

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ
для проведення лабораторних занять з курсу
«Комп'ютерні мережі»
(у двох частинах)

Частина 2

для студентів напряму підготовки 050202
«Автоматизація та комп'ютерно-інтегровані технології»
денної та заочної форм навчання

Затверджено
редакційно-видавничою
радою університету,
протокол № 2 від 25.06.2015 р.

Харків
НТУ «ХПІ»
2015

Методичні вказівки для проведення лабораторних занять з курсу «Комп'ютерні мережі» (у двох частинах) для студентів напряму підготовки 050202 «Автоматизація та комп'ютерно-інтегровані технології» денної та заочної форм навчання. Ч. 2 / Уклад. Подустов М.О., Лисаченко І.Г., Лобойко В.О., Шутинський О. Г. – Х. : НТУ «ХПІ», 2015. – 56 с.

Укладачі: М.О. Подустов
 І. Г. Лисаченко
 В.О. Лобойко
 О. Г. Шутинський

Рецензент: проф. І.Л. Красніков

Кафедра автоматизації хіміко-технологічних систем
та екологічного моніторингу

ВСТУП

Ці методичні вказівки розроблені для проведення лабораторного практикуму зі студентами денної та заочної форм навчання за напрямом «Автоматизація та комп'ютерно-інтегровані технології» з дисципліни «Комп'ютерні мережі». Видання є продовженням першої частини методичних вказівок до лабораторного практикуму, в якому вивчалися основи побудови та використання послідовних інтерфейсів.

Методичні вказівки містять необхідні теоретичні відомості для ознайомлення зі стандартами на інтерфейс *Ethernet*, який є апаратною основою побудови комп'ютеризованих систем управління технологічними процесами. Під час виконання лабораторного практикуму студенти вивчають основи використання протоколу *TCP/IP* в мережі *Ethernet*. Також у лабораторному практикумі розглянуті протокол *HTTP* та основні принципи реалізації *WEB*-технологій в контролерах, які все частіше використовують для дистанційного моніторингу і диспетчеризації автономних технологічних об'єктів. Виконання лабораторного практикуму дозволяє студентам більш ефективно засвоїти в подальшому теорію і практику застосування промислових мереж та програмних засобів доступу до даних в ПЛК.

Як апаратне забезпечення в лабораторному практикумі використані програмовані логічні контролери (ПЛК) виробництва компанії «*BO OBEH*» (Україна) [1] та *FESTO* (Німеччина) [2], а також мережне обладнання для створення локальної мережі.

Програмне забезпечення для виконання лабораторного практикуму потрібно для розроблення прикладного програмного забезпечення. Це середовище *CoDeSys V2.3* [3, 4] для програмування контролерів *ПЛК150 OBEH* і *SoftPLC PLCWinNT* та середовище *FST 4.10* [5] – для програмування контролерів *FC34 FESTO*.

Лабораторна робота 1

ОРГАНІЗАЦІЯ ЗВ'ЯЗКУ МІЖ *SoftPLC* ТИПУ *CoDeSys SP PLCWinNT* ТА *ПЛК150 ОВЕН* ЗА ПРОТОКОЛОМ *UDP*

1.1. Мета та час проведення лабораторної роботи

На цю лабораторну роботу відповідно до робочої програми вивчення дисципліни «Комп'ютерні мережі» відводиться 4 акад. год. аудиторного та 4 акад. год. позааудиторного часу (СПС).

Цілі лабораторної роботи:

- закріпити на практиці отримані теоретичні знання про основи використання протоколу *UDP* транспортної складової протоколу *TCP/IP*;
- вивчення основних принципів мережного обміну даними за протоколом *UDP* між *ПЛК150 ОВЕН* і *SoftPLC*.

1.2. Апаратне забезпечення лабораторної роботи

Лабораторна робота виконується на робочих місцях з ПК та підключеними до них *ПЛК150 ОВЕН*, які встановлені на стендах. На одному з комп'ютерів запущений *SoftPLC*, який в лабораторній роботі є джерелом та приймачем даних. Студенти створюють проекти в середовищі *CoDeSys V2* та налаштовують свої ПЛК, тобто *ПЛК150 ОВЕН*, таким чином, щоб приймати дані від *SoftPLC*, а також генерувати дані.

Для проведення лабораторної роботи використовується штатна локальна мережа кафедри з сервером, мережними пристроями та вузлами (ПК). Всі ПК та ПЛК підключені до локальної мережі кафедри, яка має такі параметри: *TCP/IP*-адреса мережі – 172.016.028.000, маска мережі – 255.255.255.000.

На рис. 1.1 наведено фрагмент схеми комунікаційних зв'язків у локальній комп'ютерній мережі, де на рисунку присутні комп'ютери, які виконують роль сервера і вузла із запущеним *SoftPLC* та *ПЛК150 ОВЕН*.

1.3. Програмне забезпечення лабораторної роботи

Для створення з'єднань через *Ethernet* за протоколом *UDP* між *ПЛК150 OBEH* та *SoftPLC* застосовується середовище *CoDeSys V2* і бібліотека *NetVarUDP_LIB_V23.lib*. Для запуску *SoftPLC* використовується програмний модуль *SP PLCWinNT*, який входить до складу *CoDeSys*.

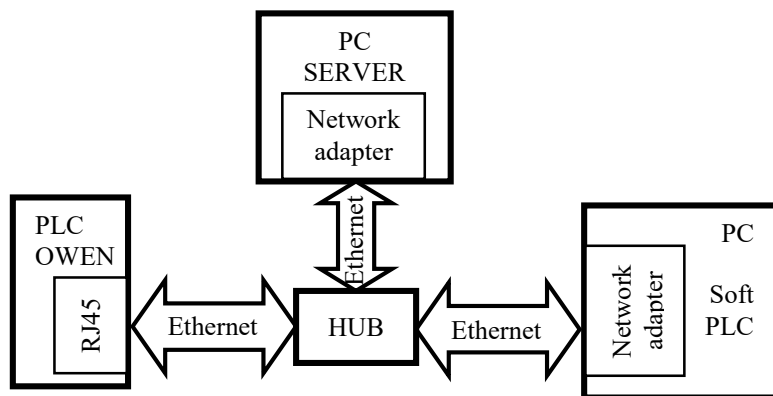


Рисунок 1.1 – Фрагмент побудови локальної мережі

ПРИМІТКА. Обмін даними за протоколом *UDP* можна реалізувати у середовищі *CoDeSys* на віртуальному контролері так званому *SoftPLC*. Цей ПЛК може бути запущений на ПК і працювати під управлінням операційної системи *MS Windows XP/Vista/7*. Для запуску *SoftPLC* необхідно в директорії зі шляхом доступу – *C:\Program Files\3S Software\CoDeSys SP PLCWinNT** запустити файл з ім'ям *PLCWinNT24.exe*. Далі, у вікні, що відкрилося, з'явиться повідомлення про те, що робота *SoftPLC* можлива протягом двох годин лише в демо-режимі. Необхідно натиснути на кнопку «OK». В результаті з'явиться вікно управління *SoftPLC*, яке показано на рис. 1.2.

У цьому вікні відображається вид з'єднання, повідомлення системи про статус ПЛК та кнопки управління режимами роботи ПЛК. Проте, для того, щоб контролер запрацював, необхідно заздалегідь створити

в середовищі *CoDeSys* проект для платформи *3S CoDeSys SP PLCWinNT24*. Особливість налаштування *SoftPLC* полягає лише в порядку підключення до нього. Оскільки *SoftPLC* запущений на локальному ПК, то можна при налаштуванні параметрів зв'язку вибрати «петльову» *IP*-адресу (127.0.0.1) або в рядок адреси потрібно ввести «localhost». Якщо *SoftPLC* запущений на видаленому ПК, то в рядок адреси потрібно вказати *IP*-адресу цього ПК. Імітувати значення змінних можна за допомогою створеної в проекті візуалізації.

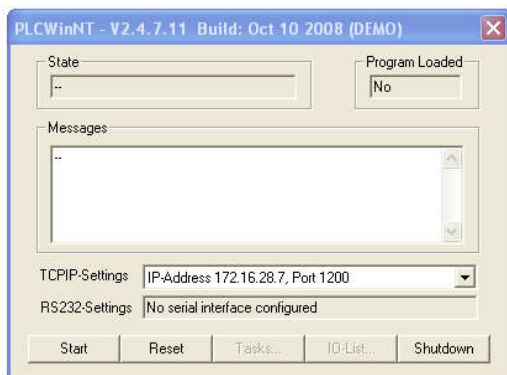


Рисунок 1.2 – Вікно управління *SoftPLC*

1.4. Короткі теоретичні відомості

Протокол *UDP* (скор. від англ. *User Datagram Protocol* – Протокол датаграм користувача) належить до стеку протоколів *TCP/IP* та виконує функцію транспортного сервісу. За допомогою протоколу *UDP* комп'ютерні прикладні програми можуть посилати повідомлення (так звані «датаграми») іншим програмам по *IP*-мережі без необхідності попереднього повідомлення про встановлення з'єднання. Протокол був розроблений Девідом П. Рідом у 1980 році й офіційно визначений як стандарт у коментарях з ім'ям *RFC 768* (скор. від. англ. *Request for Comments* – Запитання для коментарів).

Протокол *UDP* вважає, що нижчим протоколом багаторівневої моделі взаємозв'язку відкритих систем (*RM OSI*) є *Internet Protocol (IP)* [*RFC760*]. Цей протокол надає прикладній програмі (верхній рівень моделі *OSI*) процедуру для відправлення повідомлень іншим програмам, причому механізм виконання протоколу є мінімальним. Протокол *UDP* орієнтований на транзакції; при цьому отримання датаграм та захист від дублювання не гарантовані. Додатки, що вимагають гарантованого отримання потоків даних, повинні використовувати протокол управління пересиланням (*Transmission Control Protocol – TCP*) [*RFC761*], який також входить до стеку протоколів *TCP/IP*.

Для передачі даних протокол *UDP* використовує просту модель, без неявних «рукоприкладств» для забезпечення надійності, впорядкування або цілісності даних. Тобто *UDP* надає ненадійний сервіс, і датаграми можуть прийти не по порядку, дублюватися або зовсім зникнути. Протокол *UDP* має на увазі, що перевірка помилок і виправлення або не потрібні, або повинні виконуватися в прикладних програмах, тобто на верхніх рівнях моделі *OSI*. Прикладні програми, які критичні за часом виконання, використовують цей протокол, оскільки краще скинути, ніж чекати пакети, які затрималися, що може виявитися неможливим у системах реального часу.

Невеликі накладні витрати на використання протоколу *UDP*, відсутність необхідності підтвердження отримання пакета роблять цей протокол дуже популярним під час реалізації мережних сервісів та додатків, особливо тих, що пов'язані з локальними мережами та засобами мультимедіа. Хоча протокол *UDP* не гарантує доставки, за умовчанням передбачається, що вірогідність втрати пакета достатньо мала. Протокол *UDP* без збереження стану корисний для серверів, які відповідають на невеликі запити від величезної кількості клієнтів, наприклад сервіс *DNS* та поточкові мультимедійні застосування: *IPTV*, *Voice over IP*, тощо. Також протокол *UDP* використовує деякі мережні сервіси та додатки, наприклад *NFS* (*Network File System* – Мережна файлова система), *TFTP*

(*Trivial File Transfer protocol, RFC-1350* – Простий протокол передавання файлів), *RPC (Remote Procedure Call, RFC-1057* – Протокол виклику процедур дистанційного доступу) та *SNMP (Simple Network Management Protocol, RFC-1157* – Простий протокол управління мережею). В комп'ютеризованих системах управління теж можливе застосування протоколу *UDP*, наприклад, для швидкого обміну даними між ПЛК на декількох ділянках технологічної лінії.

Для встановлення з'єднання між хостами прикладні програми використовують так звані датаграмні сокети. Прикладна програма зв'язує свій сокет з його кінцевою точкою передачі даних (іншим сокетом), яка є комбінацією *IP*-адреси та сервісного порту. Порт – це програмна структура, яка визначається номером – 16-бітовим цілочисловим значенням (тобто від 0 до 65 535). Протокол *UDP* забезпечує багатоканальне передавання (за допомогою використання портів) та перевірку цілісності (за допомогою контрольних сум) заголовка та службових даних. Таким чином, якщо на рівні *IP* для визначення місця доставки пакету використовується адреса отримувача, то на рівні *UDP* – номер порту.

Процедура взаємодії у протоколі *UDP* виглядає таким чином. Прикладний процес, який надає деякі послуги (програма-сервер), чекає повідомлень, направлених у його порт, який спеціально виділений для цих послуг. Програма-сервер чекає, коли яка-небудь програма-клієнт запитає послугу. Дані, які відправляються прикладним процесом через модуль *UDP*, досягають місця призначення як єдине ціле. Наприклад, якщо процес-відправник проводить п'ять записів у порт, то процес-одержувач повинен буде зробити п'ять зчитувань. Розмір кожного записаного повідомлення повинен збігатися з розміром кожного прочитаного. Протокол *UDP* зберігає розмір повідомлень, які визначені прикладним процесом. Він ніколи не об'єднує декілька повідомлень в одне і не ділить одне повідомлення на частини.

Нижче, в табл. 1.1 наведено структуру заголовка пакета в протоколі *UDP*. Як видно з табл. 1.1, заголовок пакета протоколу *UDP* склада-

ється з чотирьох полів, кожне по два байти (16 біт). Два поля в заголовку не є обов'язковими для використання згідно зі стандартом *IPv4* (сірі поля в таблиці), проте у стандарті *IPv6* – необов'язково вказувати лише порт відправника.

Таблиця 1.1 – Структура заголовку пакета в протоколі *UDP*

Біти	0 – 15	16 – 31
0 – 31	Порт відправника (Source port)	Порт отримувача (Destination port)
32 – 63	Довжина датаграми (Length)	Контрольна сума (Checksum)
64 – ...	Дані (Data)	

Довжина повідомлення дорівнює кількості байт в *UDP*-датаграмі, включаючи заголовок. Поле *UDP* «контрольна сума» містить код, отриманий у результаті контрольного підсумовування *UDP*-заголовка та поля даних. Не важко помітити, що цей протокол використовує заголовок мінімального розміру (8 байт). У табл. 1.2 наведені деякі номери *UDP*-портів. Номери портів від 0 до 255 є стандартизованими і використовувати їх у прикладних завданнях не рекомендується. Але і в інтервалі 255...1023 багато номерів портів зайнято, тому перш ніж використовувати якийсь порт у своїй програмі, необхідно звертатися до стандарту *RFC-1700*. У другому стовпці зазначено стандартне ім'я, яке прийняте в *Internet*, а в третьому – записані імена, які прийняті в ОС *UNIX*.

Наведемо приклад. Нехай сервер *SNMP* завжди чекає повідомлення, адресованого в порт «161». Якщо клієнт *SNMP* бажає отримати послугу, він посилає запит в *UDP*-порт «161» на вузол, де працює сервер. На кожному ПК може бути тільки один агент *SNMP*, оскільки існує тільки один порт «161». Цей номер порту є загальновідомим, тобто фіксованим номером, офіційно виділеним у мережі *Internet* для послуг *SNMP*. Загальновідомі номери портів визначаються стандартами *Internet* (див. табл. 1.2).

Таблиця 1.2 – Стандартні номери *UDP*-портів (вибірковий перелік)

Десятк. номер порту	Позначення порту		Короткий опис та призначення порту
	в <i>Internet</i>	в <i>Unix</i>	
1	2	3	4
20	<i>FTP-data</i>	<i>ftp-data</i>	<i>FTP</i> (дані)
21	<i>FTP</i>	<i>ftp</i>	Протокол пересилання файлів (управління)
23	<i>telnet</i>	<i>telnet</i>	Підключення термінала
25	<i>SMTP</i>	<i>SMTP</i>	Протокол передавання пошто- вих повідомлень
42	<i>nameserver</i>	<i>name</i>	Сервер імен
49	<i>login</i>	–	Протокол входу до ПК
53	<i>Domain</i>	<i>nameserver</i>	Сервер імен доменів (<i>DNS</i>)
66	<i>SQL*net</i>	–	<i>Oracle SQL*net</i>
67	<i>Bootps</i>	<i>Bootps</i>	Протокол завантаження сервера
68	<i>Bootpc</i>	<i>bootpc</i>	Протокол завантаження клієнта
69	<i>TFTP</i>	<i>tftp</i>	Простий протокол передавання файлів
80	<i>WWW-HTTP</i>		<i>World Wide Web / HTTP</i>
92	<i>NPP</i>	–	Протокол мережного друку
115	<i>SFTP</i>		Простий протокол передавання файлів
118	<i>SQLserv</i>		<i>SQL</i> -сервер
161	-	<i>SNMP</i>	Мережний монітор <i>SNMP</i>
170	<i>Print-srv</i>		Мережний сервер друку

1.5. Послідовність проведення заняття

Лабораторна робота виконується комплексно. Її виконання складається з підготовки студентом контролера (*ПЛК150 OBEH*) для роботи в мережі *Ethernet* за протоколом *UDP* в якості пристрою для обміну мережними змінними з контролером викладача (*SoftPLC*).

Мережні змінні (для обміну даними між декількома контролерами) мають сенс, коли на одному технологічному об'єкті є декілька контро-

лерів, і необхідно обмінюватися даними між цими контролерами без створення додаткового каналу для мережного обміну. Обмін змінними може вестися лише по інтерфейсу *Ethernet* із застосуванням протоколу *UDP* через мережний порт *1202* (порт шлюзу *Gateway*, який використовують *ПЛК150 ОВЕН* в середовищі *CoDeSys*).

Студенти повинні підключитися до *SoftPLC* викладача та прочитати значення параметрів з відповідної папки з глобальними змінними, а також записати значення змінних в інші папки для глобальних змінних, які створені у *ПЛК* викладача для кожного *ПЛК* студента.

Кожен студент у середовищі *CoDeSys2.3* створює проект з *POU PLC_PRG* на мові *ST* для свого *ПЛК*. Для читання та запису даних в *ПЛК* студент повинен створити дві папки з *глобальними змінними*: одна – для отримання даних від *ПЛК* викладача (папка з ім'ям *soft_plc*, для читання), друга – для відправлення значень до *ПЛК* викладача (папка *PLC_X*, де *X* – номер стенда з *ПЛК*, від 1 до 10).

Початкові дані проекту в ПЛК викладача

Для реалізації проекту в *SoftPLC* викладача використовується таргет-файл *CoDeSys SP PLCWinNT*, який входить до комплексу *CoDeSys*. Вказаний *SoftPLC* встановлений та запущений на ПК (*S2*) з *TCP/IP*-адресою «*172.016.028.042*». Папка для читання змінних з *ПЛК* викладача знаходиться у вкладці ресурсів проекту в переліку *Глобальні змінні* і має номер ідентифікатора *COB-ID*, який дорівнює значенню «*11*». До складу папки входять такі змінні:

```
VAR_GLOBAL
    a1 : WORD;
    a2 : WORD;
    rez_udp : REAL;
    thk_udp: REAL;
    maska_din_udp: BYTE;
END_VAR
```

До переліку входять змінні для зберігання значення температури (*thk_udp*), опору змінного резистора (*rez_udp*), стану дискретних входів (*maska_din_udp*), а також дві змінні типу *WORD* (*a1*, *a2*), які генеруються в головному організаційному блоці *PLC_PRG* проекту в ПЛК викладача.

Лістинг організаційного блока *PLC_PRG* в проекті для *SoftPLC* для формування значень вказаних змінних наведено нижче:

```
a1:=a1+1;
a2:=a2+a1;
rez_udp:=rez;
thk_udp:=thk;
maska_din_udp:=PACK(din1, din2, din3, din4, din5, din6, 0, 0);
```

У цьому лістингу формуються циклічна зміна змінних *a1*, *a2*, передача значень від фізичних аналогових входів (*rez_udp*, *thk_udp*) в змінні папки глобальних змінних та операція пакування стану дискретних входів *SoftPLC* викладача в маску (*maska_din_udp*) у вигляді змінної типу *byte*.

Для кожного ПЛК студента в проекті для ПЛК викладача передбачені папки для зчитування даних, причому кожна має ім'я *PLC_N*, де *N* – номер стенда з ПЛК студента.

Увага! Регістр букв в імені папки має значення

Вміст папки з глобальними змінними студента повинен мати такий вигляд:

```
VAR_GLOBAL
  rez5: REAL;
  byte5: BYTE;
END_VAR
```

Тут, наприклад, змінні *rez5* і *byte5* відображають значення опору резистора та маски дискретних входів на об'єкті з ПЛК, який встановлений на стенді № 5. У табл. 1.3 вказані *TCP/IP*-адреса контролерів, номери стендів та номери комп'ютерів, до яких підключений ПЛК.

Таблиця 1.3 – Адреси та номери ПЛК та ПК у локальній мережі

№ з/п	TCP/IP-адрес ПЛК	Номер ПК	TCP/IP-адрес ПК
1	172.16.28.249	S 1	172.16.28.41
2	172.16.28.241	S 7	172.16.28.47
3	172.16.28.248	S 3	172.16.28.43
4	172.16.28.246	S 5	172.16.28.45
5	172.16.28.242	S 6	172.16.28.46
6	172.16.28.247	S 4	172.16.28.44
7	172.16.28.244	S 10	172.16.28.50
8	172.16.28.240	S 8	172.16.28.48
9	172.16.28.243	S 9	172.16.28.49
10	172.16.28.245	S 11	172.16.28.51

Фрагмент запущеного проекту в ПЛК викладача показаний на рис. 1.3. Тут показані вікна з реалізацією основної програми (PLC_PRG) і глобальними папками для читання (папка з ім'ям `soft_plc`) та запису (папка з ім'ям `PLC_5`), які отримані від ПЛК студента з номером «5».

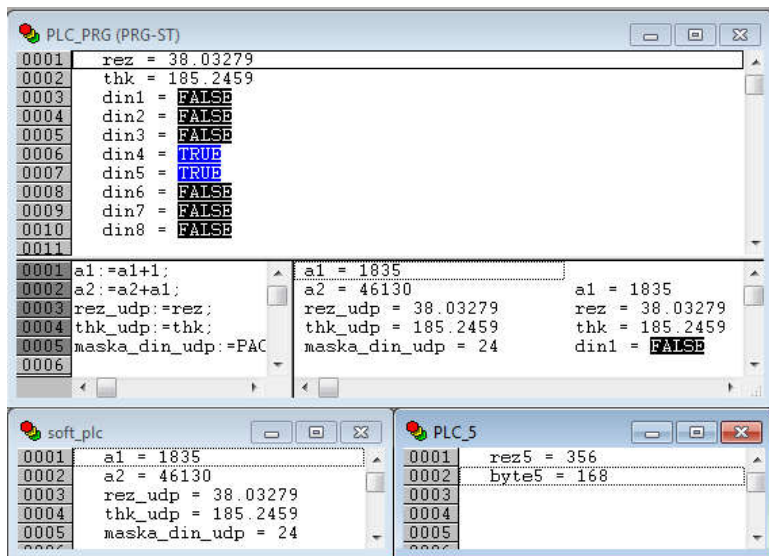


Рисунок 1.3 – Фрагмент запущеного проекту в ПЛК викладача

На рис. 1.4 та рис. 1.5 показані зовнішній вигляд вікон з візуалізацією для змінних в ПЛК викладача для зчитування з боку ПЛК студентів та вікно з візуалізацією для запису параметрів від ПЛК студентів.



Рисунок 1.4 – Візуалізація параметрів, які генеруються у ПЛК викладача

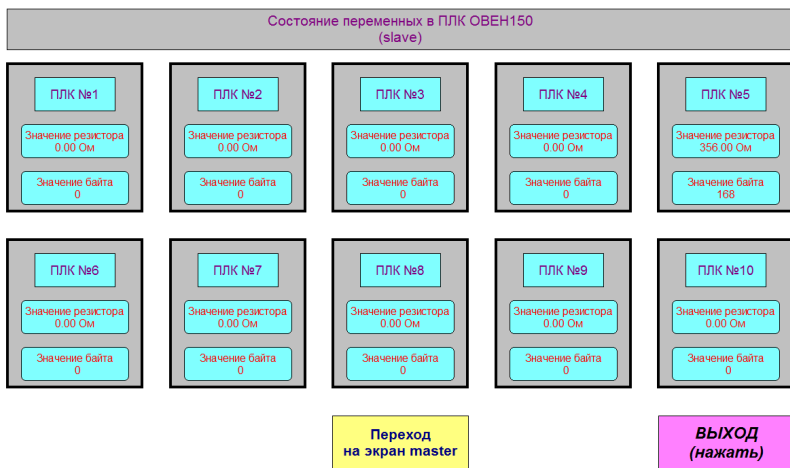


Рисунок 1.5 – Візуалізація параметрів, які генеруються у ПЛК студентів

Проект викладача запущений на платформі *SoftPLC* типу *CoDeSys SP PLCWinNT*, тому для імітування його сигналів використані віртуальні змінні. Дискретні сигнали імітуються віртуальними «кнопками» з позначками *D1...D8*, значення уставки та температури імітується за допомогою графічного елементу «слайдер». У візуалізації також створені кнопки для переходу в інший екран та для завершення роботи ПЛК.

Студенти повинні створити проекти з відповідною конфігурацією ПЛК та програмою, а також візуалізацію, яка подібна до зображеної на рис. 1.6.

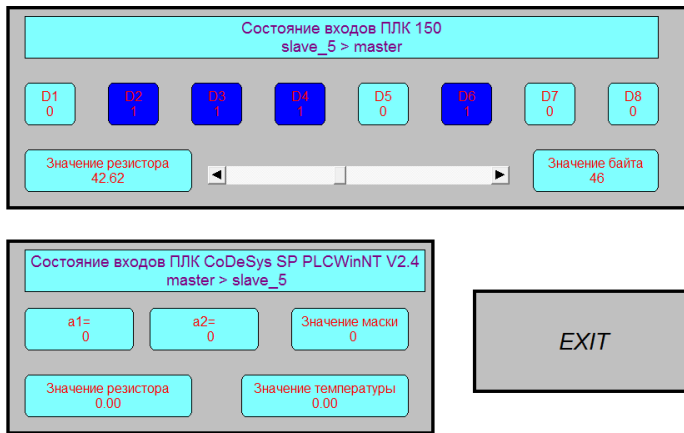


Рисунок 1.6 – Візуалізація проекту у ПЛК студентів

Отже, створіть проект у середовищі *CoDeSys*. Як таргет-файл для проекту оберіть модель *ПЛК150-I-L*. Мова реалізації основного програмного організаційного компонента *ST*. Далі встановіть мінімальний час циклу контролера рівним 10 мс та налаштуйте аналогові входи на оброблення сигналу від термомпари та резистора опору. Крім того, стан дискретних входів об'єднайте в основній програмі *PLC_PRG* у вигляді маски. Значення цих змінних необхідно привласнити глобальним змінним в папці *PLC_N*. Приклад конфігурації ресурсів ПЛК студента наведено на рис. 1.7. Для створення маски входів необхідно використати фрагмент лістингу програми для ПЛК викладача (див. лістинг на стор. 12).

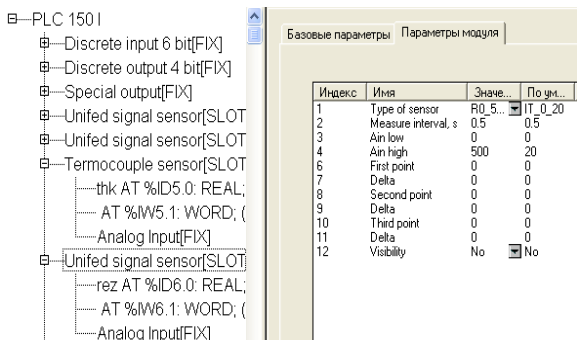


Рисунок 1.7 – Конфігурація ресурсів ПЛК у проекті студента

Далі, для підключення контролера до мережі *Ethernet* за протоколом *UDP*, необхідно відкрити вкладку проекту *Ресурси*, а в ній – категорію *Настройки целевой платформы*. У налаштуваннях ПЛК виберіть вкладку *Сетевая функциональность*, в якій у поле *Поддерживаемые сетевые интерфейсы* введіть тип мережного з'єднання – *UDP*. На рис. 1.8 показаний фрагмент з результатами проведених налаштувань.

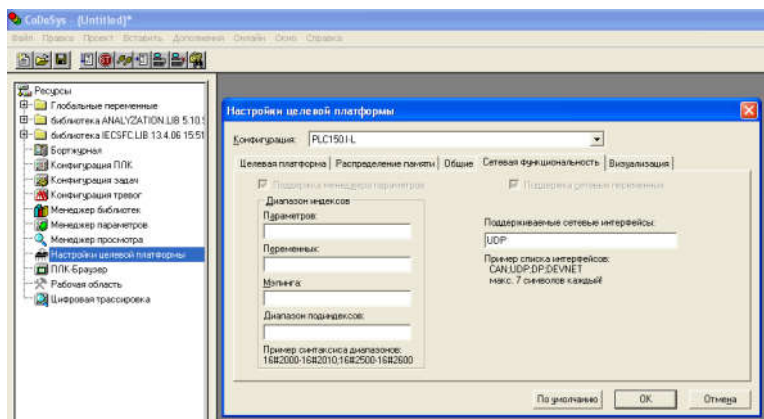


Рисунок 1.8 – Фрагмент налаштувань проекту у ПЛК студентів

До проекту необхідно підключити бібліотеки, які потрібні для організації обміну мережними змінними. Для цього відкрийте вкладку

проекту *Ресурси*, а в ній – категорію *Менеджер бібліотек* та підключіть бібліотеки *Syslibcallback.lib*, *Syslibsockets.lib*, *Netvarudp_lib_v23.lib*. Вказані бібліотеки можна знайти в папці зі встановленим середовищем *CoDeSys2.3* зі шляхом доступу *C:\program Files\3s Software\Codesys V2.3\library*.**. У разі відсутності необхідної бібліотеки уточніть її місцезнаходження у викладача. Бібліотеки вкладені в архіві або на диску, який додається до ПЛК, або на сайті компанії «ВО ОВЕН» (Україна) – www.owen.ua. Рекомендується перед підключенням бібліотек перенести їх в директорію, яка створена після інсталяції середовища *CoDeSys*, за умовчанням це: *c:\program Files\3s Software\codesys V2.3\library*.**

Далі створіть та налаштуйте папки з мережними змінними. Для цього відкрийте список *Global variables* у вкладці *Ресурси*, як це показано на рис. 1.9. Вікно *Свойства* викликається через контекстне меню з допомогою кліка правою кнопкою миші по команді *Свойства объекта....*

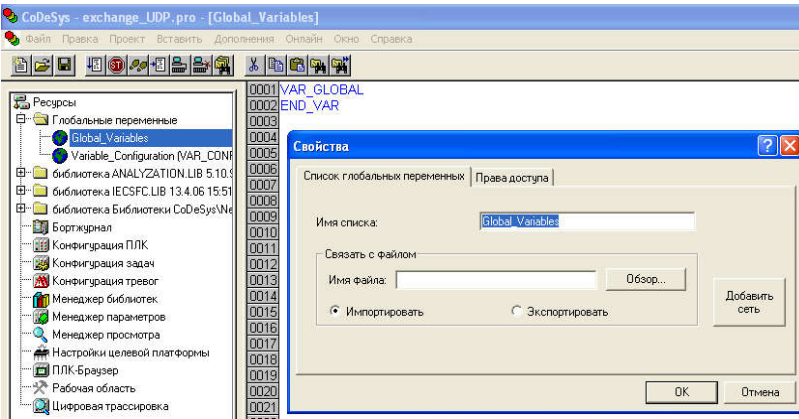


Рисунок 1.9 – Фрагмент з вікном налаштувань властивостей списку *Global variables* у проекті для ПЛК студентів

У вікні властивостей лівою кнопкою миші натисніть на кнопку *Добавить сеть*. Вікно властивостей розшириться до вигляду, показаного

на рис. 1.10. Це налаштування відповідає читанню змінних зі списку *Global variables* з номером з'єднання *COB-ID*, який дорівнює «1». Це відповідає списку змінних у ПЛК студента з номером «1». На рис. 1.10 показані налаштування списку *Global variables* у ПЛК студента для читання змінних з ПЛК викладача.

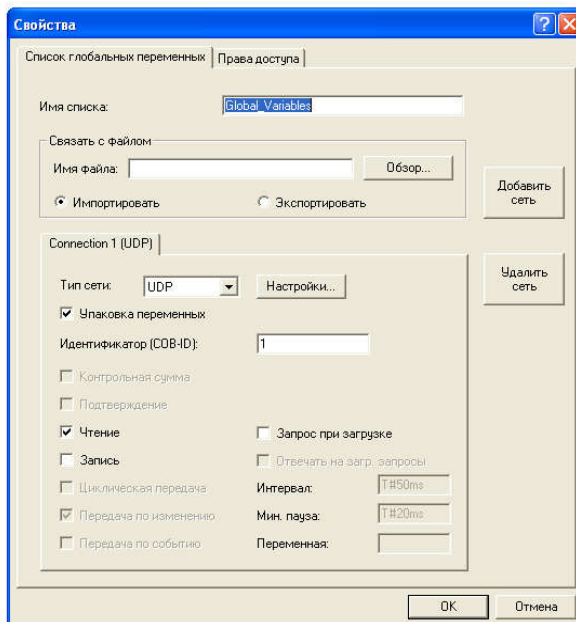


Рисунок 1.10 – Фрагмент з вікном налаштувань властивостей списку *Global variables* у проєкті для ПЛК студентів

Як раніше було зазначено, ім'я списку змінних повинне відповідати імені списку в ПЛК викладача, наприклад, папка `soft_plc` – для зчитування даних і папка `PLC_N` – для запису даних.

УВАГА! В обох (декількох) ПЛК ім'я цього списку має бути ідентичним імені в іншому ПЛК. При цьому регістр букв має значення

Крім того, для списків змінних необхідно вказувати функцію («читання» або «запис») та номер ідентифікатора з'єднання (*COB-ID*). Останній параметр встановлюється залежно від номера стенда з ПЛК для запису і ставиться значення «11» – для читання даних з ПЛК викладача (див. табл. 1.3). Тому в списку для отримання значень поставте позначку в полі «Чтение», а для передавання – в полі «Запись». Значення параметра «Ідентифікатор (*COB-ID*)» повинно відповідати завданню, наприклад, для читання даних з ПЛК поставте «11», для запису – номер стенда з ПЛК. Далі поставте позначку в полі визначення необхідної функції.

Якщо обрано функцію «Запись» – стають доступними для введення даних три поля, які вказують на режим обміну даними: *Циклическая передача, Передача по изменению значения* або *Передача по команде*.

УВАГА! Не рекомендується ставити одночасно позначки на читання та запис для одного і того списку в одному ПЛК

Таким чином, лише в одному контролері змінні одного списку мають бути в папці «Запись», в решті контролерів – в папці «Чтение». Якщо необхідно, щоб у кожного контролера були змінні, які і зчитуються, і задаються – необхідно зробити декілька таких списків.

Після зроблених налаштувань додайте у знову створений список змінні, які необхідні для обміну даними (змінні для термопари та змінного резистора). В головній програмі PLC_PRG ці змінні обробляються як звичайні змінні.

1.6. Завдання для самостійного виконання

1) Організувати з'єднання двох ПЛК150 ОВЕН для взаємного обміну значеннями температури та змінного резистора за протоколом UDP із застосуванням інтерфейсу *Ethernet* за такою схемою табл. 1.4. У таблиці IP-адреси вузлів визначені лише останнім байтом, тобто адреса мережі така – 172.016.028.*** .

Таблиця 1.4 – Варіанти завдань для обміну даними між ПЛК150

ПЛК-1				ПЛК-2			
№ стенда	IP-адреса ПЛК	№ ПК	IP-адреса ПК	№ стенда	IP-адреса ПЛК	№ ПК	IP-адреса ПК
1	*.*.*.249	S 1	*.*.*.41	3	*.*.*.248	S 3	*.*.*.43
2	*.*.*.241	S 7	*.*.*.47	5	*.*.*.242	S 6	*.*.*.46
4	*.*.*.246	S 5	*.*.*.45	6	*.*.*.247	S 4	*.*.*.44
8	*.*.*.240	S 8	*.*.*.48	9	*.*.*.243	S 9	*.*.*.49
7	*.*.*.244	S 10	*.*.*.50	10	*.*.*.245	S 11	*.*.*.51

2) Організувати з'єднання двох ПЛК150 ОВЕН для управління дискретними виходами за протоколом *UDP* із застосуванням інтерфейсу *Ethernet* за такою схемою табл. 1.5. Для кожної пари ПЛК перший ПЛК буде передавати значення на дискретному вході, а другий – буде зчитувати це значення і включати відповідний дискретний вихід на 10 с. Теж саме повинен робити другий ПЛК по відношенню до першого ПЛК. Логічну комбінацію задіяних входів та виходів показано в табл. 1.5.

Таблиця 1.5 – Варіанти завдань для вибору входів і виходів ПЛК

ПЛК-1		ПЛК-2	
..*.249	(X2 AND X3) OR (X4 AND X5)	*.*.*.248	DO4
..*.241	(X1 OR X3) AND (X2 OR X4)	*.*.*.242	DO3
..*.246	(X1 AND X6) OR X4 OR NOT X5	*.*.*.247	DO2
..*.240	(X4 AND X5) XOR (X3 AND X6)	*.*.*.243	DO1
..*.244	(X2 XOR X3) AND (X4 XOR X6)	*.*.*.245	ВІР

3) Організувати з'єднання двох *SoftPLC* для обміну даними за протоколом *UDP* із застосуванням інтерфейсу *Ethernet* двома парами даних у кожному ПЛК: змінні типу *REAL* і *INT*. Для введення і виведення значень змінних використовуйте візуалізацію з відповідними елементами. Комбінацію варіантів з'єднань ПЛК наведено у табл. 1.6.

Таблиця 1.6 – Варіанти завдань для двох *SOFTPLC*

Номер ПК	IP-адреса ПК	Номер ПК	IP-адреса ПК
S 1	*.*.*.41	S 3	*.*.*.43
S 7	*.*.*.47	S 6	*.*.*.46
S 5	*.*.*.45	S 4	*.*.*.44
S 8	*.*.*.48	S 9	*.*.*.49
S 10	*.*.*.50	S 11	*.*.*.51

1.7. Оформлення результатів проведення заняття і контрольні запитання

ПІДТВЕРДЖЕННЯМ РЕЗУЛЬТАТІВ ТА ФАКТУ
ВИКОНАННЯ РОБОТИ ВВАЖАЄТЬСЯ ІДЕНТИЧНІСТЬ
ЗНАЧЕНЬ ЗМІННИХ
В ПЛК ВИКЛАДАЧА ТА ПЛК СТУДЕНТА

Складіть звіт про виконану лабораторну роботу, де наведіть схему з'єднань, умови обміну даними та лістинг програми в ПЛК.

За підсумками проведення заняття студент повинен дати логічне пояснення отриманих результатів. Крім того, студент повинен відповісти на контрольні запитання, приклади яких наведено нижче:

- 1) У чому полягає принцип обміну даними за протоколом *UDP* по інтерфейсу *Ethernet* між *ПЛК150 OBEH*?
- 2) Перелічіть основні характеристики протоколу *UDP*.
- 3) Дайте характеристику заголовка датаграми в протоколі *UDP*.
- 4) Які бібліотеки необхідні для реалізації обміну даними за протоколом *UDP* між контролерами *ПЛК150 OBEH*?
- 5) Який порядок активації *SoftPLC* та доступу до його даних?
- 6) Якому рівню моделі відповідає протокол *UDP*?
- 7) Яким чином забезпечується багатоканальний обмін за протоколом *UDP*?
- 8) У яких випадках доцільно використовувати протокол *UDP*? Наведіть приклади та надайте необхідні пояснення.

Лабораторна робота 2

СТВОРЕННЯ *WEB*-СЕРВЕРА В *ПЛК150 ОВЕН*

2.1. Мета та час проведення лабораторної роботи

На цю лабораторну роботу відповідно до робочої програми вивчення дисципліни «Комп'ютерні мережі» відводиться 2 акад. год. аудиторного та 2 акад. год. позааудиторного часу (СРС).

Цілі лабораторної роботи:

- закріпити на практиці отримані теоретичні знання про основи використання *WEB*-технологій в комп'ютеризованих системах управління;
- вивчити основні принципи реалізації *WEB*-технологій в *ПЛК150 ОВЕН* за допомогою середовища *CoDeSys V2*.

2.2. Апаратне забезпечення лабораторної роботи

Лабораторна робота виконується на робочих місцях з ПК та підключеними до них *ПЛК150 ОВЕН*, які встановлені на стендах. Для проведення лабораторної роботи використовується штатна локальна мережа кафедри з сервером, мережними пристроями та вузлами (ПК та ПЛК). Всі ПК та ПЛК підключені до локальної мережі кафедри, яка має такі параметри: *TCP/IP*-адреса мережі – 172.016.028.000, маска мережі – 255.255.255.000.

На рис. 2.1 показано фрагмент схеми комунікаційних зв'язків в локальній комп'ютерній мережі, де присутній контролер *ПЛК150 ОВЕН*, в якому запущений *WEB*-сервер, та вузли із запущеним браузером для доступу до даних ПЛК.

2.3. Програмне забезпечення лабораторної роботи

Для створення *WEB*-сервера для *ПЛК150 ОВЕН* застосовується середовище *CoDeSys V2*. Основна складність реалізації проекту з *WEB*-сервером полягає у відсутності операційної системи в контролері. Тому в проекті використані додаткові бібліотеки, які розроблені спеціально

для контролерів *ОВЕН*. Це, наприклад, бібліотека для оброблення символічних даних – *Symlib.lib*.

Доступ до *WEB*-сторінок у ПЛК здійснюється за допомогою програми *WEB*-браузера, наприклад, *Internet Explorer* або іншого.

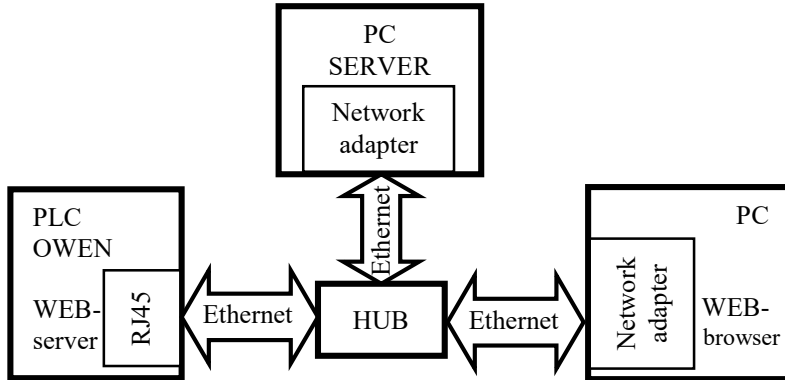


Рисунок 2.1 – Фрагмент побудови локальної мережі

2.4. Короткі теоретичні відомості

WEB-сервер – це програма-сервер, яка приймає *HTTP*-запити (скор. від англ. *HyperText Transfer Protocol* – протокол передавання гіпертексту) від клієнтів, зазвичай *WEB*-браузерів, та відправляє їм *HTTP*-відповіді. Як правило, відповіді приходять до клієнта разом з *HTML*-сторінкою (скор. від англ. *HyperText Markup Language* – мова гіпертекстової розмітки), зображенням, файлом, медіапотоком або іншими даними. *WEB*-сервером називають як програмне забезпечення, що виконує функції сервера, так і безпосередньо комп'ютер (апаратне забезпечення), на якому це програмне забезпечення працює.

WEB-клієнт, тобто *WEB*-браузер, передає серверу запити на отримання ресурсів, позначених *URL*-адресами (скор. від англ. – *Unifical Resource Locator* – уніфікований локатор ресурсу). Ресурси – це *HTML*-сторінки, зображення, файли, медіапотоки або інші дані, які необхідні клієнтові. У відповідь сервер передає клієнтові запитані дані.

Під час розроблення локатор *URL* був створений як система для максимально точного вказівника на місцезнаходження ресурсів у мережі. Локатор мав бути легко розширюваним та використовувати лише обмежений набір *ASCII*-символів (наприклад, «пропуск» ніколи не застосовується в *URL*). У зв'язку з цим існує така традиційна форма запису вказівника *URL*:

`<схема>://<логин>:<пароль>@<хост>:<порт>/
<URL-путь>?<параметри>#<якорь>`,

де `<схема>` – мережний протокол, тобто схема звернення до ресурсу;
`<логин>` – ім'я користувача для доступу до ресурсу;
`<пароль>` – пароль користувача;
`<хост>` – повне ім'я хоста в сервісі *DNS* або *IP*-адреса хоста;
`<порт>` – порт хоста, до якого здійснюється підключення;
`<URL-путь>` – шлях до ресурсу всередині хоста, залежить від протоколу;

`<параметри>` – рядок запитання до сервера з параметрами (метод *GET*), які розділені символом «&»;

`<якорь>` – ідентифікатор «якоря» для посилання на частину документа, що відкривається.

Приклад структури ресурсу *URL* у загальному вигляді занотовується наступним чином:

http://user:password@domain.ru:port/path/some.html?query_string.

Обмін даними відбувається за протоколом *HTTP*. Основою роботи протоколу *HTTP* є технологія «клієнт–сервер». Тобто передбачається існування споживачів (клієнтів), які ініціюють з'єднання і посилають запит, та постачальників (серверів), які чекають з'єднання для отримання запиту, проводять необхідні дії і повертають назад повідомлення з результатом. Цей протокол використовується для отримання інформації із *WEB*-сайтів у мережі *Internet*. Особливістю протоколу *HTTP* є можливість вказати в запиті/відповіді спосіб подання одного і того ж ресурсу за різними параметрами: форматом, кодуванням, мовою тощо (зокрема

для цього використовується *HTTP*-заголовок). Саме завдяки можливості вказівки способу кодування повідомлення клієнт і сервер можуть обмінюватися двійковими даними, хоча цей протокол є текстовим. Цей протокол належить до протоколів прикладного рівня еталонної моделі взаємозв'язку відкритих систем.

Кожне *HTTP*-повідомлення складається з трьох частин, які передаються у наперед визначеному порядку:

- стартовий рядок (від англ. *Starting line*) – визначає тип повідомлення;
- заголовки (від англ. *Headers*) – характеризують тіло повідомлення, параметри передачі та інші відомості;
- тіло повідомлення (від англ. *Message Body*) – безпосередньо дані повідомлення. Тіло обов'язково повинно відділятися від заголовків пустим рядком.

Заголовки та тіло повідомлення можуть бути відсутніми, але стартовий рядок є обов'язковим елементом, оскільки вказує на тип запити/відповіді.

2.5. Послідовність проведення заняття

Лабораторна робота виконується комплексно. Її виконання складається з підготовки контролера для його роботи в мережі *Ethernet* та завантаження у нього *WEB*-сторінки за допомогою браузера в середовищі *CoDeSys V2*.

Створить проект в середовищі *CoDeSys*. Як таргет-файл для проекту оберіть модель *ПЛК150-I-L*. У лабораторній роботі основна увага приділена вивченню *WEB*-технологій та принципам налаштування *WEB*-сервера в ПЛК, тому надається готовий проект з *WEB*-сервером. Отже, скопіюйте файл з проектом з сервера мережі та завантажте його в середовище *CoDeSys V2* для подальшого завантаження у ПЛК. Далі встановіть мінімальний час циклу контролера рівним 10 мс.

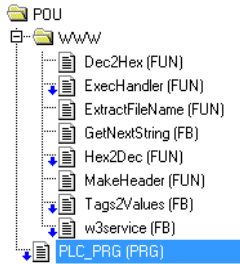


Рисунок 2.2 – Структура проекту

Для запуску *WEB*-сервера в дереві проекту є функціональний блок з ім'ям *w3service* (FB). Зазначимо, що в основній програмі викликається три екземпляри вказаного функціонального блока, тобто можливе підключення до *WEB*-сервера не більше трьох клієнтів.

Скомпілюйте та завантажте проект у ПЛК.

Далі необхідно завантажити у ПЛК *WEB*-сторінки, які розроблені заздалегідь. Для цього необхідно перейти до вкладки *Ресурси* та запустити *ПЛК-браузер* для зв'язку з контролером. За допомогою браузера завантажте у ПЛК контент сайту, який буде відкриватися після запиту у *WEB*-браузері клієнта. Контент сайту складається з двох файлів сторінок

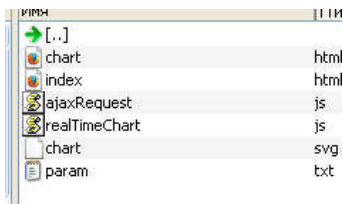
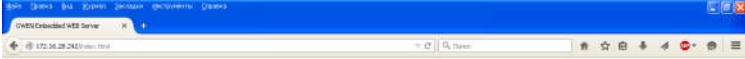


Рисунок 2.3 – Контент сайту

(для зміни *IP*-адреси і відображення графіків змінних) та аплетів. На рис. 2.3 наведено склад контенту сайту, який завантажується у ПЛК. До складу сайту входять сторінки для зміни *IP*-адреси та відображення графіка поведінки змінних (стану дискретних

входів, температури, яка вимірюється термопарою та термометром опору, а також значення змінного резистора). На рис. 2.4 та 2.5 показано зображення відповідних сторінок. Спостерігайте за поведінкою змінних всередині ПЛК за допомогою *WEB*-браузера.



Изменение сетевых настроек

IP Адрес . . .

Подсеть . . .

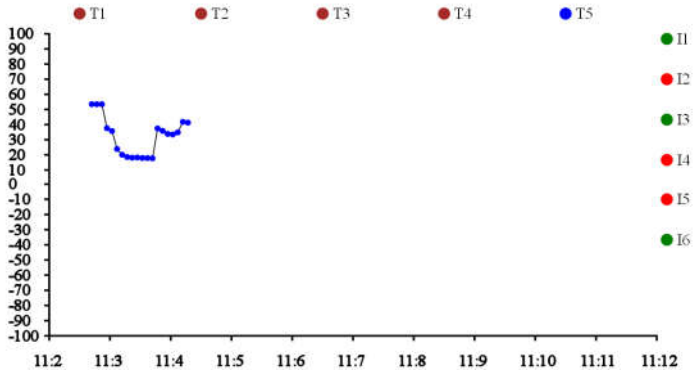
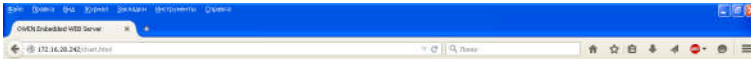
Шлюз . . .

MAC адрес 6A : 77 : 00 : 00 : 85 : 54

[Пример графика](#)

[OWEN\(c\) 2015](#)

Рисунок 2.4 – Сторінка WEB-сервера з мережними параметрами



[OWEN\(c\) 2015](#)

Рисунок 2.5 – Сторінка WEB-сервера зі змінними процесу

2.6. Оформлення результатів проведення заняття і контрольні запитання

ПІДТВЕРДЖЕННЯМ РЕЗУЛЬТАТІВ ТА ФАКТУ ВИКОНАННЯ РОБОТИ ВВАЖАЄТЬСЯ МОЖЛИВІСТЬ ДОСТУПУ ДО ДАНИХ В ПЛК З БУДЬ-ЯКОГО КОМП'ЮТЕРА

Складіть звіт про виконану лабораторну роботу, де наведіть схему локальної мережі, умови обміну даними та лістинг контенту сайту в ПЛК.

За підсумками проведення заняття студент повинен дати логічне пояснення отриманих результатів. Крім того, студент повинен відповісти на контрольні запитання, приклади яких наведено нижче:

1) У чому полягає принцип доступу до даних в ПЛК за протоколом *HTTP*?

2) Перелічіть склад запиту за протоколом *HTTP*.

3) Опишіть приклад структури ресурсу *URL* у загальному вигляді. Наведіть приклад.

4) Які бібліотеки необхідні для реалізації *WEB*-сервера в контролері *ПЛК150 OVEN*?

Лабораторна робота 3

НАЛАШТУВАННЯ WEB-СЕРВЕРА В ПЛК FC34 FESTO

3.1. Мета та час проведення лабораторної роботи

На цю лабораторну роботу відповідно до робочої програми вивчення дисципліни «Комп'ютерні мережі» відводиться 2 акад. год. аудиторного та 2 акад. год. позааудиторного часу (СРС).

Цілі лабораторної роботи:

- закріпити на практиці отримані теоретичні знання про основи використання *WEB*-технологій;
- вивчити основні принципи реалізації *WEB*-сервера в ПЛК *FC34 FESTO*;
- отримати навички роботи з контролером за допомогою термінала і командного інтерпретатора *через WEB*-сервер.

3.2. Апаратне та програмне забезпечення лабораторної роботи

Лабораторна робота виконується на робочих місцях з ПК та підключеними до них ПЛК *FC34 FESTO*, які встановлені на стендах.

Для проведення лабораторної роботи використовується штатна локальна мережа кафедри з сервером, мережними пристроями та вузлами (ПК). Всі ПК та ПЛК підключені до локальної мережі кафедри, яка має такі параметри: *TCP/IP*-адреса мережі – 172.016.028.000, маска мережі – 255.255.255.000. У табл. 3.1 наведено *IP*-адреси контролерів *FC34 FESTO* та номери комп'ютерів, з яких проводиться підключення за допомогою *WEB*-браузера.

Таблиця 3.1 – Адреси контролерів *FC34 FESTO*

Номер ПК	<i>IP</i> -адреса ПЛК
S1, S2, S3, S4	172.016.028.230
S5, S6, S7, S8	172.016.028.231
S9, S10, S11	172.016.028.232

На рис. 3.1 показано фрагмент схеми комунікаційних зв'язків у локальній комп'ютерній мережі, де присутні ПЛК *FC34 FESTO*, які вико-

нують функцію *WEB*-сервера та вузли із запущеним браузером для доступу до даних ПЛК.

Для створення *WEB*-сервера в *FC34 FESTO* застосовується середовище *FST4.0*. Доступ до даних *WEB*-сервера реалізовано за допомогою звичайного браузера. Редагування контенту сторінки виконується за допомогою програми *Блокнот*, яка входить до стандартного набору програм операційної системи *Windows*.

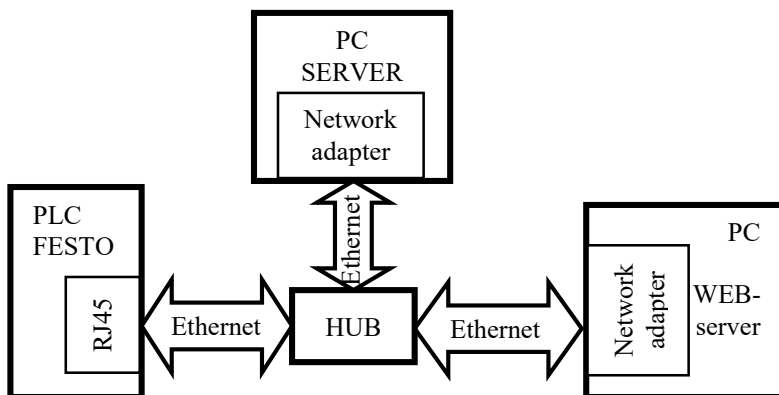


Рисунок 3.1 – Фрагмент побудови локальної мережі

3.3. Короткі теоретичні відомості

Середовище *FST4.10* призначено для створення проектів з програмами для ПЛК *FC34 FESTO*. Для реалізації *WEB*-сервера в контролері *FC34* у середовищі є додаткові програмні драйвери з переліку бібліотек. Це драйвери з ім'ям *TCPIPFEC* та *WEB_SRVR*. Ці драйвери підключаються до проекту за допомогою менеджера проекту у вкладці *Driver Configuration*.

Додавання драйвера *TCPIPFEC* дає контролеру можливість функціонувати в локальній мережі. Після додавання драйвера потрібно його налаштувати, як це показано на рис. 3.2, тобто ввести *IP*-адресу та маску. Драйвер підтримує такі функції:

- власне конфігурування;
- доступ до модуля;
- підтримання протоколу *BOOTP/DHCP*.

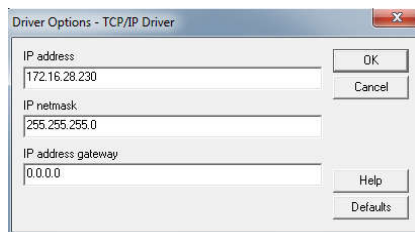


Рисунок 3.2 – Параметри налаштування драйвера *TCPIPFE*

У результаті конфігурування драйвера контролер *FC34 FESTO* буде працювати за протоколом *TCP/IP* за адресою, яку призначить *DHCP*-сервер (скор. від англ. Dynamic Host Configuration Protocol – протокол динамічного налаштування вузла), або за адресою, яку призначить користувач. Протокол *DHCP* – це мережний протокол, який дозволяє комп'ютерам та іншим хостам (ПЛК) автоматично отримувати *IP*-адресу та інші параметри, необхідні для роботи в мережі *TCP/IP*. Цей протокол працює за моделлю зв'язку «клієнт–сервер». Для отримання адреси за протоколом *DHCP* у рядок *IP-address* введіть рядок «*0.0.0.0*» (див. рис. 3.2).

Драйвер *WEB_SRVR* для реалізації *WEB*-сервера в контролері розширює його можливості за рахунок додавання функцій доступу до нього з використанням *WEB*-технологій з боку браузера. При цьому дані всередині ПЛК повинні бути сформовані у форматі протоколу *HTML*. Таким чином, запити з боку браузера завантажуються у вигляді *WEB*-сторінок. На рис. 3.3 показано параметри налаштування *WEB*-сервера. В полі *WebSrvr rootdirectory* потрібно ввести місце розміщення *WEB*-сторінок на *flash*-диску в ПЛК. У цьому випадку це диск «*B*», а шлях доступу до сторінок – *B:\WEB* . У цю директорію завантажуються контент сайту, який заздалегідь створений.

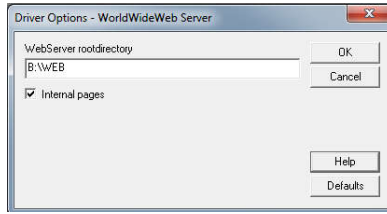


Рисунок 3.3 – Параметри налаштування драйвера *WEB_SRVR*

3.4. Послідовність проведення заняття

Лабораторна робота виконується комплексно. Її виконання складається з підготовки контролера для його роботи в мережі *Ethernet* і завантаження у нього проекту та *WEB*-сторінки за допомогою браузера у середовищі *FST4.10*. Для завантаження у ПЛК пропонується готовий проект з програмою, яка зображена на рис. 3.4.

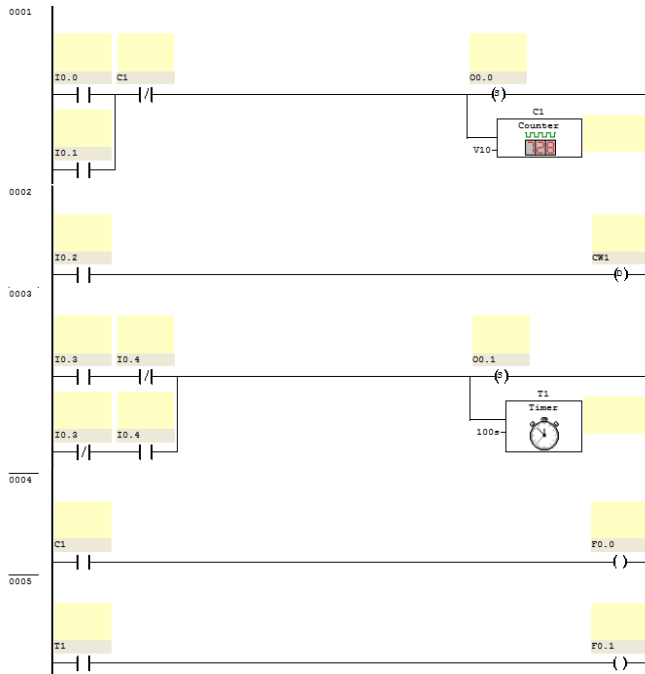


Рисунок 3.4 – Програма управління у проекті для *FC34 FESTO*

У цій програмі за сигналами від дискретних датчиків запускаються таймер та інкрементуючий лічильник. У табл. 3.2 наведено дані про операнди, їхні абсолютні адреси та символічні імена.

Таблиця 3.2 – Адреси та типи операндів у проєкті для *FC34 FESTO*

Символьное имя	Абсолютный адрес	Тип операнда
in1	I0.00	BIT
out1	O0.00	BIT
in2	I0.01	BIT
in3	I0.02	BIT
cou1	C1	BIT
tim1	T1	BIT
flag_coun	F0.00	BIT
flag_tim	F0.01	BIT
tim1_car	TW1	WORD(Bin)
tim_net	FW1	WORD(Bin)
cou1_car	CW1	WORD(Bin)
count_net	FW2	WORD(Bin)

Отже, запустіть середовище *FST4.10* та завантажте наданий проєкт з програмою (див. рис. 3.4) в ПЛК. Також за допомогою браузера завантажте *WEB*-сторінки в енергонезалежну пам'ять ПЛК. Для цього використайте в меню *Online* команду *File Transfer*.

Запустіть *Internet*-браузер та в рядок адреси введіть *IP*-адресу контролера. Відкриється вікно зі стартовою сторінкою, яке зображено на рис. 3.5. На цій сторінці наведено інтерактивний курс *WEB*-дизайну та рядок переходу на онлайн-сторінку контролера (*To FST Online index page*). На рис. 3.6 показано склад цієї сторінки. Ця сторінка надає можливість переглянути стан ПЛК та статус програми, стан операндів і має рядок переходу до термінального режиму зв'язку ПЛК за допомогою програми *Terminal Emulator* і рядок повернення до головної сторінки *WEB*-сервера (див. рис. 3.5). У термінальному режимі за допомогою ко-

мандного інтерпретатора є можливість впливати на стан операндів та статус програми, а також управляти режимами роботи ПЛК.



Рисунок 3.5 – Головна сторінка *WEB*-сервера у ПЛК *FC34 FESTO*



Рисунок 3.6 – Сторінка *WEB*-сервера у ПЛК *FC34 FESTO*

Наприклад, у термінальному режимі запускайте таймер або впливайте на лічильник шляхом вмикання відповідного дискретного входу (див. лістинг програми на рис. 3.4)

3.4.1. Особливості роботи командного інтерпретатора

Командний інтерпретатор (в подальшому – КІ) дає можливість управляти роботою ПЛК зовні, наприклад, за допомогою *WEB*-сервера. Це дає певні можливості для організації зв'язку між ПК та ПЛК. КІ може відображати, змінювати операнди та управляти програмами. Після активації термінального режиму з'явиться вікно програми *HyperTerminal*, яка вбудована у стандартний набір комунікаційних утиліт ОС *Windows XP/2000/ME*, яке показано на рис. 3.7.

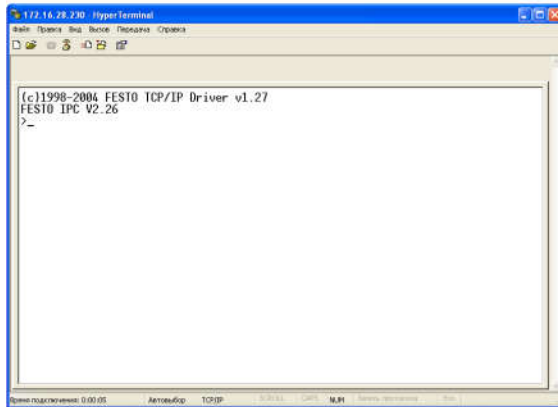


Рисунок 3.7 – Вікно запусненої програми *HyperTerminal* для зв'язку з *WEB*-сервером у ПЛК *FC34 FESTO*

Після підключення до ПЛК з'явиться повідомлення про тип з'єднання (*TCP/IP*) та версію «прошивки» ПЛК (*FESTO IPC V2.26*), а також із запрошенням (>).

Розглянемо основні команди КІ. Насамперед необхідно розглянути синтаксис команд, який має певний формат:

- символ команди;

- параметр (символ або цифра, залежно від типу параметра);
- значення параметра (присутній не завжди).

Отже, команда запиту буде виглядати так:

вхідний запит – <Command letter>[<Parameter>] [=Value];
повідомлення КІ – залежно від вхідного запиту.

До складу команд входять символи обох регістрів, введення яких обов’язково повинно закінчитись комбінацією <CR>. У табл. 3.3 та 3.4 наведено основні символи команд та їх короткий опис з роз’ясненнями скорочень. Дозволені значення параметра залежать від типу відповідного параметра та/або операндів. На правильні та помилкові команди КІ дає відповідні повідомлення. У табл. 3.5 наведені можливі повідомлення на запити КІ. На помилкові команди з’являється відповідь «ACCESS ERROR» або її скорочення «ERR». Одночасно подається звуковий сигнал.

Таблиця 3.3 – Основні символи команд КІ

<i>Символи команд</i>	<i>Короткий опис</i>	
B	BREAK	Переривання виконання програми
DC4 (Ctrl T)	LOGIN	Реєстрування, вхід у систему
D	DISPLAY	Відобразити операнди
LC	PASSWORD	Встановлення/зміна пароля
LX	PASSWORD	Захист паролем програм в ПЛК
M	MODIFY	Модифікування операндів
R	RUN	Запуск виконання програми
S	STOP	Припинення виконання програми
X	LOGOUT	Вихід з системи
Y	INIT	Видалення пам’яті користувача

Розглянемо докладніше відповіді ПЛК на різні запити КІ.

Для відображення статусу операндів та стану програми використовують команду «D». На прикладі для відображення стану виходу A0.1 це виглядає так:

Запит КІ: >DA0.1
Відповідь ПЛК: >DA0.1=0.

Це означає таке: символи «D», «A0.1» відправляються до ПЛК, а він їх повертає та дає відповідь «=0», тобто операнд скинуто.

Таблиця 3.4 – Призначення параметрів основних команд КІ

<i>Параметр</i>	<i>Призначення</i>	<i>Скорочення</i>
A [<YN>] <WN><BN>	Вихідний біт	Замість скорочення, вводять дійсне значення, яке залежить від типу операнда <BN>: Номер біта <MN>: Номер модуля <DN>: Номер привода <PN>: Номер програми <RN>: Номер регістра <TN>: Номер таймера <WN>: Номер слова <CN>: Номер лічильника <YN>: Номер станції
AW [<YN>] <WN>	Вихідне слово	
B<BN>	Програмний модуль	
BF<BN>	Функціональний модуль	
D	Формат відображення	
E [<YN>] <WN> <BN>	Вхідний біт	
EW [<YN>] <WN>	Вхідне слово	
F	Слово помилки	
M<WN><BN>	«Прапорець»-біт	
MW<WN>	«Прапорець»-слово	
O<WN>	Глобальні ФБ FU0 ... FU31 та FU39 ... FU255	
O<PN><WN>	Локальні ФБ FU32... FU38	
P<PN>	Статус програми	
R<RN>	Індекс	
S<PN>	«Прапорець» ініціювання	
T<TN>	Імпульсний таймер	
TA<TN>	Перемикач вимкнення таймера затримки	
TE<TN>	Перемикач ввімкнення таймера затримки	
TV<TN>	Попереднє налаштування таймера	
TW<TN>	Слово таймера	
V	Швидкість (у бодах)	
Z<CN>	Лічильники	
ZV<CN>	Попереднє налаштування лічильника	
ZW<CN>	Слово лічильника	

Отже, в КІ доступні такі команди для відображення:

- DA [<YN>] <WN><BN> для вихідного біта;
- DAW [<YN>] <WN> для вихідного слова;
- DB<BN> для програмного модуля;
- DBF<BN> для функціонального модуля.

Відповідь ПЛК має такий синтаксис:

=<Type>,0, <Status>, <Step>, де:

- перший параметр вказує мову: *STL*=0, *LDR/FUP*=1 або *C*=2;
- другий параметр – це область пам'яті, завжди становить «0»;
- третій параметр відображує статус вибраної програми: «0» – програма неактивна; «2» – програма активна, але зупинена; «3» – програма активна;
- останній параметр – це поточний номер кроку в межах модуля.

Для відображення формату багаторозрядних операндів використовують команду D з такими ідентифікаторами:

- DE [<YN> .] <WN> . <BN> для вхідного біту;
- DEW [<YN> .] <WN> для вхідного слова.

Відповідь ПЛК буде такою:

- «=D», якщо операнд має десятковий формат без знака;
- «=S», якщо операнд має десятковий формат зі знаком;
- «=H», якщо операнд має *HEX*-формат.

Таблиця 3.5 – Відповіді на запити KI

Команди KI	Відповіді KI
Команда	<Command>"\r"
Відповідь на правильну команду	<Command><Response>"\r\n>\21"
Відповідь на помилкову команду	<Command>"\b\r\nACCESS ERROR\r\n>\21"

Можливе також відображення слова помилки за командою DF та статусу помилки ПЛК за командою DE». У випадку відсутності помилки ПЛК відповідає кодом «=0, 0, 0». Також можливі такі відповіді:

- для загальних помилок, це «=<Error number>,<Program number>,<Step number>»;
- для помилок діагностики, це «=<42>,<CPX error number>,<CPX module number>»;
- для помилок введення/виведення, це «=<Error no.>,<255>,<No. of input or output word>».

Для відображення глобальних (*FU0...FU31* та *FU39...FU255*) і локальних (*FU32...FU38*) блоків у КІ є відповідні команди `DO<WN>` та `DO<PN>.<WN>`.

Для відображення статусу інших операндів в КІ є такі команди:

- для «прапорця»-біта: `DM<WN>.<BN>`;
- для «прапорця»-слова: `DMW<WN>`;
- для регістру: `DR<RN>`;
- для «прапорця» стану програми ініціювання: `DS<PN>`;
- для імпульсного таймера: `DT<TN>`;
- для таймера із затримкою вимикання: `DTA<TN>`;
- для таймера із затримкою вмикання: `DTE<TN>`;
- для лічильника: `DZ<CN>`;
- для попередніх налаштувань таймера: `DTV<TN>`;
- для попередніх налаштувань лічильника: `DZV<CN>`;
- для слова таймера: `DTW<TN>`;
- для слова лічильника: `DZW<CN>`.

КІ дає можливість за допомогою команди «DV» запитати поточну швидкість передавання даних у бодах. ПЛК дає відповідь з можливими значеннями: «=1200», «=2400», «=4800», «=9600», «=19200», «=38400» або «=56000».

Під час запитів з боку КІ є можливість отримувати не тільки значення операндів та їх статус. За допомогою команди «M» можливе змінювання статусу операндів та їх примусове призначення. Щоб безпосередньо змінити операнди, необхідно ввести необхідне значення і підтвердити нове значення введенням `<CR>`. Наприклад, це буде виглядати так:

- Вхідний запит: >MAW1=255;
- Відповідь ПЛК: >MAW1=255.

Або, якщо ввести лише команду «M» та потрібний операнд, а потім <CR>:

- Вхідний запит: >MAW1;
- Відповідь ПЛК: >MAW1=255:.

В останньому випадку КІ повідомляє поточне значення операнда. Після двокрапки можна ввести нове значення та підтвердити його натисканням <CR>. Значення можна вводити в десятковому, десятковому зі знаком або *HEX* форматі. Зміна формату відображення здійснюється за командою MD={D | S | H}, де відповідно десятковий без знака «=D», десятковий зі знаком «=S» або *HEX* «=H».

Формат команд модифікування буде таким:

а) для вихідного:

- біта: MA[<YN>].<WN>.<BN>={0 | 1};
- слова: MAW[<YN>].<WN>=<Value>;

б) для вхідного:

- біта: ME[<YN>].<WN>.<BN>={0 | 1};
- слова: MEW[<YN>].<WN>=<Value>;

в) для вхідного:

- «прапорця»-біта: MM<WN>.<BN>={0 | 1};
- «прапорця»-слова: MFW<WN>=<Value>.

Для зміни слова похибки існує окрема команда у форматі MF=<Value>. Значення «0» видаляє поточну помилку. Кожне інше значення генерує відповідну динамічну помилку.

Команда MO<WN>=<Value> змінює глобальні функціональні блоки від *FU0* до *FU31* та від *FU39* до *FU255*. У випадку зміни локального функціонального блока (від *FU32* до *FU38*) використовують команду MO<PN>.<WN>=<Value>.

Існують також окремі команди по відношенню до функціональних блоків для кожної програми:

- зміна значення регістра: MR<RN>=<Value>;

- зміна статусу імпульсного таймера (старт/стоп): $MT<TN>=\{0 | 1\}$;
- вимкнення перемикача таймера із затримкою: $MTA<TN>=\{0 | 1\}$;
- ввімкнення перемикача таймера із затримкою: $MTE<TN>=\{0 | 1\}$;
- зміна стану лічильника: $MZ<CN>=\{0 | 1\}$;
- зміна попередніх налаштувань таймера: $MTV<TN>=<Value>$;
- зміна попередніх налаштувань лічильника: $MZV<CN>=<Value>$;
- зміна слова таймера: $MTW<TN>=<Value>$;
- зміна слова лічильника: $MZW<CN>=<Value>$.

Зміна швидкості передавання даних здійснюється за командою $MV=<Baudrate>$. Швидкість передавання даних може бути встановлена командами: « $MV=1200$ », « $MV=2400$ », « $MV=4800$ », « $MV=9600$ », « $MV=19200$ », « $MV=38400$ » або « $MV=56000$ ». Значення може бути скорочено до 2-х знаків, наприклад, « $MV=96$ ».

Запуск програми здійснюється командою «R». Проте команди «B» або «BP<PN>» переривають всі запущені або одну певну програму. Причому, якщо в параметрах налаштування ПЛК вибрана опція «Reset programs» («Скидання програм»), то знову запускається та виконується програма P0. Якщо ця опція не вибрана, то всі зупинені (перервані) програми продовжують виконуватись.

Команда $RB<Number> [, <FU32> [, <FU33> [, \dots [, <FU37> [, <FU38>]]]]]$ вибирає завантажений ПМ (один, що міститься у файлі проекту). При цьому команда використовує локальні ФБ з програми P63, яку потрібно резервувати для цієї мети. Команда «RF» з подібним синтаксисом вибирає завантажений ФМ. При цьому повинні бути вказані параметри виклику. Якщо ніякий параметр не вказаний, то використовується його останнє значення. У цьому випадку відповідь буде такою: $=<FU32> , <FU33> , <FU34> , <FU35> , <FU36> , <FU37> , <FU38>$.

Наступний приклад ілюструє наведене. Нехай у модулі СМР 7 параметри ФБ дорівнюють таким значенням: $FU32=14$, $FU33$ =останнє значення, $FU34=9$. Результат запиту буде таким: $FU32=4712$, $FU33=103$,

від *FU34* до *FU38* дорівнює 0. Тоді відповідь ПЛК буде такою: RB7,14,,9=4712,103,0,0,0,0,0.

Запуск або продовження виконання програми здійснюється за командою RP<PN>.

Насамкінець, зупинка певної або всіх програм, що виконуються, здійснюється за командою SP<PN> або S.

Всі цифрові входи та виходи можуть бути примусово призначені до «0» або «1». Проте примусове призначення входів та виходів доступне не на всіх типах ПЛК. Якщо вхід або вихід має примусове значення, це може бути виявлено програмою або КІ. Вміст таблиці примусових призначень не зберігається. Він автоматично видаляється командою Y або при завантаженні проекту. Для встановлення входів/виходів доступна команда YF, яка видаляє таблицю примусових призначень.

Для відображення *примусового вихідного біта або слова* використовують команди DAF<WN>.<BN> та DAWF<WN>. Відповідь для запиту *біта* буде такою: =0:, якщо призначений «0», =1:, якщо призначена «1», =N:, якщо не призначений. У випадку *слова* це буде виглядати так «=xxxxxxxxxxxxxxxxxxxx». Причому відповіді будуть подібними наведеному раніше.

Для відображення *примусового вхідного біта або слова* синтаксис відповідей буде подібним наведеному вище, але команди будуть іншими, тобто:

- для бітів: DEF<WN>.<BN>;
- для слова: DEWF<WN>.

Введення статусу *вихідного біта* в таблицю примусових призначень здійснюється за допомогою команди MAF<WN>.<BN>={0 | 1 | N}, для *вихідного слова* – це команда MAWF<WN>={Value | N}. Те ж саме стосується команд введення значень *вихідних бітів* та *слів*: MEF<WN>.<BN>={0 | 1 | N} та MEWF<WN>={Value | N}. Відповіді будуть подібними наведеному вище поясненню.

Команда ініціювання *Y* видаляє все дані проекту з пам'яті ПЛК з можливістю поновлення, а якщо з додаванням знака «!», тобто – *Y!*, видаляє все дані безповоротно.

Наступні команди КІ дають можливість встановлювати, замінити або видаляти пароль захисту в діалоговому режимі, а також включати або виключати захисний пароль для ПЛК. Пароль може складатись від 3 до 20 знаків у форматі *ASCII*. Розділові та спеціальні знаки, наприклад, коми, інтервали, табулятори тощо не дозволяються. Формат встановлення/зміни пароля має такий вигляд: *LC*<old>, <new> (для заміни пароля) або *LC*, <new> (для встановлення пароля).

Коли вводиться новий пароль, то обов'язково потрібно ввести старий пароль. Наприклад, введення *LC**TEST*, *FEC* змінює пароль з «*TEST*» на «*FEC*». Якщо раніше не було ніякого пароля, то старий пароль не потрібно вводити. Проте повинна бути обов'язково кома між «*LC*» та «*FEC*». Для початку або кінця сеансу захисту паролем використовується команда *LX*. Наприклад, якщо встановлений пароль «*FEC*», то:

- команда *LXFEC* деактивує захист паролем;
- команда *LX* або *LX* з неправильним паролем активує захист паролем.

Середовище *FST* дає можливість драйверам одержувати певні команди. Ці команди починаються зі знака «!» та номера драйвера <DN>, потім слідує безпосередньо команда, наприклад: !<DN><Command>.

Майже всі команди КІ можуть бути відправлені до ПЛК суміщеними. Це означає, наприклад, що КІ може послідовно обробити команди «D» та «M» та згрупувати відповіді. При цьому команди повинні бути відокремлені крапкою з комою. Наприклад, команди для запуску програми «P0» та для запиту її статусу мають вигляд: >RP0>DP0=0,0,3,2,0,0>.

Отже, послідовність команд буде такою:

- вхідний запит: >RP0;DP0
- відповідь ПЛК: >RP0;DP0=0,0,3,2,0,0.

Для відображення статусу операндів R0, FW16 та I0.3 разові команди мають вигляд: >DR0=432>DMW16=0>DE0.3=1>. Проте команда послідовність буде такою:

- вхідний запит: >DR0,MW16,E0.3
- відповідь ПЛК: >DR0,MW16,E0.3=432=0=1.

Таким чином, було розглянуто основні команди, що використовуються в діалоговому режимі для здійснення обміну даними між ПЛК та ПК за допомогою термінального режиму, який запущений всередині *WEB*-сервера.

3.5. Оформлення результатів проведення заняття і контрольні запитання

ПІДТВЕРДЖЕННЯМ РЕЗУЛЬТАТІВ ТА ФАКТУ ВИКОНАННЯ РОБОТИ ВВАЖАЄТЬСЯ МОЖЛИВІСТЬ ДОСТУПУ ДО WEB-СТОРІНКИ В ПЛК З БУДЬ-ЯКОГО КОМП'ЮТЕРА

Складіть звіт про виконану лабораторну роботу, де наведіть схему з'єднань, умови обміну даними та лістинг програми в ПЛК.

За підсумками проведення заняття студент повинен дати логічне пояснення отриманих результатів. Крім того, студент повинен відповісти на контрольні запитання, приклади яких наведено нижче:

- 1) Наведіть порядок створення *WEB*-сервера в ПЛК *FC34 FESTO*?
- 2) Яким чином здійснюється завантаження проекту в ПЛК *FC34 FESTO*?
- 3) Яким чином здійснюється завантаження *WEB*-сторінок в ПЛК *FC34 FESTO*?
- 4) Яким чином працює командний інтерпретатор в термінальному режиму всередині *WEB*-сервера в ПЛК *FC34 FESTO*? Наведіть приклад.

Лабораторна робота 4
НАЛАШТУВАННЯ *WEB*-ВІЗУАЛІЗАЦІЇ
У *SoftPLC* ТИПУ *CoDeSys SP PLCWinNT*

4.1. Мета та час проведення лабораторної роботи

На цю лабораторну роботу відповідно до робочої програми вивчення дисципліни «Комп'ютерні мережі» відводиться 4 акад. год. аудиторного та 4 акад. год. позааудиторного часу (СРС).

Цілі лабораторної роботи:

- закріпити на практиці отримані теоретичні знання про основи використання *WEB*-технологій;
- вивчити основні принципи реалізації *WEB*-візуалізації в *ПЛК150 ОВЕН*.

4.2. Апаратне та програмне забезпечення лабораторної роботи

Для проведення лабораторної роботи використовується штатна локальна мережа кафедри з сервером, мережними пристроями та вузлами (ПК). Всі ПК та ПЛК підключені до локальної мережі кафедри, яка має такі параметри: *TCP/IP*-адреса мережі – 172.016.028.000, маска мережі – 255.255.255.000.

На рис. 4.1 наведено фрагмент схеми комунікаційних зв'язків у локальній комп'ютерній мережі, де є комп'ютер із запущеним програмним модулем *SP PLCWinNT* та програмою-сервером для реалізації *WEB*-візуалізації, які входять до складу комплексу *CoDeSys V2*. Інший комп'ютер із запущеним браузером для доступу до даних *SoftPLC* виконує функцію клієнта.

4.3. Короткі теоретичні відомості

Поняття «*WEB*-візуалізація» є програмною технологією, яка дозволяє спостерігати та управляти *CoDeSys*-візуалізацією за допомогою *WEB*-браузера на будь-якій апаратній платформі.

Середовище *CoDeSys* може формувати описи об'єктів візуалізації проекту у форматі *XML* та завантажувати їх в контролер. *WEB*-сервер

обробляє дані контролера також у форматі *XML* та створює постійно відновлювану візуалізацію. Таким чином, вона відобразатиметься у *WEB*-браузері на будь-якому підключеному через *Internet* комп'ютері незалежно від платформи (наприклад, з метою дистанційного управління технологічним об'єктом).

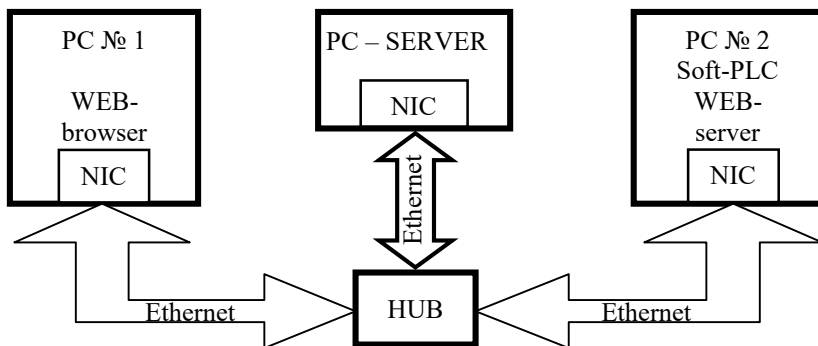


Рисунок 4.1 – Фрагмент побудови локальної мережі

Технологія *WEB*-візуалізації у середовищі *CoDeSys* не завжди підтримана в контролерах. Так, наприклад, у ПЛК *OBEH 100*-ї серії *WEB*-візуалізація неможлива. Тому в лабораторній роботі в проєкті для створення *WEB*-візуалізації використано програмний контролер типу *SP PLCWinNT*, який був описаний в лабораторній роботі 1.

Додаткове програмне забезпечення для запуску *WEB*-візуалізації для контролеру типу *SP PLCWinNT* знаходиться в директорії *Visu* зі шляхом доступу (за умовчанням це місце розміщення середовища *CoDeSys*) *C:\Program Files\3S Software\CoDeSys V2.3\Visu*. У цій папці є файл *webvisu.jar*, який є контейнером *Java*-файлів (у файловому браузері *Total Commander* після натиснення клавіш *Ctrl+PgDn* на цьому файлі можна побачити вміст контейнера). Цей файл з рештою вмісту папки *Visu* автоматично копіюється в контролер. *WEB*-сервер за запитом клієнта видасть сторінку *webvisu.htm*, яка завантажить *Java*-файли,

і в результаті відобразить візуалізацію в браузері клієнта. Логічно, що для цього на ПК-клієнті знадобиться встановлена *Java*, останню версію якої можна завантажити тут – <http://java.sun.com/>.

4.4. Послідовність проведення заняття

Лабораторна робота виконується комплексно. Її виконання складається з підготовки контролера для роботи в локальній мережі та налаштування локального *WEB*-сервера для відпрацювання *WEB*-візуалізації за запитами з інших комп'ютерів за допомогою *WEB*-браузера для спостереження за змінними проекту. Студенти повинні підключитися до *SoftPLC* на своєму комп'ютері та налаштувати *WEB*-сервер, щоб викладач мав можливість спостерігати за змінними проекту.

Підготовка *WEB*- візуалізації в *CoDeSys*:

1) Створіть проект та візуалізацію до нього в *CoDeSys* звичайним способом. Якщо ви хочете встановити певний об'єкт візуалізації як стартовий, назвіть його «*PLC_VISU*». Він буде автоматично завантажений, при виклику візуалізації через Інтернет.

2) Деактивуйте опцію «*WEB*-візуалізація» в діалозі «*Об'єкт*»-«*Свойства*» для об'єктів візуалізації, які не повинні входити в *WEB*-візуалізацію.

3) При необхідності модифікуйте базову сторінку *WebVisu.htm*. Наприклад, ви можете перевизначити стартовий об'єкт *PLC_VISU*.

4) Врахуйте можливість створювати елементи візуалізації для перемикання сервера між декількома контролерами.

5) Виконайте команди «*Проект*»-«*Очистить все*», далі «*Проект*»-«*Компилировать*». Перед завантаженням проекту зверніть увагу на опцію «*Не загружать файлы визуализации*» в налаштуваннях цільової платформи.

6) Встановіть з'єднання з цільовою системою за допомогою команди «*Подключение*» і запустіть проект на виконання.

7) Виконайте конфігурацію та запуск *WEB*-сервера.

4.4.1. Розроблення та запуск проекту на платформі SP PLCWinNT

Отже, запустіть середовище *CoDeSys*. Оберіть потрібну платформу контролера. Це буде програмний ПЛК типу *SP PLCWinNT*, який запускається локально на ПК зі встановленим програмним забезпеченням (див. лабораторну роботу 1, рис. 1.2). Далі виберіть мову реалізації проекту – це буде мова *SFC*. У налаштуваннях таргет-файла виберіть вкладку *Візуалізація* та поставте позначку в полі навпроти рядка з написом *Web-візуалізація*. Це дозволить зробити завантаження потрібних файлів *WEB*-візуалізації в контролер в процесі компілювання.

Для імітування динамічних процесів всередині ПЛК у проекті створіть програму. Ця програма імітує рух по периметру графічного об'єкта «квадрат» шляхом розрахунку двох координат – *X* та *Y*. На рис. 4.2 наведено програму *PLC_PRG*, яка написана на мові *SFC*. Кожному кроку (*Init*, *Step2*, *Step3*, *Step4*) основної програми відповідають дії, які написані на мові *ST* (показані на рис. 4.2 праворуч).

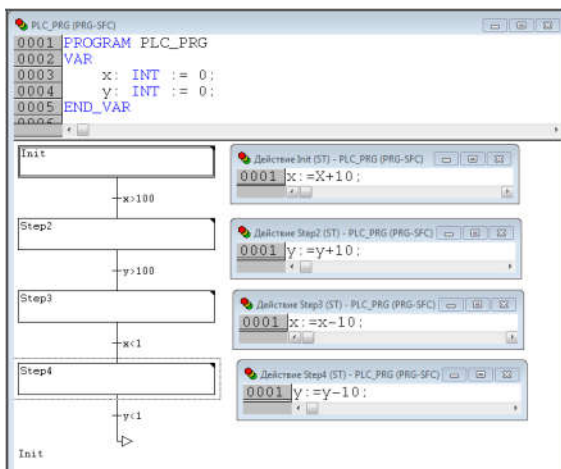


Рисунок 4.2 – Фрагмент програми користувача з вкладеними діями

У діях кроків інкрементуються та декрементуються координати *X* та *Y*, які визначають положення об'єкта. Умовою переходу на наступ-

ний крок є виконання порівнянь поточного значення координат X та Y зі значенням констант. Таким чином координати X та Y спочатку змінюються від 0 до 100 одиниць, а далі – від 100 до 0 одиниць. На рис. 4.3 наведено візуалізацію проекту з параметрами налаштування графічного елементу «квадрат».

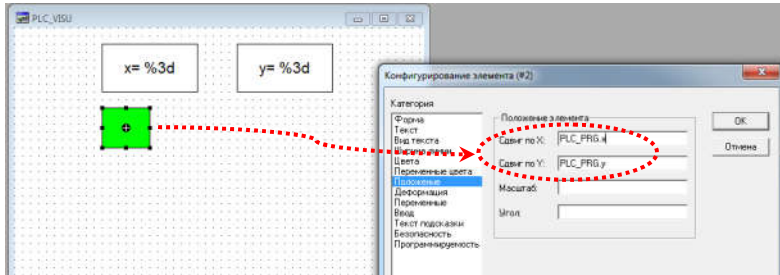


Рисунок 4.3 – Візуалізація проекту з параметрами налаштування рухомого графічного елементу «квадрат»

Створіть в проекті візуалізацію з ім'ям PLC_VISU, подібну тій, що зображена на рис. 4.3. Для відображення значення координат X та Y у візуалізації створіть графічний елемент «прямокутник» з текстом, який динамізується. Таким чином після запуску проекту рухомий елемент буде змінювати своє положення відповідно до значень координат, які зображені в прямокутниках.

Для перевірки проекту запустіть спочатку *SoftPLC SP PLCWinNT*, а далі завантажте до нього проект. Спостерігайте у візуалізації за рухомих об'єктом та його координатами.

Зауважимо, що розрахунок координат здійснюється контролером з часом циклу, який дорівнює часовому таймінгу операційної системи. Цей час становить приблизно 50 мс. У цьому випадку візуалізація, яка відображається в *WEB*-браузері, буде оновлюватись з деякою затримкою, яка зумовлена часом формування запитів та оброблення прийнятих відповідей від сервера. Тому потрібно збільшити час сканування контролера. Це робиться за допомогою створення в конфігурації ресурсів ПЛК у вкладці

Конфігурація задач окремої задачі типу *Циклическая* із часом циклу 2 с. Тобто, перерахунок координат буде здійснюватись один раз на дві секунди. На рис. 4.4 показано параметри налаштування циклічної задачі.

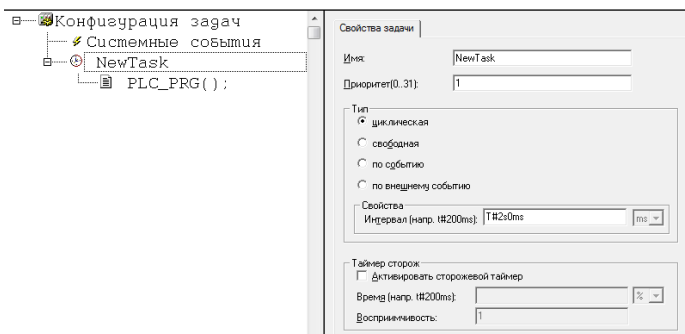


Рисунок 4.4 – Налаштування задачі в проєкті

4.4.2. Конфігурування та запуск WEB-візуалізації

Після інсталяції середовища *CoDeSys* у складі в директорії *Visu* зі шляхом доступу (за умовчанням це місце встановлення середовища *CoDeSys*) *C:\Program Files\3S Software\CoDeSys V2.3\Visu* знаходиться файл *webvisu.htm*, який є базовою *HTML*-сторінкою для *WEB*-візуалізації. При необхідності можна відредагувати цей файл, перш ніж він буде завантажений у контролер разом з кодом проєкту.

Після запуску на редагування, наприклад програмою *Блокнот*, можна побачити вміст файлу, зображений нижче:

```
<HTML>
<HEAD>
  <TITLE>Applet HTML Page</TITLE>
</HEAD>
<BODY>
<APPLET CODEBASE=. CODE=webvisu/WebVisu.class archive="webvisu.jar,minml.jar" name="WebVisu" width="1600" height="1200">
```

```
<param name="STARTVISU" value="PLC_VISU">
<param name="UPDATETIME" value="1000">
<param name="USECURRENTVISU" value="FALSE">
</APPLET>
</BODY>
</HTML>
```

У цьому файлі визначені параметри налаштування *WEB*-візуалізації:

- параметр **STARTVISU** – визначає стартовий об'єкт візуалізації. За умовчанням це **PLC_VISU**;
- **UPDATETIME** – час моніторингу у мілісекундах;
- **USECURRENTVISU** – визначає чи повинна змінюватися візуалізація, якщо в ПЛК-програмі змінюється значення системної змінної «*CurrentVisu*».

Параметри "**width**" (ширина) та "**height**" (висота) встановлюють розмір екрана. Зверніть увагу на можливість обмежити цей розмір вже при створенні візуалізації в *CoDeSys*.

При підключенні до контролера відбудеться компіляція проекту та завантаження файлів візуалізації і коду проекту у контролер.

4.4.3. Конфігурування файла конфігурації *WEB*-сервера та його запуск

Файл конфігурації сервера повинен створюватися у форматі *XML* і носити ім'я **webserver_conf.xml**. Він повинен знаходитись в одному каталозі з **webserver.exe**. Якщо файл конфігурації відсутній, то використовуються згадані вище налаштування за умовчанням.

Приклад конфігурації *WEB*-сервера може виглядати так:

```
<webserver-configuration>
<webserver-port-nr> 8080 </webserver-port-nr>
<target-port-nr> 1200 </target-port-nr>
<target-ip-address> localhost </target-ip-address>
<use-file-upload-dir> true </use-file-upload-dir>
```

```

<file-upload-dir> C:\Program Files\3S Software\CoDeSys V2.3
\Visu\ </file-upload-dir>
<use-intel-byte-order> true </use-intel-byte-order>
<plc-description-file> PlcHandler.ini </plc-description-file>
<plc-entries>
<plc-entry>
<plc-name> MASTER </plc-name>
<plc-directory> .\MASTER </plc-directory>
</plc-entry>
<plc-entry>
<plc-name> FD </plc-name>
<plc-directory> .\FD </plc-directory>
</plc-entry>
<plc-entry>
<plc-name> DL </plc-name>
<plc-directory> .\DL </plc-directory>
</plc-entry>
</plc-entries>
</webserver-configuration>

```

Якщо необхідно підключитися до ПЛК з іншого хоста, то в рядок *IP*-адреси ПЛК `<target-ip-address> localhost </target-ip-address>` замість `localhost` потрібно вказати *IP*-адресу того хоста, на якому запущений *SoftPLC SP PLCWinNT*.

Командний рядок виклику сервера має такий синтаксис:

```

WebServer [webserver-Port-nr] [target-port-nr] [target-IP-address] | [file-
upload-dir] .

```

Лише перераховані параметри можна використовувати в командному рядку. Інші параметри, такі як «порядок байт» та «МультиПЛК функціональність» мають бути задані в конфігураційному файлі.

Таким чином, виклик, відповідний наведеному вище прикладу конфігурації, виглядав би так:

> webserver 8080 1200 localhost c:\Programme\CoDeSysV23.

Для виклику *WEB*-візуалізації через Інтернет в рядок для адреси введіть у браузері таку адресу:

<http://<IP-Адрес Web сервера>:<Порт Web сервера>/webvisu.htm> .

Це буде, наприклад, такий рядок:

<http://localhost:8080/webvisu.htm> .

WebVisu – є стандартною HTML-сторінкою. Вона містить <applet>, який запускає *WebVisu*-аплет, який в свою чергу запускає відображення потрібної сторінки візуалізації. Після цього можна безпосередньо працювати з візуалізацією.

4.5. Оформлення результатів проведення заняття і контрольні запитання

ПІДТВЕРДЖЕННЯМ РЕЗУЛЬТАТИВ ТА ФАКТУ ВИКОНАННЯ РОБОТИ ВВАЖАЄТЬСЯ МОЖЛИВІСТЬ ДОСТУПУ ДО *WEB*-ВІЗУАЛІЗАЦІЇ В ПЛІК З БУДЬ-ЯКОГО КОМП'ЮТЕРА

Складіть звіт про виконану лабораторну роботу, де наведіть схему з'єднань, умови обміну даними та лістинг програми в ПЛК.

За підсумками проведення заняття студент повинен дати логічне пояснення отриманих результатів. Крім того, студент повинен відповісти на контрольні запитання, приклади яких наведено нижче:

1) Який алгоритм доступу до *WEB*-візуалізації в *SoftPLC SP PLCWinNT* за допомогою *WEB-сервера*?

2) Перелічіть основні параметри налаштування конфігураційного файлу *WEB-сервера*.

3) Який синтаксис рядка запиту *WEB*-візуалізації у *SoftPLC SP PLCWinNT*?

4) Охарактеризуйте вміст файлу *webvisu.htm*. Які параметри потребують обов'язкового налаштування?

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ

для проведення лабораторних занять з курсу
«Комп'ютерні мережі»
(у двох частинах)

для студентів напряму підготовки 050202
«Автоматизація та комп'ютерно-інтегровані технології»
денної та заочної форм навчання

Частина 2

Укладачі: ПОДУСТОВ Михайло Олексійович
ЛИСАЧЕНКО Ігор Григорович
ЛОБОЙКО В'ячеслав Олексійович
ШУТИНСЬКИЙ Олексій Григорович

Відповідальний за випуск М.О. Подустов

Роботу до видання рекомендувала Н. М. Самойленко

Редактор Л. А. Пустановойтова

План 2015 р., поз. 175

Підп. до друку __.__.15. Формат 60×84 1/16. Папір офсетний.

Друк – ризографія. Гарнітура Times New Roman.

Ум. друк. арк. 3,5. Наклад 50 прим. Зам. № ____. Ціна договірна.

Видавничий центр НТУ "ХПІ".

Свідоцтво про державну реєстрацію ДК № 3657 від 24.12.2009 р.
61002, Харків, вул. Фрунзе, 21.

ТОВ "Видавництво"Підручник НТУ "ХПІ".

Свідоцтво про державну реєстрацію ДК № 3656 від 24.12.2009 р.
61002, Харків, вул. Фрунзе, 21.