_b a = \frac{\ln a}{\ln b}.$$

В особливому випадку для основи 10 формула така

$$\log_b a = \frac{\ln a}{\ln 10} = 0.4342945 \ln a.$$

4.6 Завдання

Варіант 1. Засобами WinPLC7 скласти програму для розрахунку неактивної (холостої) потужності $P_s = U I \sin\psi$.

Варіант 2. Засобами WinPLC7 скласти програму для розрахунку гіпотенузи прямокутного трикутника $c = \sqrt{a^2 + b^2}$.

Варіант 3. Засобами WinPLC7 скласти програму для розрахунку енергії тіла, що рухається $E = m v^2 / 2$.

Варіант 4. Засобами WinPLC7 скласти програму для розрахунку об'єму циліндра $V = (\pi D^2 / 4) h$.

Варіант 5. Засобами WinPLC7 скласти програму для розрахунку кількості тепла, що необхідне для нагріву рідини $Q = c m (T_2 - T_1)$.

4.7 Зміст звіту

1. Мета роботи.
2. Лістинг програми реалізації математичної функції мовою LAD (відповідно до варіанта завдання).
3. Приклад роботи програми.
4. Висновки.

Контрольні запитання

1. Основні математичні та арифметичні операції, які можна реалізувати за допомогою програмного пакета WinPLC7.
2. Подання блока додавання у LAD (на прикладі).
3. Подання блока тригонометричних функцій (наприклад \sin) у LAD (на прикладі).
4. Розрахунок квадратного кореня, знаходження експоненти та натурального логарифма засобами VIPA.

Лабораторна робота № 5

ПРОГРАМУВАННЯ ЗА ДОПОМОГОЮ СИМВОЛІВ, СТВОРЕННЯ ПРОГРАМИ З ФУНКЦІОНАЛЬНИМИ БЛОКАМИ І БЛОКАМИ ДАНИХ

Мета роботи: навчитися використовувати символічний редактор, проектувати програми з функціональними блоками і блоками даних.

5.1 Порядок виконання роботи

Завантажте WinPLC7, потім створіть проект з назвою MyLab5 та відкрийте його. Створіть організаційний блок OB1 та коло, що зображене на рисунку:

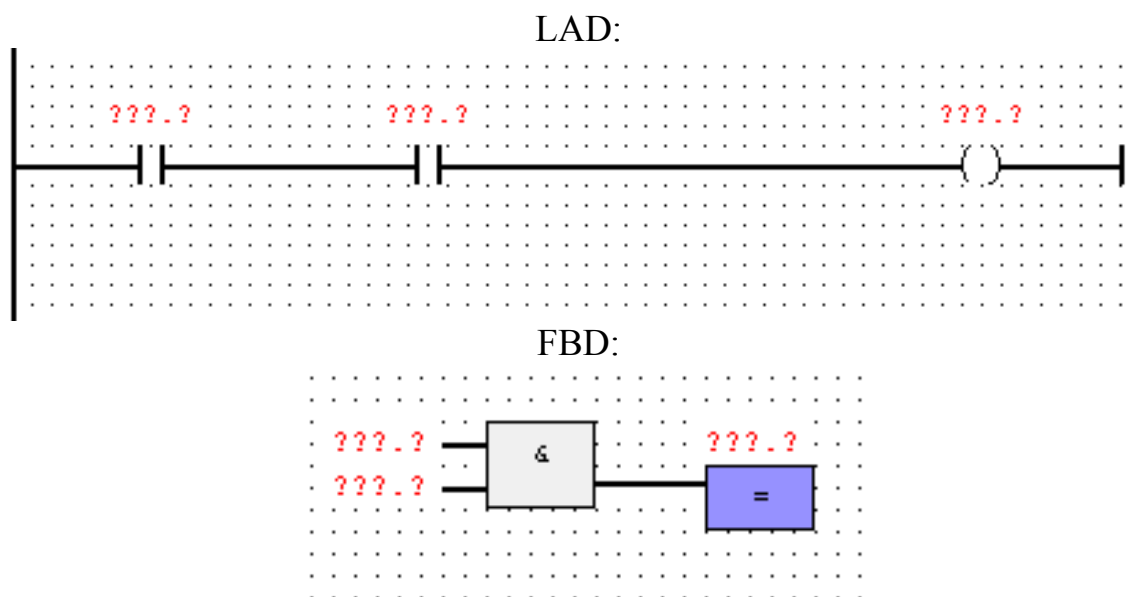


Рисунок 5.1 – Мережа розімкнених контактів та котушки

Абсолютна адресація Кожен вхід та вихід має абсолютну адресу, що визначена заздалегідь конфігурацією апаратури. Ця адреса вказується безпосередньо, тобто абсолютно.

Зазначимо наші входи та виходи за допомогою абсолютної адресації:

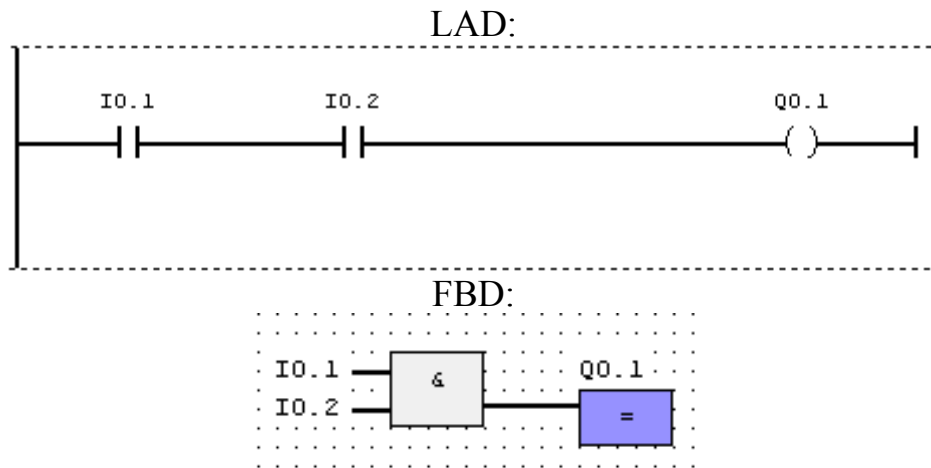


Рисунок 5.2 – Абсолютна адресація входів\виходів

Щоб зазначити входи і виходи за допомогою символів, відкрийте Symbolic editor (редактор символів).

...	Обозначение	Адрес	Тип	Симв.-Комментарий
1				
2				

Рисунок 5.3 – Інтерфейсна панель символного редактора (Symbolic editor)

Введіть в поле Symbol першого рядка in_1, в поле Adress I0.1, поле Type заповниться автоматично, в полі Symb-Comment можна написати коментар (опис), що пояснює призначення даного входу (виходу). Натисніть кнопку зберігання. Поверніться в організаційний блок (OB1) та подивіться, що вийшло

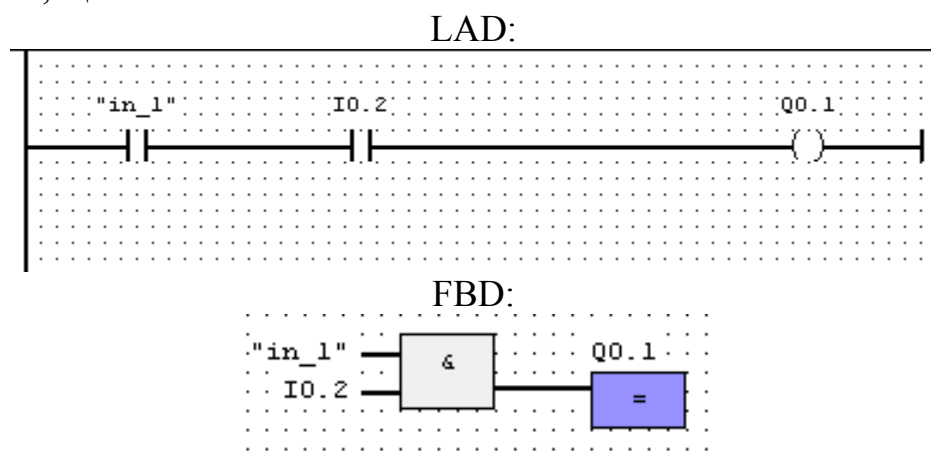


Рисунок 5.4 – Вигляд організаційного блока після зазначення входів\виходів у Symbolic editor

Також опишіть в Символьному редакторі вхід I0.2 як in_2, а вихід Q0.1 як out та збережіть.

Примітка. Абсолютна адреса може бути помічена на будь-яке символічне ім'я за вашим вибором.

Створіть функціональний блок, для цього натисніть Ctrl+N або на панелі інструментів введіть FB1 та натисніть ОК.

Всі дані передаються функціональному блоку (FB) з організаційного блока (OB) як параметри блока і тому повинні бути визначені в таблиці опису змінних як вхідні і вихідні параметри ("in", "out").

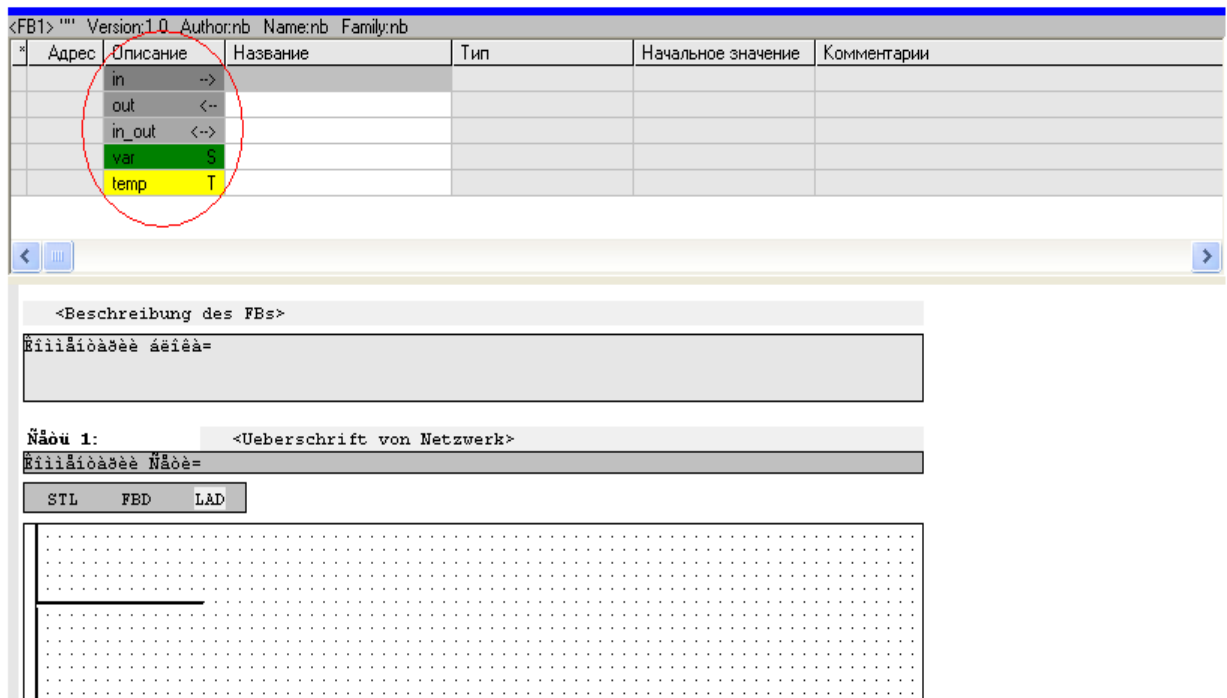


Рисунок 5.5 – Інтерфейсна панель функціонального блока (FB)

Введіть в поле Name значення op натисніть Enter, курсор переміститься в поле Type, в ньому просто натисніть ще раз Enter, буде автоматично вибраний тип змінної BOOL, натисніть ще раз Enter, курсор переміститься одразу в поле Comment, автоматично заповнивши поле Initial value значенням FALSE.

Після натискання Enter в полі Comment буде додана ще одна лінія входу (виходу). Аналогічним чином зробіть опис інших змінних, що зображені на рисунку нижче.

Подивіться на зелений рядок "var S", тут буде опис локальних змінних, значення яких будуть зчитуватися з блока даних (DB).

Зробіть опис змінної temp типу BOOL початкове значення якого FALSE.

Складіть коло за рисунком:

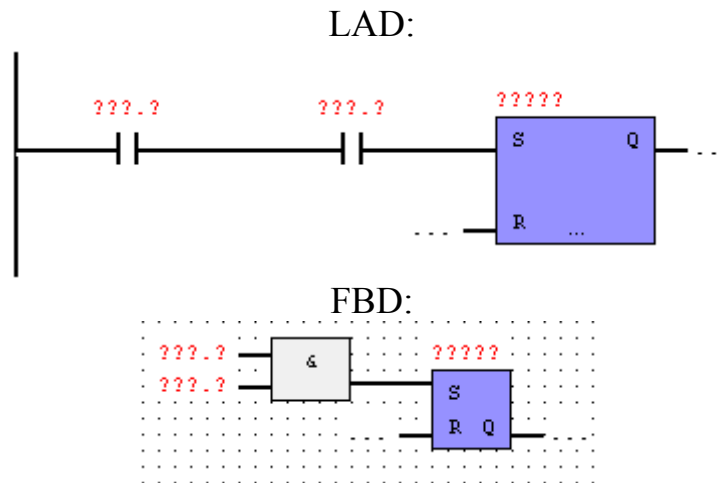


Рисунок 5.6 – Організаційний блок (ОБ) з SR тригером

Виберіть ім'я для першого входу, для цього виділіть його: натисніть Enter для початку введення і введіть знак #, після чого натисніть Enter, буде відкрите вікно з описаними вище змінними, виберіть в вікні змінну on та натисніть Enter.

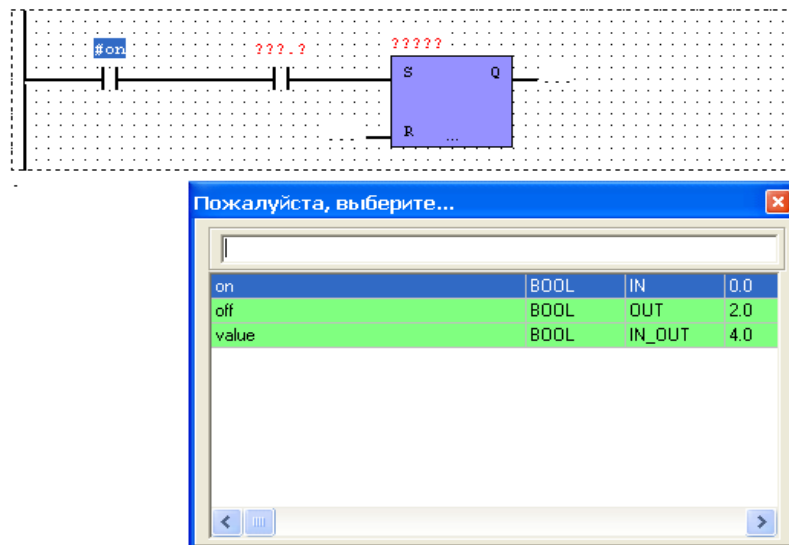


Рисунок 5.7 –Таблиця активних змінних у редакторі символів

Примітка. Зі знака # пишуться локальні змінні блока. Глобальні змінні, що описані в Символьному редакторі (Symbolic editor), пишуться в подвійних лапках (''').

Далі виберіть другий вхід, натисніть Enter для початку введення, введіть (('' '')) (подвійні лапки)) та натисніть Enter, буде відкрите вікно з вибором глобальних змінних, що описані в Символьному редакторі (Symbolic editor), виберіть out та натисніть Enter. Аналогічно можете вибрати змінну value для ім'я SR елемента (можна також ввести вручну).

Далі додайте два елементи “I” (AND) та назвіть їх як показано на рисунку:

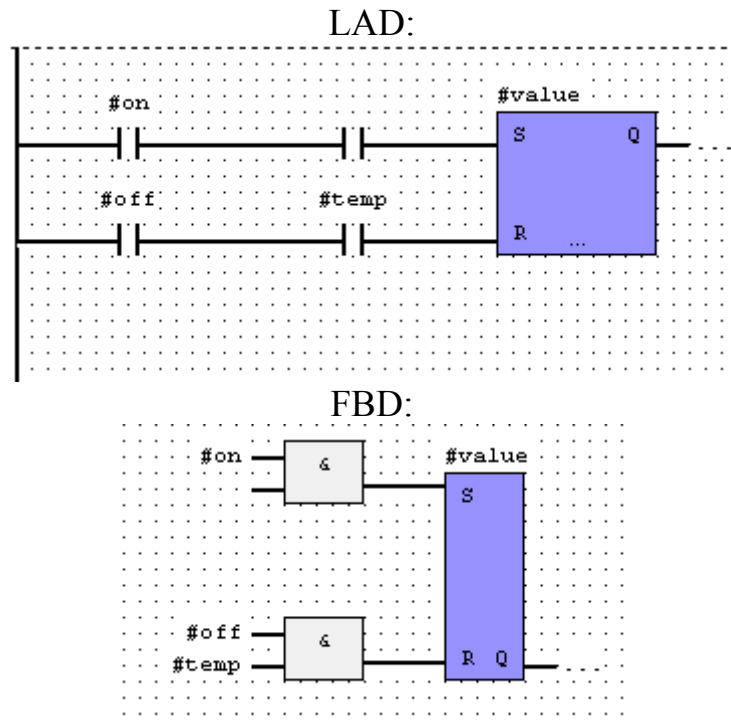


Рисунок 5.8 – Організаційний блок (OB) з паралельними ключами

Збережіть блок, натиснувши Ctrl+S або на панелі задач. Натисніть Ctrl+N або на панелі задач, у вікні створення нового блока, що з’явилося, введіть DB1 та натисніть ОК, в наступному вікні, що з’явилося, виберіть ”Data block with assigned function block” і в списку виберіть FB1.

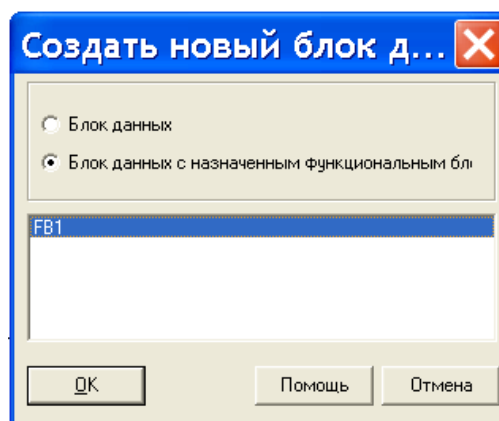


Рисунок 5.9 – Створення нового блока даних

Збережіть блок, натиснувши Ctrl+S. Поверніться в організаційний блок (OB1), натисніть Ctrl+R або на панелі задач та виберіть ”Insert after network1”

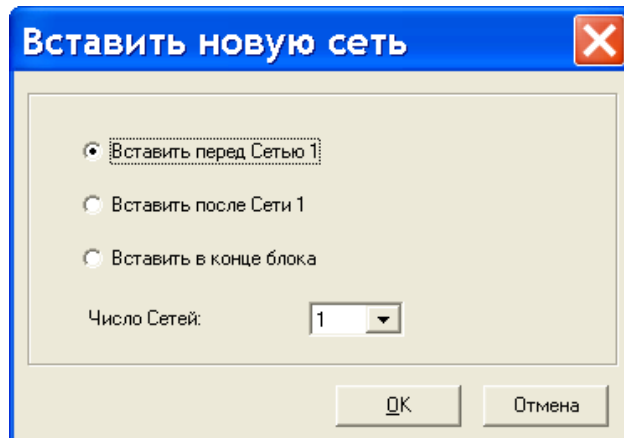


Рисунок 5.9 – Створення нового рядка

Додайте функціональний блок FB1 в організаційний блок OB1. Далі відкрийте Символьний редактор та додайте туди змінну result для виходу Q0.2.

Примітка. Можна також додати ім'я для організаційного, функціонального блоків і блока даних, додавши необхідне ім'я в поле Symbol, а в поле Adress ввести OB1, FB1 чи DB1, відповідно і натиснувши Enter (поле Type заповниться автоматично).

Виділіть "???" (DB)" вище функціонального блока та натисніть Enter, введіть db1 або в лапках ім'я, що описане в Символьному редакторі. Виділіть вихід функціонального блока (value) і введіть "result", натисніть Enter, виберіть зі списку result і знову натисніть Enter). Для кожного з входів On чи Off впишіть значення false або true.

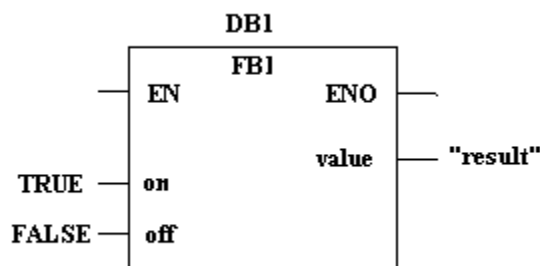


Рисунок 5.10 – Створений блок даних DB

Значення блока даних (DB) необхідно встановити перед запуском програми (до відсилання всіх блоків в контролер). Задайте значення TRUE змінній temp.

5.2 Завдання

Варіант 1. Розробити програму додавання двох чисел типу integer з використанням функціональних блоків і блоків даних. На вхід

функціонального блока подається перше число, а друге видобувається з блока даних. Результат вивести в пам'ять з адресою mw1.

Варіант 2. Розробити програму порівняння двох чисел типу integer з використанням функціональних блоків і блоків даних. На вхід функціонального блока подається друге число, а перше видобувається з блока даних. Вихід Q0.1 повинен встановитися в одиницю, якщо перше число більше чи дорівнює (\geq) другому.

Варіант 3. Розробити програму, яка знаходить різницю між двома змінними типу integer і ту, що порівнює отриманий результат зі значенням змінної з блока даних. Вихід Q0.3 повинен встановитися в одиницю, якщо різниця двох чисел менша змінної з блока даних.

Варіант 4. Розробити програму, яка знаходить добуток двох чисел типу integer і ту, що порівнює результат з третім. Перше і третє числа подаються на вхід функціонального блока, а друге видобувається з блока даних. Вихід Q0.1 повинен встановитися в одиницю, якщо добуток більше чи дорівнює (\geq) третьому числу.

Варіант 5. Розробити програму з використанням функціональних блоків і блоків даних, яка порівнює два числа (A і B) якщо A більше C, то розрахувати суму чисел A і B. Результат додавання вивести в пам'ять mw6. Число C видобувається з блока даних.

5.3 Зміст звіту

1. Мета роботи.
2. Абсолютна адресація.
3. Символьний редактор.
4. Створений новий блок даних.
5. Лістинг програми зі створеним блоком даних (відповідно до варіанта завдання).
6. Приклад роботи програми.
7. Висновки.

Контрольні запитання

1. Які функції виконує організаційний блок OB1?
2. Що таке абсолютна адресація?
3. Що таке символна адресація?
4. Як перейти від абсолютної адресації до символної адресації?
5. Які функції виконує блок даних (DB)?
6. Локальні змінні блока даних.
7. Глобальні змінні блока даних.
8. Функції редактора символів (Symbolic editor).
9. Які функції виконує функціональний блок FB1?

Лабораторна робота № 6

ВИКОРИСТАННЯ ПЛК VIRA ДЛЯ РОЗРОБКИ СИСТЕМ АВТОМАТИКИ НА ПРИКЛАДІ ТЕРМОРЕГУЛЯТОРА

Мета роботи: вивчення принципу дії терморегулятора за допомогою ПЛК VIRA 200 V. Вимірювання і аналіз вольт-амперних характеристик терморезисторів, визначення температурних коефіцієнтів опору. Ознайомлення з технологією виготовлення і можливим застосуванням терморезисторів.

6.1 Основні теоретичні відомості

6.1.1 Принцип роботи терморезисторів

Терморезистори (термістори) являють собою напівпровідникові резистори з нелінійною вольт-амперною характеристикою, відмінною особливістю яких є різко виражена температурна залежність електричного опору в діапазоні від -100 до 200 °С. Найбільше поширення одержали терморезистори, опір яких зменшується при збільшенні температури, тобто, терморезистори з від'ємним температурним коефіцієнтом опору (ТКО). Разом з тим, існують резистори, опір яких збільшується з ростом температури. Їх зазвичай називають позисторами. Позистори виготовляють на основі титанато-барієвої кераміки.

Терморезистори з від'ємним ТКО, виготовлені з напівпровідникових матеріалів. Зменшення опору напівпровідника зі збільшенням температури може бути обумовлено різними причинами – і збільшенням концентрації носіїв заряду або збільшенням їхньої рухливості, а також фазовими перетвореннями.

Перше явище характерне для терморезисторів, виготовлених з германію, кремнію, карбиду кремнію. Температурна залежність питомого опору напівпровідника визначається в основному зміною концентрації носіїв заряду, тому що відносно слабкою зміною їхньої рухливості в більшості випадків можна знехтувати.

При абсолютному нулі температури всі енергетичні рівні валентної зони невідродженого напівпровідника зайняті електронами. У цьому випадку валентні електрони не можуть брати участі в електричному струмі, тому що будь-який їхній рух пов'язаний зі збільшенням енергії й, відповідно, з переходом на більш високий енергетичний рівень, що не можливо в межах валентної зони. Тому при $T = 0$ К напівпровідник подібний до ізолятора, і його провідність дорівнює нулю. Для переходу

електрона в зону провідності напівпровідника без сумішей, необхідно передати йому енергію, рівну ширині забороненої зони ΔE_g . Таку енергію валентні електрони можуть одержати, якщо кристал нагріти до деякої температури. Завдяки наявності вільних рівнів у зоні провідності, електрони, що перейшли туди, зможуть рухатися під дією електричного поля. Провідність напівпровідника в цьому випадку буде обумовлена не тільки наявністю електронів у зоні провідності, але й появою дірок у валентній зоні.

Ймовірність переходів електронів з валентної зони в зону провідності, а, отже, і число вільних електронів, що утворилися, і дірок значно (за експонентним законом) зростають зі збільшенням температури

$$n_i \sim T^{3/2} e^{-\frac{\Delta E_g}{2kT}}, \quad (6.1)$$

де n_i – концентрація вільних електронів (індекс i вказує на те, що напівпровідник власний; помітимо, що у власному напівпровіднику концентрація вільних дірок $p = n_i$);

ΔE_g – ширина забороненої зони, що сама залежить від температури;

T – абсолютна температура;

K – стала Больцмана.

Якщо в напівпровіднику є домішки, то це приводить до виникнення енергетичних рівнів усередині забороненої зони. Атоми навіть при відносно низьких температурах можуть поставляти електрони в зону провідності (у цьому випадку домішка називається донорною, а напівпровідник – n -типу) або дірки у валентну зону (домішка називається акцепторною, а напівпровідник – p -типу), тому що необхідна для цього енергія звичайно значно менша ширини забороненої зони.

Більшу частину терморезисторів, що випускаються промисловістю, виготовляють із полікристалічних оксидних напівпровідників, у яких переважає іонний зв'язок. Електропровідність цих матеріалів відмінна від електропровідності розглянутих вище ковалентних напівпровідників. Як правило, напівпровідниками є оксиди перехідних металів, для яких характерна наявність незаповнених електронних оболонок і змінна валентність. При утворенні такого оксиду в певних умовах (наявність домішок, відхилення від стехіометрії) в однакових кристалографічних положеннях виявляються іони з різними зарядами. Електропровідність оксидних напівпровідників пояснюється обміном електронами між цими іонами. Тому, що енергія, яка необхідна для такого обміну, невелика, всі електрони (або дірки), які можуть переходити від одного іона до іншого, можна вважати вільним носіями заряду, а їхню концентрацію постійною при температурах у робочому для терморезистора діапазоні.

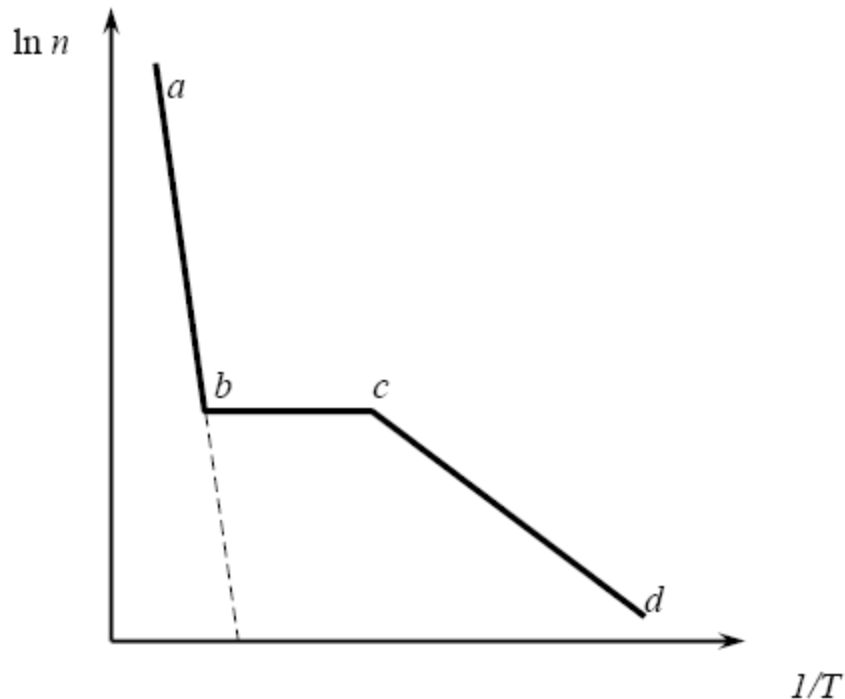


Рисунок 6.1 – Схематичне зображення температурної залежності концентрації електронів у домішковому (донорному) і власному напівпровідниках

На рисунку 6.1. зображено: штрихова лінія - власний напівпровідник; суцільна лінія - донорний напівпровідник; ab - область власної провідності донорного напівпровідника; bc - область виснаження домішки; cd - область слабкої іонізації домішки. Ділянки ab і cd не є прямолінійними через наявність слабкої температурної залежності передекспоненційного множника у вираженні для концентрації вільних електронів.

Через сильну взаємодію носіїв заряду з іонами рухливість носіїв заряду в оксидному напівпровіднику виявляється доволі низькою й експоненційно зростає з ростом температури. В результаті залежність опору оксидного напівпровідника від температури виявляється такою ж, як у ковалентних напівпровідниках, але вона обумовлена не зміною концентрації вільних носіїв заряду, а зміною їхньої рухливості.

В оксидах ванадію V_2O_4 й V_2O_3 , на відміну від розглянутих вище напівпровідників, причиною значних (на кілька порядків) змін їхнього опору є фазовий перехід при температурах 68 і – 110 °С, відповідно. На основі цих оксидів створені терморезистори з дуже великим температурним коефіцієнтом опору.

6.1.2 Технологія виготовлення й застосування терморезисторів

У масовому виробництві терморезисторів використовують методи керамічної технології, що основані на спіканні порошкових матеріалів у тверде компактне тіло певної форми і розмірів.

Терморезистори у формі циліндричних стрижнів, трубок, дисків і прямокутних пластин виготовляють із пластичної тістоподібної маси, що містить мілкодисперсні порошки напівпровідникових оксидів з органічним в'язучим і пластифікатором. Масу завантажують у циліндр мунштукового шприц-преса й видавлюють у вигляді довгих стрижнів або трубок, які після просушування нарізають на заготовки таких розмірів, які потребуються. Заготовки піддають випалу при високій температурі (близько 1000 °С і більше) в окисному, нейтральному або відновлювальному газовому середовищі залежно від виду напівпровідникових матеріалів, що використовуються. У процесі випалу органічне в'язуче вигорає, і матеріал спікається. При спіканні зразки набувають значну механічну міцність. Контакти створюють методом випалювання срібла або інших металів (золота, платини) з відповідних паст. При випалюванні у відбудовному газовому середовищі застосовують мідні й залізні пасту. Для підвищення стабільності й пригонки параметрів до заданих номінальних значень зразки піддають термообробці. Після цього на них надягають контактні ковпачки або припаюють провідники, що підводять струм, потім покривають шаром ізоляційної вологостійкої емалі або герметизують у захисних корпусах. Готові терморезистори піддають штучному старінню й сортують за величиною електричного опору. Циліндричні терморезистори виготовляють діаметром до 10 мм, довжиною від 1 до 50 мм.

Технологія виготовлення шайбових і дискових терморезисторів аналогічна тій, що застосовується для циліндричних і відрізняється тільки тим, що заготівлі одержують пресуванням у металевих прес-формах, іноді навіть без органічного в'язуючого. Диски роблять діаметром від одиниць до декількох десятків міліметрів, товщиною до 10 мм.

Кулькові терморезистори виготовляють із маси, що складається з порошкового напівпровідникового матеріалу, змішаного з відповідним пластифікатором. Крапельки цієї маси наносять на два платинові дроти, натягнуті паралельно один одному на відстані, яка в 5-10 разів перевищує діаметр дроту. Поверхневий натяг в'язкої рідкої маси надає крапельці форму кульки. Кульки просушують на повітрі й потім піддають випалу при температурі близько 1000 - 1300 °С у залежності від використовуваної сировини. Внаслідок спікання й усадки матеріалу кульки значно скорочуються в об'ємі та міцно зчіплюються із платиновим дротом. Отримані бусинки відокремлюють одну від одної, платинові виводи з'єднують із контактними дротами. Бусинки покривають шаром скла або

іншого захисного ізоляційного матеріалу. Діаметр бусинки може бути від 0,1 до 1-2 мм.

Терморезистори застосовують у системах виміру та регулювання температури, протипожежної сигналізації, теплового контролю та захисту машин і механізмів, у схемах температурної компенсації елементів електричних кіл, зокрема, для термокомпенсації кварцових резонаторів і генераторів, для стабілізації режимів роботи транзисторних каскадів, для виміру потужності, вакууму, швидкостей руху рідин і газів і т. д.

Описані вище терморезистори відносяться до класу терморезисторів прямого підігріву. На відміну від них терморезистори непрямого підігріву мають додаткове джерело тепла - підігрівач. Підігрівач роблять у вигляді обмотки, поміщеної на ізоляційну трубку, усередині якої розташований терморезистор, або у вигляді нитки розжарення, що розташована усередині трубчастого терморезистора.

Особливістю терморезисторів непрямого підігріву є наявність електрично ізольованих один від одного кіл – керуючого й керованого. Тому їх використовують в основному резистори як змінного опору без ковзного контакту, у телемеханіці для дистанційного керування різними системами, у радіотехнічних пристроях для автоматичного керування роботою підсилювачів, запобігання від перевантаження та у ряді інших випадків.

6.1.3 Параметри й характеристики терморезисторів

Температурна залежність опору терморезистора є однією з основних його характеристик. На рисунку 6.2 наведений приклад такої залежності. У всьому робочому діапазоні температур ця залежність досить добре описується рівнянням

$$R_{\Phi} = Ae^{\frac{B}{\Phi}}, \quad (6.2)$$

де R_{Φ} – опір терморезистора при абсолютній температурі T ;

A – величина, що залежить від матеріалу та геометричних розмірів терморезистора;

B – коефіцієнт температурної чутливості, що характеризує фізичні властивості матеріалу, з якого виготовлений терморезистор.

Φ – величина, що залежить характеру залежності опору від температури.

Температурна характеристика одного з терморезисторів

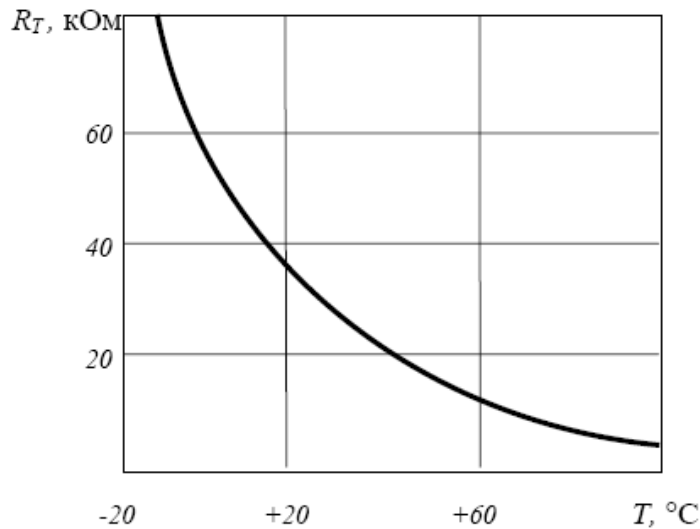


Рисунок 6.2 – Температурна характеристика терморезистора

Коефіцієнт температурної чутливості та сталу A можна легко визначити, якщо перебудувати температурну залежність опору терморезистора у спрямних координатах ($1/T, \ln R_T$). Очевидно, що нахил отриманої прямої буде визначатися значенням коефіцієнта B , а точка її перетину з віссю ординат ($1/T = 0$) – значенням сталої A . Для більшості типів терморезисторів значення коефіцієнта B лежать у межах 3000 – 16000 К.

Важливою характеристикою терморезистора є також його статична вольт-амперна характеристика (ВАХ). Вона має яскраво виражений нелінійний характер (рисунок 6.3).

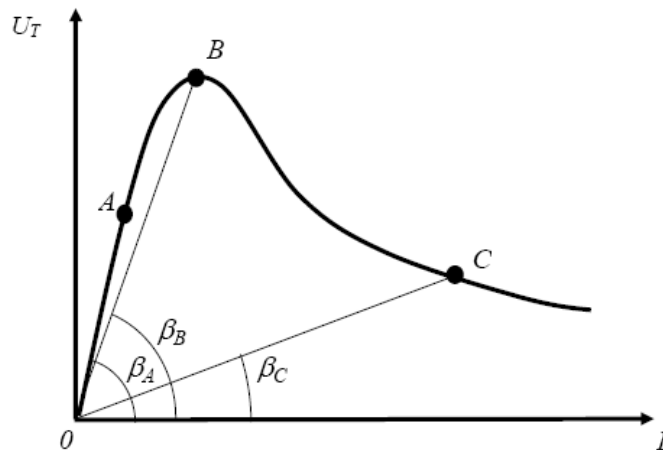


Рисунок 6.3 – Статична вольт-амперна характеристика терморезистора

ВАХ терморезистора можна розділити на три основних ділянки: OA , AB і BC . На початковій ділянці OA характеристика лінійна, тому що при

малих струмах потужність, що виділяється в терморезисторі, мала і температура терморезистора не дуже змінюється. На ділянці АВ лінійність характеристики порушується. З ростом струму розсіюється потужність, що збільшується, температура терморезистора помітно підвищується, а його опір зменшується. При подальшому збільшенні струму зменшення опорів виявляється настільки сильним, що напруга на терморезисторі зменшується, і з'являється ділянка ВАХ з негативним диференціальним опором (ділянка ВС на рисунку 6.3).

Статичний опір терморезистора в будь-якій точці характеристики визначається як тангенс кута нахилу β прямої, проведеної з початку координат в цю точку. З рисунка 6.3 видно, що величина статичного опору монотонно збуває з ростом струму.

Вольт-амперна характеристика терморезистора змінюється при зміні температури навколишнього середовища. На рисунку 6.4 зображені характеристики 1 і 2, що відповідають температурам навколишнього середовища T_1 і $T_2 > T_1$. Проаналізуємо зміну характеристики при постійному струмі, що протікає через терморезистор.

Опір терморезистора в точці D_1 при температурі навколишнього середовища T_1 характеризується кутом β_1 . При підвищенні температури навколишнього середовища опір терморезистора падає. Тому при температурі T_2 опір терморезистора повинен характеризуватися кутом $\beta_2 < \beta_1$. Інакше кажучи, точка D_1 переміщується в точку D_2 . Аналогічним чином зміняться положення інших точок характеристики. Отже, при підвищенні температури навколишнього середовища вся характеристика зміститься вниз.

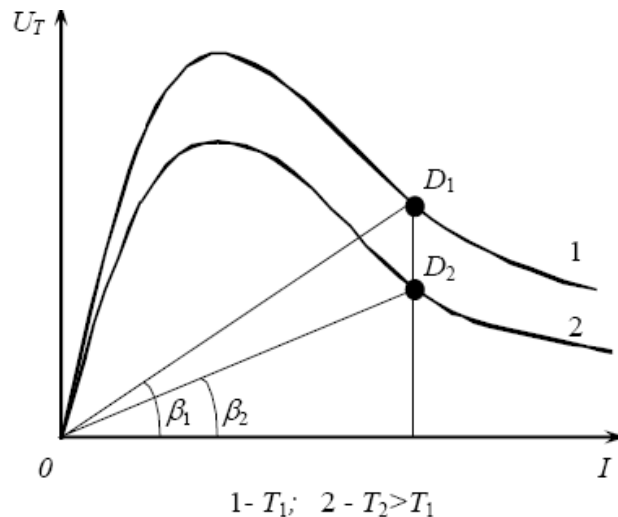


Рисунок 6.4 – Зміна статичної ВАХ терморезистора при зміні температури

Рівняння статичної вольт-амперної характеристики терморезистора може бути отримане з рівняння теплового балансу між потужністю P_T , що

виділяється в терморезисторі, і потужністю, що він розсіює в навколишній простір

$$P_T = I^2 R_T = \frac{U^2}{R_T} = H(T - T_0), \quad (6.3)$$

де H – коефіцієнт тепловіддачі (розсіювання), чисельно рівний потужності, що розсіює терморезистор при різниці температур терморезистора T і навколишнього середовища T_0 в один градус.

З рівняння (6.1) з врахуванням (6.2) одержимо рівняння статичної вольт-амперної характеристики терморезистора в параметричному вигляді

$$U = \sqrt{H(T - T_0) A \exp \frac{B}{T}}, \quad I = \sqrt{\frac{H(T - T_0) \exp(-\frac{B}{T})}{A}}. \quad (6.4)$$

До числа основних параметрів терморезистора відносять також:

Температурний коефіцієнт опору (ТКО) α_T , що характеризує зворотні відносні зміни опору при зміні температури на один градус

$$\alpha_T = \frac{1}{R_T} \frac{dR_T}{dT}. \quad (6.5)$$

Значення ТКО залежить від температури, тому опір записується з індексом T . Цю залежність можна одержати, скориставшись виразами (6.1) і (6.3)

$$\alpha_T = -B / T^2. \quad (6.6)$$

Значення ТКО при кімнатній температурі перебувають у межах $(0,3 - 20) \cdot 10^{-2} \text{ K}^{-1}$ [3].

Стала часу τ , що характеризує теплову інерційність терморезистора. Вона дорівнює проміжку часу, протягом якого температура терморезистора змінюється на 63% при перенесенні його з повітряного середовища з температурою $0 \text{ }^\circ\text{C}$ у повітряне середовище з температурою $100 \text{ }^\circ\text{C}$. Значення сталої часу лежить у межах від десятих секунди до декількох хвилин.

6.2 Практична реалізація. Лабораторний стенд

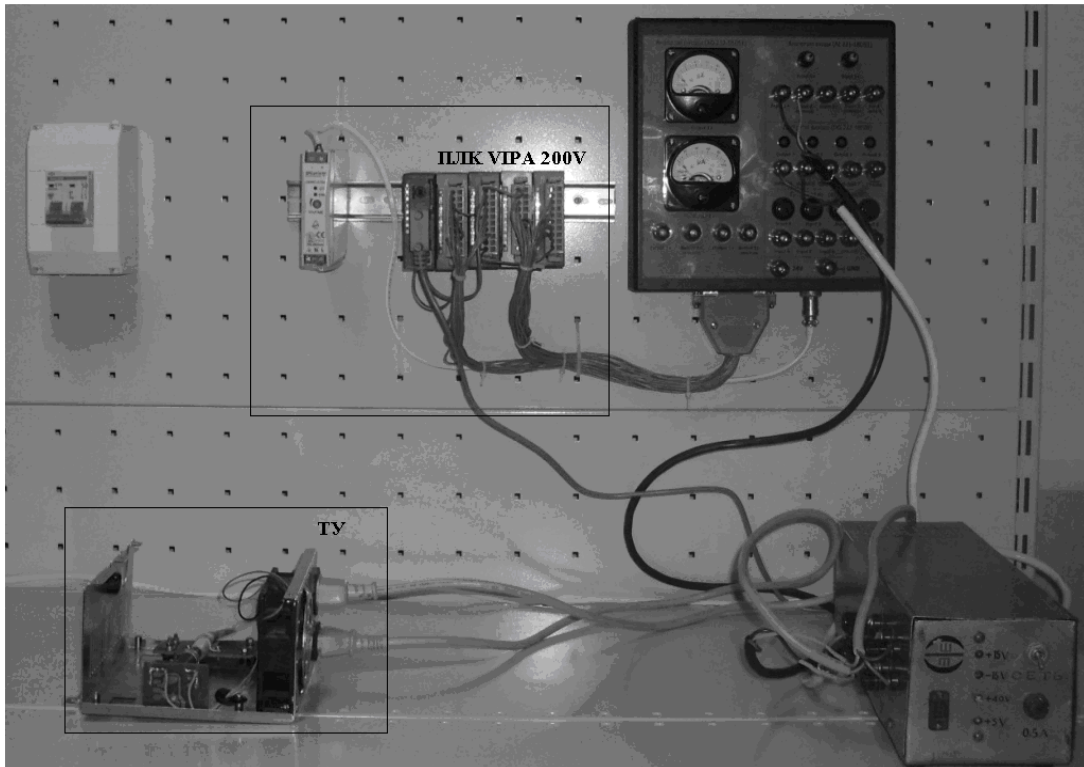


Рисунок 6.5 – Терморезистивна установка (ТУ) та ПЛК VIRA 200V

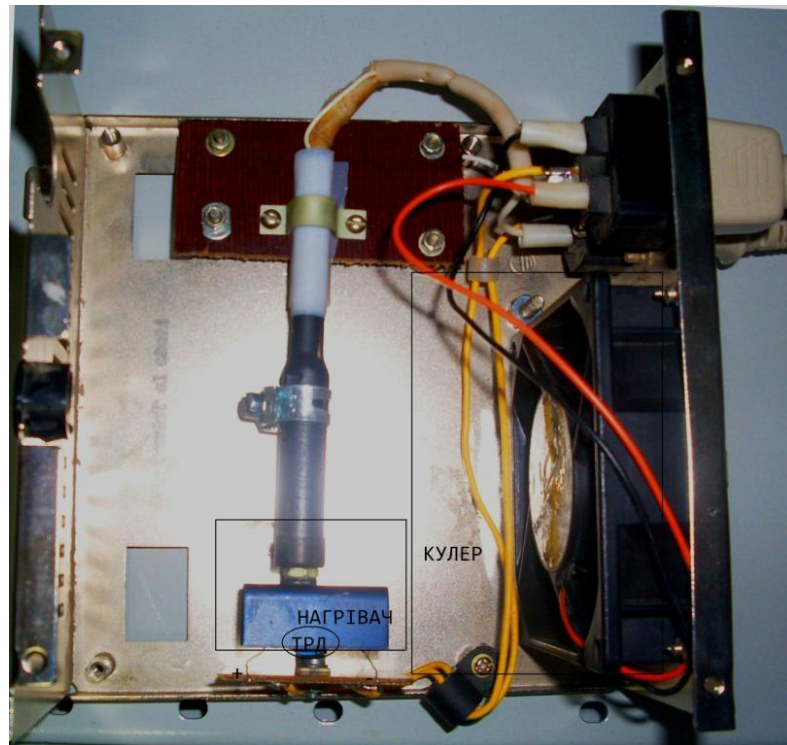


Рисунок 6.6 – Терморезистивна установка (ТУ)

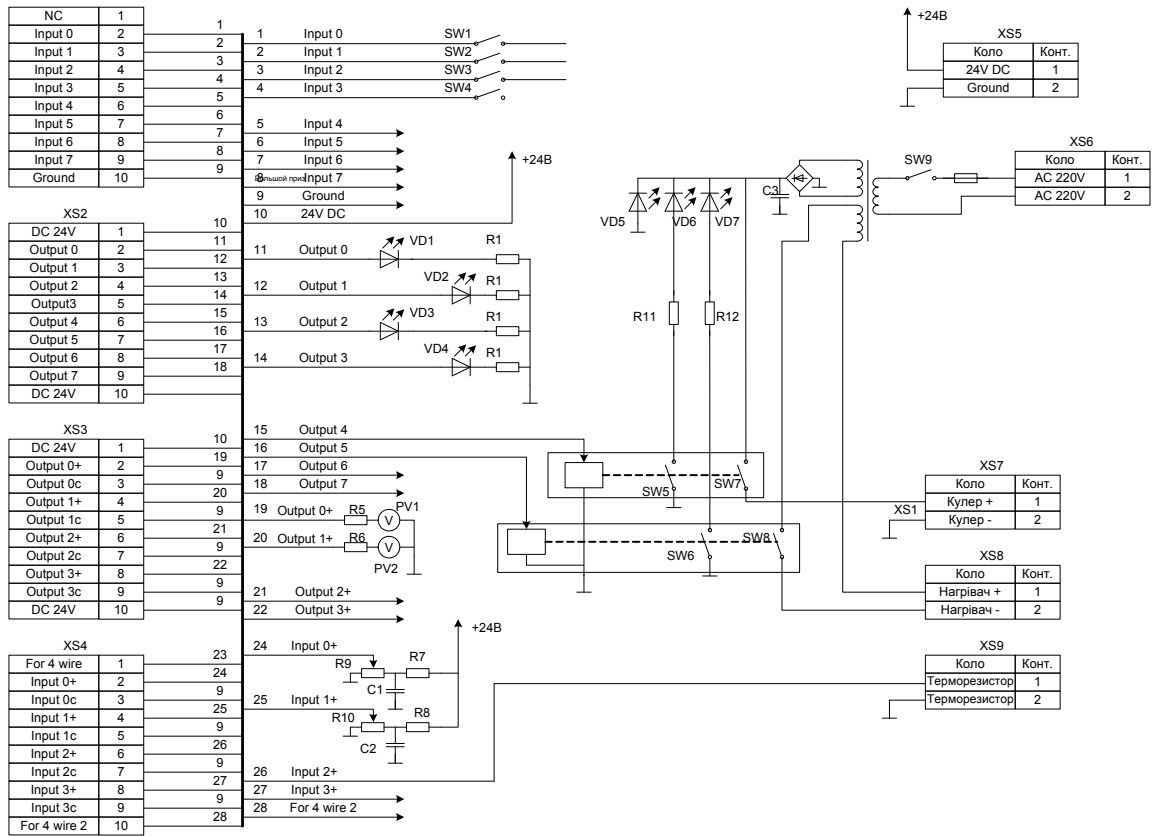


Рисунок 6.7 – Принципова схема лабораторного стенда

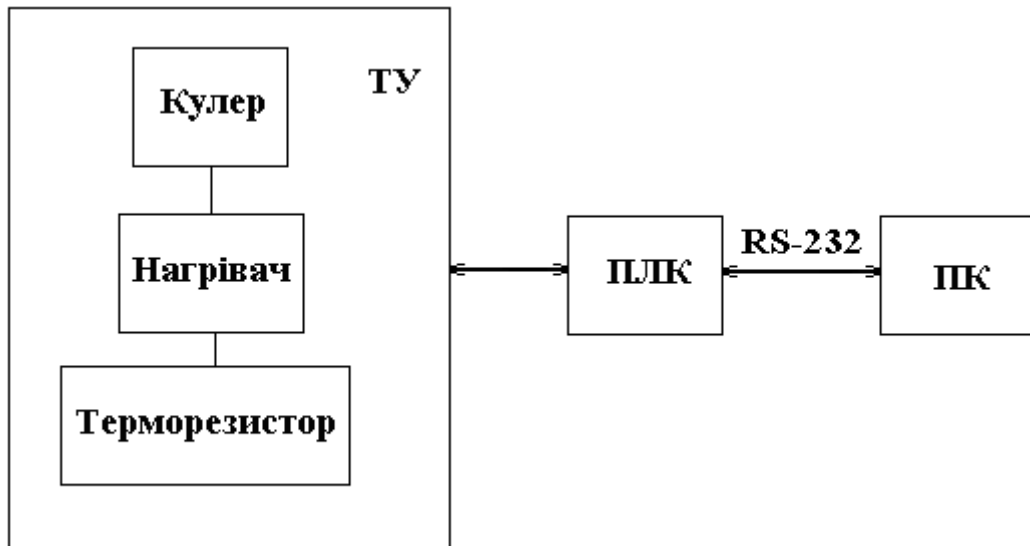


Рисунок 6.8 – Структурна модель лабораторного стенда

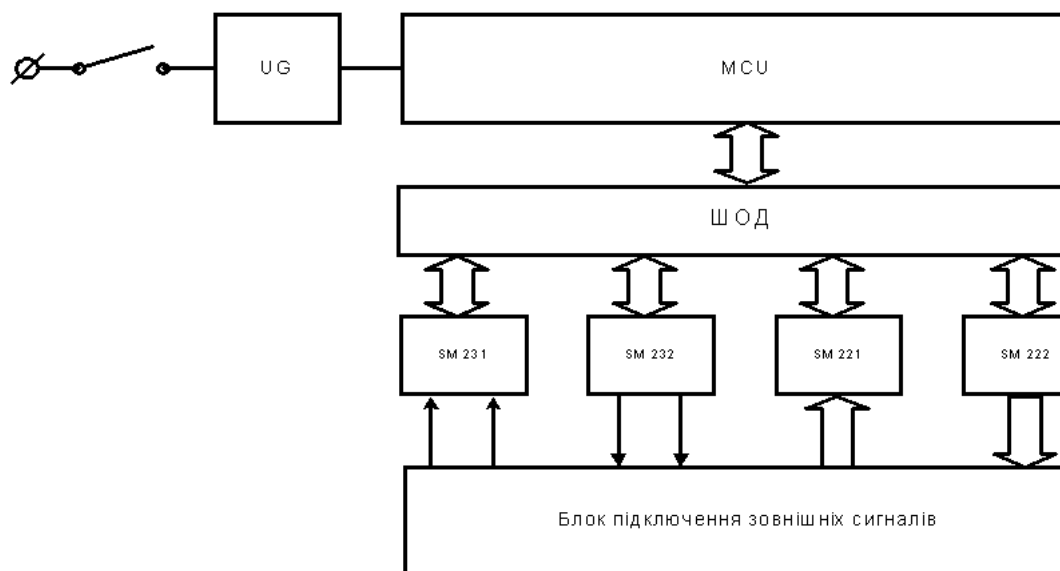


Рисунок 6.9 – Структурна схема лабораторного стенда

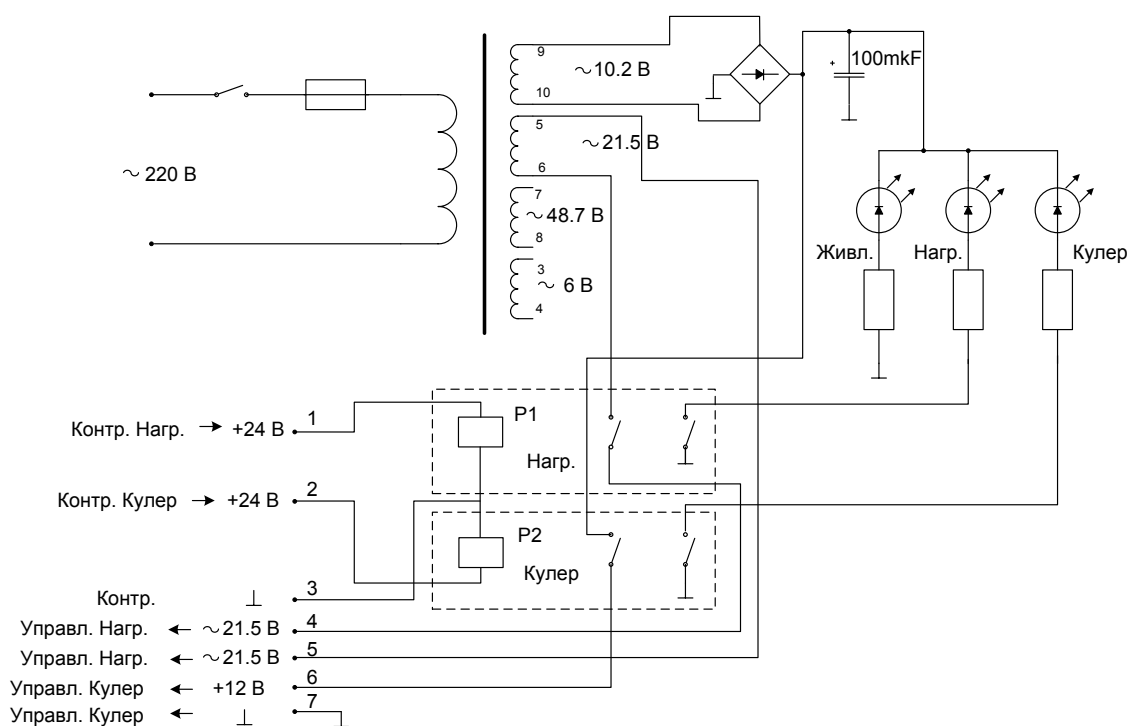


Рисунок 6.10 – Схема електрична принципова ТУ

Таблиця 6.1 – Експериментальна залежність опір – температура

Температура (°C)	Опір (Ом)	Температура (°C)	Опір (Ом)
15	1880	58	241
16	1830	59	230
17	1780	60	223
18	1730	61	214
19	1680	62	204
20	1630	63	193
21	1550	64	186
22	1450	65	180
23	1370	66	172,4
24	1320	67	164
25	1240	68	158,4
26	1200	69	152
27	1120	70	145,5
28	1060	71	140,2
29	1000	72	134
30	950	73	128
31	900	74	124
32	860	75	117
33	820	76	113,5
34	770	77	110
35	730	78	105,7
36	700	79	101
37	660	80	97
38	635	81	94,3
39	590	82	89,7
40	568	83	87,2
41	536	84	84,3
42	515	85	80,2
43	490	86	77,6
44	460	87	74,8
45	445	88	72,4
46	422	89	67,2
47	405	90	64,8
48	380	91	63,5
49	365	92	62,8
50	355	93	60,5
51	335	94	58,2
52	318	95	56,7
53	302,3	96	54,8
54	291	97	53
55	278	98	55,5
56	265	99	51,3
57	256	100	50,8

В нашому випадку терморезистор типу ММТ-12 (зовнішній вигляд терморезистора наведено на рисунку 6.11) має нелінійну залежність опір – температура, яка була отримана експериментально за допомогою еталонних вимірювальних перетворювачів. Графік залежності наведено на рисунку 6.12.

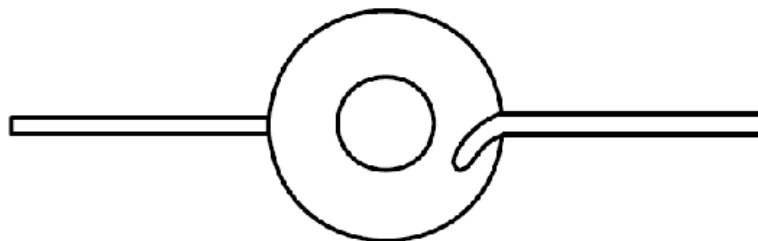


Рисунок 6.11 – Дисківий терморезистор типу ММТ-12

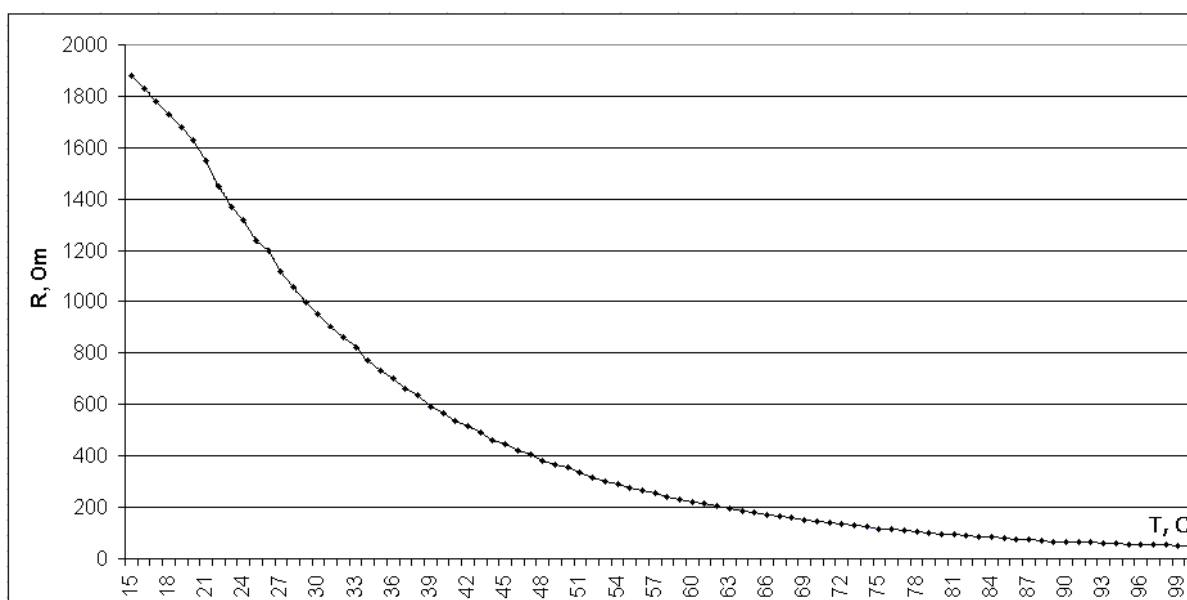


Рисунок 6.12 – Графічна залежність опір – температура

6.3 Порядок виконання роботи

Терморезистори являють собою напівпровідникові резистори з нелінійною вольт-амперною характеристикою, відмінною особливістю яких є різко виражена температурна залежність електричного опору [9, 10]. Найбільше розповсюдження отримали терморезистори, опір яких зменшується при збільшенні температури, тобто, терморезистори з від'ємним температурним коефіцієнтом опору. В залежності від температури навколишнього середовища терморезистори мають електричний опір, що пов'язаний з температурою нелінійною функцією, яка

описується формулою Стейнхард – Харта (метод параметричної ідентифікації) [11]

$$T = \frac{1}{C_0 + C_1 \cdot \ln R + C_2 \cdot (\ln R)^3}, \quad (6.7)$$

де T – температура, яка діє на терморезистор;

R – опір терморезистора;

C_0, C_1, C_2 – коефіцієнти, які підбираються з умови найкращого наближення до отриманої експериментальної залежності.

Коефіцієнти отримуємо експериментально при підстановці параметрів при температурі: 15, 50 та 100 градусів, тоді:

$$T = \frac{100}{7.2 - 2.1 \cdot \ln R + 0.036 \cdot (\ln R)^3}. \quad (6.8)$$

Лабораторний стенд призначений для прикладу практичної реалізації можливостей ПЛК Vira 200V. За допомогою контролера можна керувати кулером або нагрівачем. Для роботи системи на виходи контролера куди під'єднані елементи необхідно подати логічну “1” – тобто, щоб протікав струм. Якщо струм протікає, то індикатор виходу світиться, інакше струм не протікає і на вихід подано логічний “0”.

Після написання програми її необхідно зберегти та передати безпосередньо у пам'ять контролера у режимі RS-232.

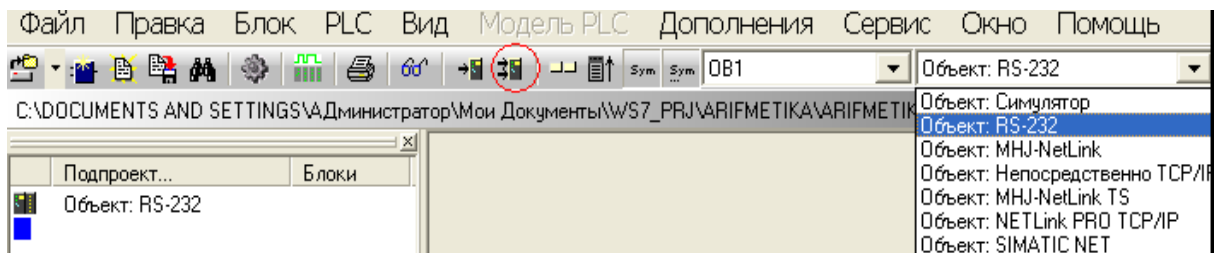


Рисунок 6.13 – Об'єкт RS-232

Після визначення робочих характеристик терморезистора можна перейти до розробки програмної частини. У програмі необхідно передбачити, щоб сенсор при визначенні опору це значення передав у ПК для подальшої обробки, тобто, для підстановки у отриману математичну модель для визначення температури навколишнього середовища. Приклад програми зображено на рисунку 6.14.

Опір терморезистора лежить в діапазоні від 0 до 4700 (Ом) і відображається за допомогою аналогового блока FC 105, а саме, на місці

змінної MD4 . Змінна MD4 відображає електричний опір терморезистора в реальному часі і передає значення опору на вхід математичного ланцюга, на виході якого відображається значення температури навколишнього середовища (змінна MD1).

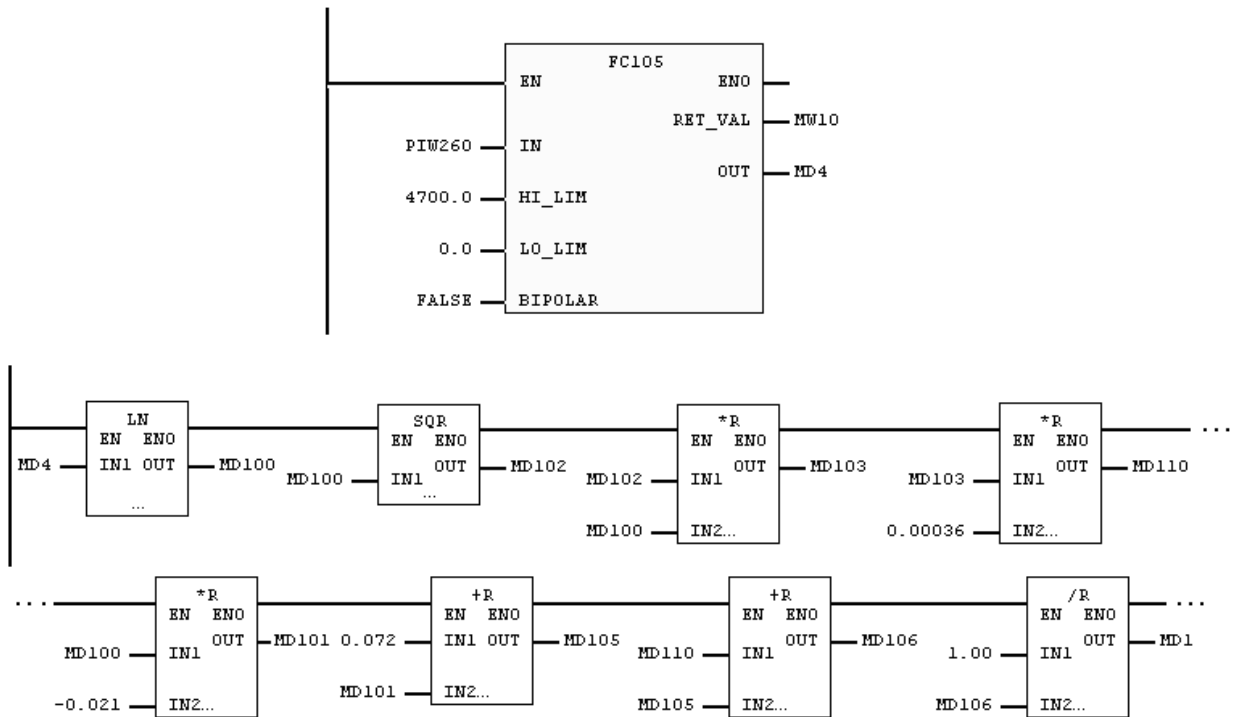


Рисунок 6.14 – Програмна реалізація математичної моделі у пакеті WINPLC7

Програма завантажена у контролер VIPA і зберігається у пам'яті. Після збереження програми керувати лабораторною установкою можна без допомоги комп'ютера.

6.4 Завдання

Варіант 1: При активації входу I0.1 активізується нагрівач, а при активації входу I0.2 активізується кулер.

Варіант 2: При активації входу I0.2 активізується нагрівач, а при активації входу I0.3 активізується кулер.

Варіант 3: При активації входу I0.3 активізується нагрівач, а при активації входу I0.4 активізується кулер.

Варіант 4: При активації входу I0.4 активізується нагрівач, а при активації входу I0.3 активізується кулер.

Варіант 5: При активації входу І0.1 активізується нагрівач, а при активації входу І0.4 активізується кулер.

6.5 Зміст звіту

1. Мета роботи.
2. Характеристика основних апаратних вузлів лабораторного стенда.
3. Лістинг програми математичної моделі терморезистора.
4. Лістинг програми керування терморегулятором (відповідно до варіанта завдання).
5. Приклади роботи програм.
6. Приклад роботи ПЛК (без зв'язку з ПК).
7. Висновки.

Контрольні запитання

1. Зобразити температурну залежність концентрації вільних носіїв заряду в домішковому напівпровіднику, охарактеризуйте кожну з ділянок цієї залежності.
2. Назвіть причини температурної залежності опору терморезисторів.
3. Поясніть причину нелінійності ВАХ терморезисторів.
4. Назвіть й охарактеризуйте основні параметри терморезисторів.
6. Як виготовляються терморезистори? Де вони використовуються?
7. Лабораторний стенд. Його функціональна, структурна, принципова схеми.
8. Терморезистивна установка. Терморезистор ММТ-12. Кулер. Нагрівач.
9. Подання формули Стейнхард – Харта в загальному вигляді.
10. Інтерфейс RS-232.

Додаток А

Лістинг програми реалізації терморегуляції мовою SL

```

CALL FC105
  IN:=PIW260
  HI_LIM:=4700.0
  LO_LIM:=0.0
  BIPOLAR:=FALSE
  RET_VAL:=MW10
  OUT:=MD4
NOP 0
L MD 4
LN
T MD 100
AN OV
SAVE
CLR
A BR
JNB _001
L MD 100
SQR
T MD 102
AN OV
SAVE
CLR
_001 :A BR
JNB _003
L MD 102
L MD 100
*R
T MD 103
AN OV
SAVE
CLR
_003 :A BR
JNB _004
L MD 103
L 0.00036
*R
T MD 110
AN OV
SAVE
CLR
_004 :A BR
JNB _006
L MD 100
L -0.021
*R
T MD 101
AN OV
SAVE
CLR
_006 :A BR
= Q 3.0
L 0.072
L MD 101
+R
T MD 105
AN OV
SAVE
CLR
A BR
JNB _005
L MD 110
L MD 105
+R
T MD 106
AN OV
SAVE
CLR
_005 :A BR
JNB _002
L 1.00
L MD 106
/R
T MD 1
AN OV
SAVE
CLR
_002 :A BR
= Q 1.0
A(
L MD 1
L 25.00
>R
)
= Q 0.4
= Q 0.1
A(
L MD 1
L 25.00
<R
)
= Q 0.5
= Q 0.0

```

АНГЛІЙСЬКО-УКРАЇНСЬКИЙ СЛОВНИК ОСНОВНИХ ТЕРМІНІВ

- ADD – додати;
Adress – адреса;
ACOS – арккосинус;
ASIN – арксинус;
ATAN – арктангенс;
AND – логічний елемент “Г”;
Bool – тип змінної;
CANOpen – промисловий інтерфейс;
CPU – ЦПУ;
COS – косинус;
Counter – лічильник;
Count down – зворотний рахунок;
Configure – конфігурація;
Comment – коментар;
Comparator – компаратор (пристрій порівняння);
DeviceNet – промисловий інтерфейс;
Display process – процес ініціалізації програми;
DINT – тип змінної;
DIN рейка – монтажна рейка;
DB – блок даних;
DIV – ділення з часткою як результат;
Enter – введення;
EXP – розрахунок експоненти за основою e ;
Editor – редактор;
Flash-ROM – флеш пам’ять;
False – “не правильно”, логічний “0”;
FB – функціональний блок;
Function Block Diagram (FBD) – мова функціональних блоків;
Green Cable – “зелений кабель”;
HollySys – системний інтегратор в області промислової автоматизації;
Hardware configuration – “конфігуратор заліза”, програма для зв’язку між мікроконтролером та комп’ютером;
Hot Restart – “горячий перезапуск”;
Interbus – шина;
IN – вхід;
INT – тип змінної;
Input – вхідний;
Ladder Diagram (LAD) – мова релейно-контактних схем;
LN – натуральний логарифм (логарифм за основою e);
MMC – карта флеш-пам’яті;
Multimedia Card – мультимедійна картка;
Monitoring – моніторинг;

MUL – множення;
OB – організаційний блок;
OR – логічний елемент “АБО”;
Output – вихідний;
Overall reset – повне скидання;
OUT – вихід;
On – ввімкнути;
Off – вимкнути;
Profibus – конфігуратор мережі промислових інтерфейсів;
PLC, Programmable Logic Controller – ПЛК, програмований логічний контролер;
PC – персональний комп’ютер;
RAM – оперативний запам’ятовувальний пристрій;
Result – результат;
REAL – тип змінної;
Reset – скинути;
RS-232, RS-485 – інтерфейси;
System – серія ПЛК;
Siemens – німецька компанія;
Statement List (STL) – список інструкцій;
STEP 7 – стандартне середовище розробки;
Set – встановити;
SUB – віднімання;
SIN – синус;
SQR – розрахунок квадрата числа;
SQRT – розрахунок квадратного кореня числа;
SAVE – зберегти;
Symbolic – символ;
TCP \ IP – промисловий інтерфейс для мережі Ethernet;
Timer – таймер;
True – “правильно”, логічна “1”;
TAN – тангенс;
VAR, variable – опис змінних;
Value – змінна;
VIPA – німецька компанія, спеціалізується на розробці і виробництві ПЛК;
WinPLC7, WinNCS – середовище розробки STEP7 з обмеженою функціональністю.

Список використаної літератури

1. Кучерук В. Ю. Програмування логічних контролерів SCHNEIDER ELECTRIC / Кучерук В. Ю., Поджаренко В. О., Кулаков П. І. – Вінниця : ВДТУ, 2002. – 132 с.
2. Схемотехніка електронних систем / [Бойко Б. І., Гуржій А. М., Жуйков В. Я. та ін.]. – К. : Вища школа, 2004. – 423 с.
3. Папернов А. А. Логические основы цифровой вычислительной техники \ Папернов А. А. – М. : Сов. радио, 1972.
4. Савельев А. Я. Арифметические и логические основы цифровых автоматов \ Савельев А. Я. – М. : Высшая школа, 1980.
5. Алексеенко А. Г. Основы микросхемотехники \ Алексеенко А. Г. – М. : Сов. радио, 1977.
6. Електронний ресурс. Режим доступу до сайту: www.WinPLC7.de
7. Електронний ресурс. Режим доступу до сайту: www.VIPA.de
8. Пасынков В. В. Полупроводниковые приборы / Пасынков В. В., Чиркин Л. К., Шинков А. Д. – М. : Высшая школа, 1973.-398 с.
9. Практикум по полупроводникам и полупроводниковым приборам / [под ред. К. В. Шалимовой]. – М. : Высшая школа, 1968. - 464 с.
10. Резисторы. Справочник / [под ред. И. И. Четвертакова и В. М. Терехова]. – М. : Радио и связь, 1991. – 527 с.
11. Викулин И. М. Физика полупроводниковых приборов / И. М. Викулин, В. И. Стафеев. – М. : Радио и связь, 1990. – 264 с.
12. Кучерук В. Ю., Севастьянов В. М., Дудатьев І. А. Лабораторний стенд для вивчення принципу дії терморегулятора / В. Ю. Кучерук, В. М. Севастьянов, І. А. Дудатьев // Вісник інженерної академії України. – м. Київ, 2009. – Бюл. № 3-4. – С. 174-176.
13. Кучерук В. Ю., Дудатьев І. А. Система вимірювання температури димових газів котельних установок / В. Ю. Кучерук, І. А. Дудатьев // Вісник ЧДТУ, 2011. – № 1. – С.123-128.
14. Мишель Ж. Программируемые контроллеры: архитектура и применение / Мишель Ж. — М. : Машиностроение, 1986. – 239 с.
15. Парр Э. Программируемые контроллеры: руководство для инженера / Парр Э. — М. : БИНОМ. Лаборатория знаний, 2007. — 516 с.
16. Петров И. В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования / Под ред. проф. В. П. Дьяконова. — М. : СОЛОН-Пресс, 2004. — 256 с.
17. Денисенко В. В. Компьютерное управление технологическим процессом, экспериментом, оборудованием. — М. : Горячая Линия-Телеком, 2009. – 608 с.
18. Андриюшенко О. А. Электронные программируемые реле серий EASY и MFD-Titan / О. А. Андриюшенко, В. А. Водичев. — Одесса : Одесский национальный политехнический университет, 2006. — С. 223.

19. Hans Berger. Automating with the SIMATIC S5-135U 3rd / Hans Berger. – Berlin: 1993.
20. Программируемые контроллеры: руководство для инженера / Э. Парр. — М. : БИНОМ. Лаборатория знаний, 2007. — 516 с.
21. Минаев И. Г. Программируемые логические контроллеры. Практическое руководство для начинающего инженера / И. Г. Минаев, В. В. Самойленко – Ставрополь : АГРУС, 2009. – 100 с.
22. Минаев И. Г. Программируемые логические контроллеры в автоматизированных системах управления / И. Г. Минаев, В. М. Шарапов, В. В. Самойленко, Д. Г. Ушкур. 2-е изд., перераб. и доп. – Ставрополь : АГРУС, 2010. – 128 с.

Список рекомендованой літератури

23. Балашов Е. П. Микро- и мини ЭВМ / Балашов Е. П., Григорьев В. Л., Петров Г. А. – Л. : Энергоатомиздат, 1984. – 376 с.
24. Преснухина Л. М. Микропроцессоры / Преснухина Л. М. – М. : Высшая школа, 1986. – 351 с.
25. Токхайм Р. Микропроцессоры: Курс и упражнения / Токхайм Р. – М. : Энергоатомиздат, 1987. – 338 с.
26. Майоров С. А. Введение в микроЭВМ / Майоров С. А., Кириллов В. В., Приблуда А. А. – Л. : Машиностроение, Ленингр. отд., 1988. – 303 с.
27. Морисита И. Аппаратные средства микроЭВМ / М. : Мир, 1988. – 279 с.
28. Соучек Б. Микропроцессоры и микроЭВМ / Морисита И. – М. : Сов. радио, 1979. – 517 с.
29. Гивоне Д. Микропроцессоры и микрокомпьютеры: Вводный курс / Гивоне Д., Россер Р. – М. : Мир, 1983. 463 с.
30. Нестеров П. В. Микропроцессоры. Архитектура и ее оценка. / Нестеров П. В. – М. : Высшая школа, 1984. 104 с.
31. Уокерли Дж. Архитектура и программирование микроЭВМ / Уокерли Дж. – М. : Мир, 1984. – 486 с.
32. Хосе М. Ангуло. Микропроцессоры: Архитектура, программирование и проектирование систем / Хосе М. Ангуло. – Тбилиси : Ганатлеба, 1989.
33. Дирксен А. МикроЭВМ / А. Дирксен. – М. : Энергоиздат, 1982. – 328 с.
34. Хоровиц П., Хилл У. Искусство схемотехники / Хоровиц П., Хилл У. – М. : Мир, 1983. Т. 2. – 590 с.
35. Вуд А. Микропроцессоры в вопросах и ответах / Вуд А. – М.: Энергоатомиздат, 1985. – 185 с.
36. Басманов А. С. Микропроцессоры и однокристалль-ные микроЭВМ: Номенклатура и функциональные возможности / А. С. Басманов, Ю. Ф. Широков. – М. : Энергоатомиздат, 1988. – 127 с.

Лабораторний практикум

Кучерук Володимир Юрійович
Севастьянов Володимир Миколайович
Дудатьєв Ігор Андрійович

Мікропроцесори в ІВТ. Лабораторний практикум
у середовищі WinPLC7 з використанням ПЛК
VIPA

Лабораторний практикум

Редактор В. Дружиніна
Коректор З. Поліщук

Оригінал-макет підготовлено І. Дудатьєвим

Підписано до друку
Формат 29,7×42 ¼. Папір офсетний
Гарнітура Times New Roman
Друк різнографічний. Ум. друк. арк.
Наклад прим. Зам.№

Вінницький національний технічний університет,
навчально-методичний відділ ВНТУ.
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ. к. 2201.
Тел. (0432) 59-87-36.
Свідоцтво суб'єкта видавничої справи
серія ДК №3516 від 01.07.2009 р.

Віддруковано у Вінницькому національному технічному університеті
в комп'ютерному інформаційно-видавничому центрі.
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, ГНК, к. 114.
Тел. (0432) 59-87-38.
Свідоцтво суб'єкта видавничої справи
серія ДК №3516 від 01.07.2009 р.