

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

**СУЧАСНІ КОМП'ЮТЕРНО-ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ
У РОЗПОДІЛЕНИХ СИСТЕМАХ УПРАВЛІННЯ**

Навчально-методичний посібник

Харків – 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

І.Г. Лисаченко, М.О. Подустов, А.К. Бабіченко, А.І. Дзевочко

**СУЧАСНІ КОМП'ЮТЕРНО-ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ
У РОЗПОДІЛЕНИХ СИСТЕМАХ УПРАВЛІННЯ**

Навчально-методичний посібник

для студентів спеціальності

151 – «Автоматизація та комп'ютерно-інтегровані технології»
денної та заочної (дистанційної) форм навчання

Затверджено
редакційно-видавничою
радою університету,
протокол № 1 від 25.02.2021 р.

Харків
Друкарня Мадрид
2021

УДК 004.75:621.39:681.51

С89

Автори:

І.Г. Лисаченко, канд. техн. наук, доцент;

М.О. Подустов, д-р техн. наук, проф.;

А.К. Бабіченко, канд. техн. наук, проф.;

А.І. Дзевочко

Рецензенти:

І.Ш. Невлюдов, д-р техн. наук, проф., Засл. діяч науки і техніки України, зав. кафедри КІТАМ, ХНУРЕ;

А.О. Подорожняк, канд. техн. наук, доцент каф. ОТП, НТУ «ХП»

С89 Сучасні комп'ютерно-інформаційні технології у розподілених системах управління: Навч.-метод. посіб. / Лисаченко І.Г., Подустов М.О., Бабіченко А.К., Дзевочко А.І. – Харків: Друкарня Мадрид, 2021. – 94 с.
ISBN 978-617-7988-34-1

В навчально-методичному посібнику розглянуті питання використання сучасних комп'ютерно-інформаційних технологій у складі розподілених систем управління. Наведені основні теоретичні відомості та запропоновані практичні завдання щодо створення макетів РСУ, які використовують сучасні протоколи та інтерфейси, а також апаратно-програмні засоби для побудови гетерогенних комп'ютерних мереж із клієнт-серверною архітектурою.

Призначено для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», які навчаються за денною та заочною (дистанційною) формою навчання.

Іл. 57. Табл. 6. Бібліогр.: 10 назв.

УДК 004.75:621.39:681.51

ISBN 978-617-7988-34-1

© Колектив авторів, 2021

© НТУ «ХП», 2021

© ТОВ «Друкарня Мадрид», 2021

ВСТУП

Навчально-методичний посібник вміщує необхідну інформацію для проведення комп'ютерного практикуму зі студентами денної та заочної (дистанційної) форм навчання за спеціальністю 151 – «Автоматизація та комп'ютерно-інтегровані технології» з навчальної дисципліни «Розподілені системи управління». Він призначений для вивчення студентами сучасних комп'ютерно-інформаційних технологій у розподілених системах управління (PCY) та доповнює теоретичні відомості, які викладені під час проведення лекцій.

Широке використання технології *IoT* (скор. від *Internet of Things*) для домашньої автоматизації визначило розвиток іншої технології – *IIoT* (скор. *Industrial Internet of Things*) для умов промислових виробництв. Як наслідок, у світі спостерігається так звана четверта індустріальна революція, яка має своє визначення – *Industry 4.0*. Характерні риси цього тренду – це повністю автоматизовані виробництва, на яких керівництво усіма процесами здійснюється у режимі реального часу та з урахуванням мінливих зовнішніх умов.

Елементи PCY складаються із промислового контролера (ПЛК) або промислового комп'ютера (ППК), панелі оператора (ОП), пристроїв зв'язку з об'єктом (ПЗО) або локальних регуляторів (Р). Усі зазначені пристрої з'єднанні за допомогою послідовних інтерфейсів *RS-485* та *RS-232* і взаємодіють за різними промисловими протоколами. Як правило такі пристрої не мають інтерфейсів для інтегрування у локальні мережі *Ethernet*. Крім того, бувають випадки, коли необхідно здійснювати обмін даними на відстанях та в умовах, які не можуть забезпечити послідовні інтерфейси. Вказані недоліки можна усунути за рахунок побудови PCY на новітніх принципах та досягненнях у галузі інформаційних технологій. Тому виникає необхідність у створенні PCY, які використовують не лише дротові послідовні інтерфейси та локальні

мережі *Ethernet*, а, наприклад, бездротові канали зв'язку. Причому все частіше такі системи будують на основі безпроводних технологій з застосуванням радіомодемів. Це використання технології низькошвидкісного радіозв'язку *LoRa* (скор. від назви «*Long Range*»). У цих випадках локальні пристрої збору інформації та керування технологічним об'єктом не можуть безпосередньо інтегровані у такі мережі. Потрібні додаткові пристрої, які створюють шлюзи між пристроями, що мають різні інтерфейси та протоколи.

Комп'ютерний практикум у складі посібника дозволяє студентам закріпити на практиці теоретичні відомості про вказані технології та навчитись конфігурувати пристрої для організації обміну даними у гетерогенних мережах. Під час виконання завдань студенти створюють макети РСУ, які використовують промисловий протокол *ModBus*. Як об'єкт керування у практикумі використана модель теплового об'єкта, якою управляє промисловий контролер. Також, для побудови РСУ використані пристрої зв'язку із об'єктом типу *RTU* (скор. від *Remote Terminal Unit*), які виконують функції вимірювання та керування. Метою кожного завдання є організація зв'язку між пристроями з різними комунікаційними інтерфейсами для безперервного моніторингу та керування технологічним процесом у режимі реального часу.

У практикумі розглянуті можливості використання протоколу *MQTT* (скор. від *message queuing telemetry transport*, – транспортний протокол для передавання повідомлень). Цей протокол по сутності є спрощеним мережним протоколом, який працює поверх стека протоколів *TCP/IP*, та орієнтований для обміну повідомленнями між пристроями за принципом «*Видавець-Передплатник*» за допомогою програми-брокера. Це так звана технологія *M2M* (скор. від *Machine-to-Machine* – взаємодія між двома машинами). Термін «машина» тут застосовано як технічний пристрій із трансівером.

Також у посібнику розглянуті принципи створення РСУ із застосуванням хмарних технологій, тобто використання хмарних сервісів для збору та розподілення інформації про стан технологічного об'єкту та параметри технологічного процесу.

Апаратне та програмне забезпечення комп'ютерного практикуму у кожному завданні описується у необхідному об'ємі, що цілком достатньо для його виконання. Крім того наведені посилання для завантаження необхідних матеріалів та програмного забезпечення з *Internet*.

Практичне завдання 1
ПРИНЦИПИ ПОБУДОВИ РСУ НА ОСНОВІ
ВИКОРИСТАННЯ БЕЗПРОВІДНИХ ЗАСОБІВ
ДИСТАНЦІЙНОГО МОНІТОРИНГУ ТА КЕРУВАННЯ

1.1 Мета проведення практичного завдання

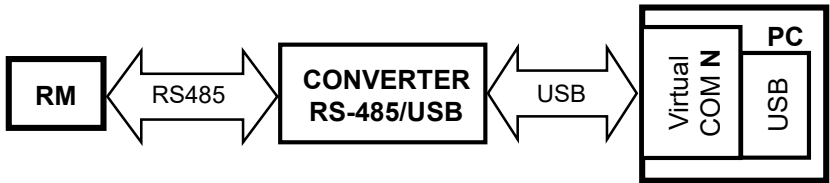
На виконання цього завдання у відповідності до робочої програми відводиться 4 ак. години аудиторного та 4 ак. години часу самостійної роботи студента (СРС).

Мета виконання завдання:

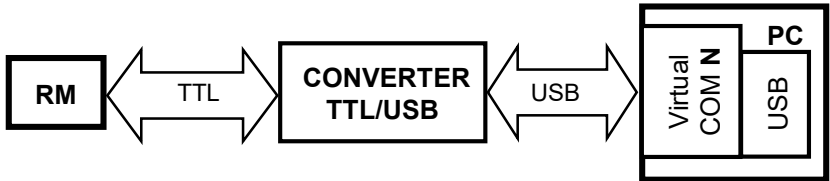
- закріпити на практиці отримані теоретичні відомості про основні принципи організації дистанційного моніторингу та керування віддаленими об'єктами на базі використання безпроводних технологій;
- ознайомитись з принципами роботи низькошвидкісного каналу обміну даними за допомогою радіомодемів;
- ознайомитись з технічними характеристиками та порядком налаштування за допомогою програми конфігурування радіомодемів стандарту *LoRa* модель E32-DTU-433L20 на базі модуля SX1278 виробництва компанії *EBYTE* (Китай) [1];
- розробити та налаштувати макет РСУ для моніторингу модулю вимірювання температури та вологи з боку ПЛК та ПК за допомогою радіомодемів у прозорому режимі обміну «точка-точка».

1.2 Апаратне забезпечення практичного завдання

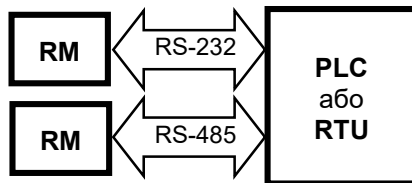
Завдання проводиться на робочих місцях з ПК (*PC*) та стендами, на яких встановлено контролери *ПЛК150 (PLC)* та радіомодеми (*RM*) *LoRa*. У наявності є дев'ять радіомодемів з різними інтерфейсами: три мають інтерфейс *USB/TTL*, один має інтерфейс *RS-232*, два мають інтерфейс *RS-485*, три радіомодеми у металевому корпусі, які мають обидва інтерфейси – *RS-232* та *RS-485*. Комп'ютер та радіомодем з'єднуються послідовним інтерфейсом *RS-232* або за допомогою перетворювачів інтерфейсів *USB/RS-485(232)* та *USB/TTL*. Контролер або віддалений модуль (*RTU*) та радіомодем з'єднуються обома послідовними інтерфейсами: *RS-232* та *RS-485*. Схеми варіантів з'єднань ПК, ПЛК, віддалених модулів та радіомодемів зображені на рис. 1.1 (а, б, в).



a



б



в

Рис. 1.1. Схеми з'єднань радіомодема з ПК, ПЛК та ПЗО

УВАГА!

Номер та параметри віртуального *COM*-порту визначаються за допомогою «Диспетчера устроїв», якій входить до стандартного набору сервісів ОС *Windows*

1.3 Програмне забезпечення практичного завдання

Для зв'язку між ПК та ПЛК послідовним інтерфейсом за допомогою ПІ *RS-485/USB* або *RS-232/USB* в ОС *Windows* за допомогою драйвера емулюється віртуальний *COM*-порт, з яким без обмежень можуть працювати будь-які програми прослуховування послідовних портів, наприклад, програма *COM Port Toolkit*. Однак програма не може інтерпретувати запити за протоколом *ModBus*. Для аналізу *ModBus*-запитів необхідно використовувати спеціальні програми для відправлення за-

питів та розшифрування отриманих відповідей. Це програми *CAS Modbus Scanner* та *CAS Modbus RTU Parser*. Для реалізації головного *ModBus*-пристрою у контролері *ПЛК150* використовується середовище *CoDeSys V2.3* [2, 3]. З іншого боку лінії зв'язку функцію підлеглого пристрою виконує програма *ModSim32*, яка обробляє запити за протоколом *ModBus*.

Для конфігурування параметрів роботи радіомодема використовується програма *RF Setting* (файл запуску програми *RF_Setting.exe*). Програма доступна для завантаження за посиланням <https://www.ebyte.com/en/pdf-down.aspx?id=199>.

1.4 Короткі теоретичні відомості

Технологія низькошвидкісного безпроводного зв'язку між пристроями різного функціонального призначення на цей час використовується у РСУ, де іншими засобами зв'язок неможливий, або утруднено реалізувати безперервний моніторинг та керування технологічними об'єктами у реальному часі, або віддалені пристрої мають автономне живлення. Рішенням є використання технології *LoRa*, у якій використовують розроблені компанією *Semtech Corp.* (США) [4] радіомодулі, спочатку для *Інтернету речей*, а потім для індустриальних застосувань. Технологія *LoRa* (скор. від назви «*Long Range*») – це метод модулювання радіосигналів з розширеним спектром, який забезпечує обмін даними невеликого об'єму на великих відстанях пристроїв з малим споживанням енергії. Усі пристрої технології *LoRa* підтримують протокол обміну *LoRaWAN* (від англ. *Long Range Wide Area Networks*) [4], який де-факто став стандартом у багатьох галузях використання різноманітних систем управління.

У завданні використані радіомодеми серії *E32*, до складу яких входить трансівер *SX1278*. Зовнішній вигляд модуля *E32* зображений на рис. 1.2.

Модуль *E32-433T30D* від компанії *EBYTE* має послідовний порт (*UART/TTL*). Він може працювати у різних режимах обміну даними у діапазоні частот 410...441 МГц (за умовчанням встановлена частота 433 МГц). Дані, що передаються за технологією *LoRa*, у модулі коду-

ються та стискаються. Швидкість передавання радіоканалом за умовчанням встановлена на рівні 2,4 кбіт/с, але може бути змінена.

Основні технічні характеристики модуля E32-433T30D:



Рис. 1.2. Зовнішній вигляд трансівера E32

- довжина лінії зв'язку до 8 км;
- швидкість передавання даних радіоканалом 0,3...19,2 кбіт/с;
- внутрішня буферна пам'ять 512 байт;
- живлення 3,3...5 В;
- струм у режимі передавання даних до 106 мА, у режимі приймання даних – 25 мА;
- потужність передавача до 20 дБм;
- чутливість приймача на рівні

146 дБм на швидкості 2,4 кбіт/с.

На платі модуля є контакти (усього їх сім) для послідовного TTL-інтерфейсу (*RxD*, *TxD*), для вибору режиму обміну (*M0*, *M1*), для живлення (*V5V*, *GND*) та контакт для підключення мікроконтролера (*AUX*), який інформує про статус або режим роботи модуля.

Модуль підтримує два режими роботи: фіксований (див. рис. 1.3) та широкомовний (див. рис. 1.4)

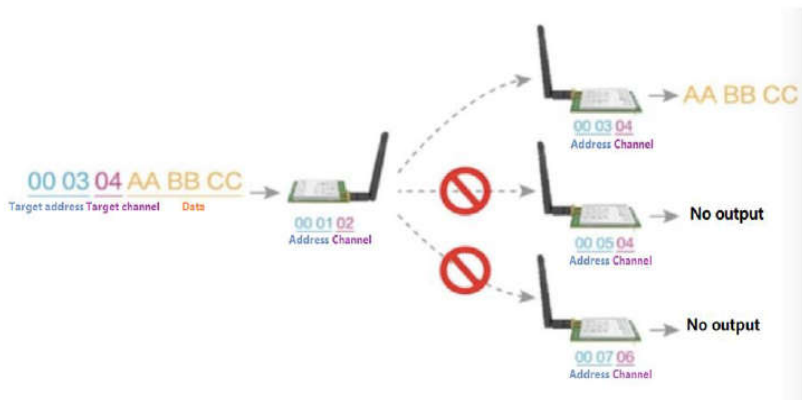


Рис. 1.3. Схема роботи модуля SX1278 у фіксованому режимі

У прикладі на рис. 1.4 адреса передавача у ширококомовному режимі 0xFFFF або 0x0000, частотний канал – 0x04. Якщо модуль передає дані у прозорому режимі, то приймати кадри можуть усі модулі, але лише на каналі 0x04.

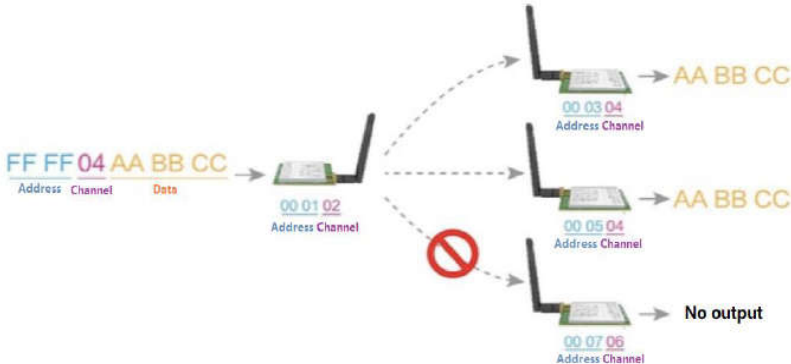


Рис. 1.4. Схема роботи модуля SX1278 у ширококомовному режимі

Контакт *AUX* інформує керуючий пристрій (*MCU*) про готовність до передавання даних. Якщо контакт *AUX* має високий рівень (тобто, $AUX=1$), то буферна пам'ять не заповнена, а *MCU* може передавати дані. У протилежному випадку ($AUX=0$) буферна пам'ять зайнята повністю, тобто *MCU* повинен призупинити передавання даних.

Режими роботи модему визначаються *DIP*-перемикачами *M0* та *M1* або джамперами на платі адаптера модуля (для безкорпусного варіанта виготовлення). У табл. 1.1 наведена інформація про режими роботи модуля в залежності від положення *DIP*-перемикачів (джамперів).

У нормальному режимі модуль отримує дані від послідовного порту та передає їх до радіоканалу пакетами розміром по 58 байт. Вхідні дані безперервно передаються до радіоканалу, якщо їхній об'єм становить 58 байтів. Якщо об'єм даних менше ніж 58 байт, то модуль робить паузу довжиною передавання трьох байтів та потім передає їх. Коли модуль приймає перший пакет з даними вихід *AUX* має низький рівень. Після початку передавання даних до радіоканалу вихід *AUX* перемикається у стан з високим рівнем. Після початку отримання з радіоканалу пакетів з даними у модулі-приймачі вихід *AUX* має низький рівень. Із затримкою

5 мс модуль-приймач передає ці дані на вихід *TxD* свого послідовного порту. Після передавання усіх отриманих даних до порту модуль-приймач перемикає вихід *AUX* у стан з високим рівнем.

Таблиця 1.1 – Режими роботи модуля

Режим роботи (0...3)	Стан <i>M0</i>	Стан <i>M1</i>	Характеристика режиму	Прим.
0 – норм.	0	0	Порт <i>UART</i> та радіоканал відкриті, «прозорий» режим передавання	Приймачі у режими «0» або «1»
1 – « <i>Wake up</i> »	1	0	Порт <i>UART</i> та радіоканал відкриті. Відмінність від режиму «0» автоматичне додавання кода <i>Wake up</i> перед даними для «пробудження» приймача у режимі «3»	Приймачі у режими «0», «1» або «2»
2 – Енергозбереження	0	1	Порт <i>UART</i> закритий, радіоканал у режимі <i>Wake up</i> . Після отримання даних порт <i>UART</i> відкривається для передавання даних	Передавач у режимі «1». Можливе передавання даних
3 – «Сплячий»	1	1	У «сплячому» режимі доступне отримання команди на встановлення параметрів роботи модуля	

У режимі «*Wake up*» модуль-передавач діє подібно до нормального режиму. Відмінність полягає у додаванні до пакетів з даними спеціального коду для «пробудження» модуля-приймача. Довжина коду залежить від часу «пробудження» у налаштуваннях модуля. У модулі-приймачі повинен бути встановлено режим енергозбереження (2). Алгоритм роботи модуля-приймача співпадає з алгоритмом модуля-приймача у нормальному режимі.

У режимі енергозбереження порт *UART* модуля-передавача закритий і він не може отримувати дані від *MCU*. Звичайно функція передавання даних по радіоканалу неможлива для модуля у цьому режимі.

Приймання даних у режимі енергозбереження здійснюється за таким алгоритмом. По-перше, передавання даних здійснює модуль, який налаштований на режим «*Wake up*». Модуль увесь час слідкує за радіоканалом. Якщо у радіоканалі з'явиться код *Wake up*, модуль починає приймати усі дані. Вихід *AUX* знаходиться у стані низького рівня. Після затримки 5 мс модуль відкриває послідовний порт для передавання до лінії *TxD* прийнятих по радіоканалу даних до *MCU*. Після завершення процесу передавання даних до *MCU* вихід модуля *AUX* перемикається на високий рівень. Далі, після встановленого часу *Wake up* модуль змінює свій режим роботи на режим енергозбереження та слідкування за кодом *Wake up*.

У «сплячому режимі» передавання та приймання даних неможливе. Цей режим призначений для конфігурування модуля послідовним інтерфейсом з параметрами зв'язку, наприклад, швидкість – 9600, формат кадру – «8-n-1». Після завершення конфігурування для того щоб встановлені параметри було активовано, потрібно модуль перезавантажити. У режимі конфігурування вихід *AUX* буде у стані низького рівня. Після перезавантаження модуля вихід *AUX* перемикається у стан високого рівня.

У режимі конфігурування модуля програма *RF Setting* генерує команди згідно з табл. 1.2.

За умовчанням команда має такий вигляд: C0 00 00 1A 17 44. Розшифрування цієї команди буде таким:

- адреса модуля у мережі – 0x0000;
- номер радіоканалу – 0x17 (23₁₀), що відповідає значенню 433 МГц;
- швидкість передавання даних радіоканалом – 2,4 кбіт/с;
- швидкість послідовного інтерфейсу – 9600 біт/с;
- формат кадру – «8-n-1»;
- потужність передавача – 1 Вт.

Четвертий байт команди (0x1A) визначає саме параметри послідовного інтерфейсу та швидкість радіоканалу. Частота сигналу радіоканалу знаходиться у діапазоні від 410 до 441 МГц зі зміщенням 1 МГц та визначається за номером. Усього у модулі є 32 канали для напівдуп-

лексного режиму обміну. Так, наприклад, номер каналу 0x17 (23₁₀) має частоту 433 МГц. Для формування декількох ліній зв'язку рекомендовано використовувати канали зі зміщенням 2...5 МГц. Останній байт команди (0x44) визначає режим роботи модуля, режим керування виходами *RxD*, *TxD* та *AUX* модуля, час *Wake up*, режим зниження впливу завад та потужність передавача.

Таблиця 1.2 – Команди для конфігурування модему

Номер команди	Формат команди	Призначення команди та її розшифрування
1	C0+параметри роботи	C0 + 5 байтів з параметрами роботи у <i>HEX</i> -форматі. Усього команда вміщує 6 байтів. Параметри запам'ятовуються після вимикання модуля
2	C1+C1+C1	Команда читання поточних параметрів. Запам'ятовування параметрів здійснюється після вимикання модуля
3	C2+параметри роботи	Відправлення C1 три рази у <i>HEX</i> -форматі. Модуль повертає збережені параметри
4	C3+C3+C3	C2 + 5 байтів з параметрами роботи у <i>HEX</i> -форматі. Усього команда вміщує 6 байтів. Параметри не запам'ятовуються після вимикання модуля
5	C4+C4+C4	Відправлення C3 три рази у <i>HEX</i> -форматі. Модуль повертає версію прошивання модуля

1.5 Послідовність виконання завдання

Практичне завдання складається з двох частин та виконується одночасно на двох робочих місцях. Найвне обладнання дозволяє виконати завдання на трьох парах робочих місць. Причому, з одного боку використані модеми на стендах з ПЛК (з інтерфейсами *RS-485* або *RS-232*), з іншого – використані модулі, які встановлені на адаптери *USB/TTL*.

Перша частина завдання полягає у налаштуванні контролера *ПЛК150* у середовищі *CoDeSys* для циклічного передавання повідомлень про виміряну температуру всередині теплового об'єкта. Програма користувача для роботи з послідовним портом контролера надається. З протилежного боку прийняти радіоканалом повідомлення надходять до програми *COM Port Toolkit*. Для виконання цього завдання задіяні ПК з модемом (на адаптері *USB-TTL*) з одного боку та *ПЛК150* з модемом (з інтерфейсами *RS-485* або *RS-232*) – з іншого боку. Завдання виконується по черзі для усіх пар виконавців завдання.

Друга частина завдання полягає у формуванні запитів від *ПЛК150* за протоколом *ModBus* до підлеглого пристрою, функцію якого виконує програма-симулятор *ModSim32*. Контролер генерує запити, а програма з іншого боку радіоканалу обробляє запити та генерує відповіді. Це завдання теж виконується на обох робочих місцях по черзі усіма парами.

1.5.1 Програмування ПЛК150 для циклічного передавання повідомлень без протоколу

Для виконання першого завдання надається програма, яка відсилає текстове повідомлення «ТЕХТ1» на послідовний порт ПЛК. Програму потрібно доопрацювати таким чином, щоб вона відправляла значення, яке виміряне на аналоговому вході контролера. Також необхідно визначити необхідний номер послідовного порту ПЛК, у залежності від типу інтерфейсу, до якого підключений радіомодем. У документації на ПЛК можна отримати посилання на номери портів його інтерфейсів. Так, у *ПЛК150* є два послідовних порти: *RS-232Debug* та *RS-485*. Ці порти мають відповідні внутрішні адреси: «4» та «0». У програмі користувача використана бібліотека *SysLibCom.lib*, яка призначена для роботи з послідовними портами контролера.

Отже, створіть проєкт у середовищі *CoDeSys* із програмою користувача, лістинг якої наведений та завантажте її код до ПЛК [5]. Нижче наведено лістинг проєкту з блоком об'явлених змінних та програмою користувача, яка розроблена на мові *ST*.

Блок об'явлення змінних для *POU PLC_PRG* буде таким.

```

PROGRAM PLC_PRG
VAR
port_opened: BOOL:=FALSE;
com_handle: DWORD;
com_num: PORTS:=0;
com_set: COMSETTINGS;
com_setEx: COMSETTINGSEX;
res: BOOL;
tOn1: TON;
start_tmr: BOOL:=FALSE;
tr1: R_TRIG;
snd_str: STRING:='TEST1$N';
port_init: BOOL;
rcvbuf: ARRAY [0..1023] OF BYTE;
sz: DWORD;
END_VAR

```

Текст програми користувача у *POU* PLC_PRG буде таким:

```

(*Приклад відправлення у інтерфейс RS-485 кожну секунду
повідомлення "TEST1"*)
(*Налаштування порту: 9600, 7 біт, без парності, один стоп-біт *)
(*Відкриття порту*)
IF NOT port_opened THEN
com_handle:=SysComOpen(com_num);
IF com_handle<>16#FFFFFFFF THEN
port_init:=TRUE;
ELSE
SysComClose(com_num);
com_handle:=SysComOpen(com_num);
port_init:=TRUE;
END_IF
(*Налаштування параметрів порту*)
com_setEx.Size:=SIZEOF(com_setEx);
com_setEx.Port:=com_num;
com_setEx.dwBaudRate:=9600;
com_setEx.byStopBits:=0;

```

```

(*byStopBits:BYTE; Кільк. стоп. біт 0=один, 1=полтора, 2=два *)
com_setEx.byParity:=0;
(* byParity:BYTE; Режим перев. на парність 0=нет, 1=непар,
2=пар*)
com_setEx.dwTimeout:=0;
(* DWORD; Не використовується, повинно бути =0 *)
com_setEx.dwBufferSize:=0;
(* DWORD; Не використовується, повинно бути =0 *)
com_setEx.dwScan:=0;
(*DWORD; Не використовується, повинно бути =0 *)
com_setEx.cByteSize :=7;
(*BYTE; Довжина символу в бітах 5-8 біт. *)
res:=SysComSetSettingsEx(com_num,ADR(com_setEx));
(*Успішно відкрили*)
    IF NOT res THEN
        port_opened:=TRUE;
        END_IF
END_IF
    IF NOT port_opened THEN
        RETURN;
        END_IF
(*Формування повідомлення*)
ton1(In:=start_tmr, pt:=#1s);
start_tmr:=TRUE;
tr1(clk:=ton1.Q);
    IF tr1.Q THEN
        SysComWrite(com_num,ADR(snd_str),LEN(snd_str),0);
start_tmr:=FALSE;
    END_IF
(*Читання буферу з прийнятими даними*)
sz:=SysComRead(com_num,ADR(rcvBUF),1024,0);

```

Запропонований проєкт з програмою користувача є навчальним та має на меті лише демонстрацію можливостей ПЛК щодо керування послідовним портом для відправлення повідомлень за допомогою радіомодема. Тому розглянемо лише основні компоненти програми кори-

стувача. Так, у організаційному компоненті PLC_PRG реалізовано такий алгоритм:

- перевірка на готовність до роботи та відкриття порту;
- встановлення параметрів порту (швидкість та формат кадру);
- якщо порт відкритий без похибок, то формування кожену секунду повідомлень;
- читання буфера даних у кожному циклі ПЛК після його очищення.

Отже, доопрацюйте проєкт згідно умов завдання, скопіюйте та завантажте його код до ПЛК. Далі запустіть проєкт на виконання.

1.5.2 Налаштування радіомодема для зв'язку ПЛК та ПК

Для налаштувань радіомодема запустіть програму *RF Setting*. Вікно запущеної програми зображене на рис. 1.5.



Рис. 1.5. Вікно програми конфігурування модуля E32

Встановіть *DIP*-перемикачами на модулі режим конфігурування («*Sleep Mode*»), тобто встановіть перемикачі M0 та M1 у стан «*OFF*». За вказівкою викладача це робиться також для модулів, які підключені за допомогою адаптерів *USB/TTL* (необхідно від'єднати «джампери» та підключити адаптери з модулями до *USB*-портів ПК).

Налаштування параметрів пристрою починається з вибору мови

інтерфейсу та вибору номера послідовного порту ПК, з яким з'єднаний модуль *E32*. Для цього у відповідному вікні ЛКМ необхідно натиснути на позначку «▼» та у переліку вибрати потрібний порт (номер порту визначити за допомогою диспетчера пристроїв ОС *Windows*). Для відкриття порту необхідно ЛКМ натиснути на кнопку «OpenPort». Натискання на кнопку «Models» відкриє перелік модулів, які можна конфігурувати за допомогою програми.

Якщо потрібний порт відкритий, то активується нижній рядок з кнопками «GetParam», «SetParam» та «Preset». Натискання ЛКМ на кнопку «GetParam» призведе до зчитування поточних параметрів конфігурації модуля, код повідомлення відобразиться у вікні для повідомлень, а самі параметри відобразяться у відповідних полях програми. Ці поля призначені для проведення налаштувань радіоканалу та послідовного інтерфейсу пристрою. Для кожної пари модулів за вказівкою викладача, оберіть однакові канали. Наприклад, для першої пари це буде канал «20», для інших пар – канали «15» та «25». Усі інші параметри можна залишити за умовчанням, тобто відповідно до рис. 1.6. Необхідно лише встановити фіксований режим обміну, тобто активувати параметр «Fixed Mode» (ЛКМ вибрати налаштування «Enable»). У цьому випадку радіоканал буде працювати у прозорому режимі із напівдуплексним каналом та з'єднанням типу «точка-точка».

Після проведених налаштувань потрібно завантажити конфігурацію до модуля та перезавантажити його для активування нової конфігурації. На рис. 1.6 зображене вікно після виконання команди зчитування конфігураційних параметрів.

Наведемо пояснення щодо окремих параметрів налаштування модуля, які встановлюються за допомогою програми *RF Setting*.

Поле «UartRate» призначене для встановлення швидкості послідовного порту модуля. Доступні стандартні значення швидкості від 1200 до 115200 біт/с. За умовчанням встановлена швидкість 9600 біт/с.

Поле «Parity» призначене для встановлення параметрів кадру, що передається, до послідовного порту модуля. Доступні такі варіанти: «8-

n-1», «8-e-1» та «8-o-1». За умовчанням встановлено варіант «8-n-1».

Поле «AirRate» призначене для встановлення швидкості радіоканалу. Модем може передавати дані на швидкостях: 0.3, 1.2, 2.4, 4.8, 9.6 та 19.2 бод. За умовчанням встановлена швидкість 2.4 бод. Якщо потрібна максимальна швидкість, то необхідно враховувати зниження якості радіоканалу та можливі помилки при передаванні даних.

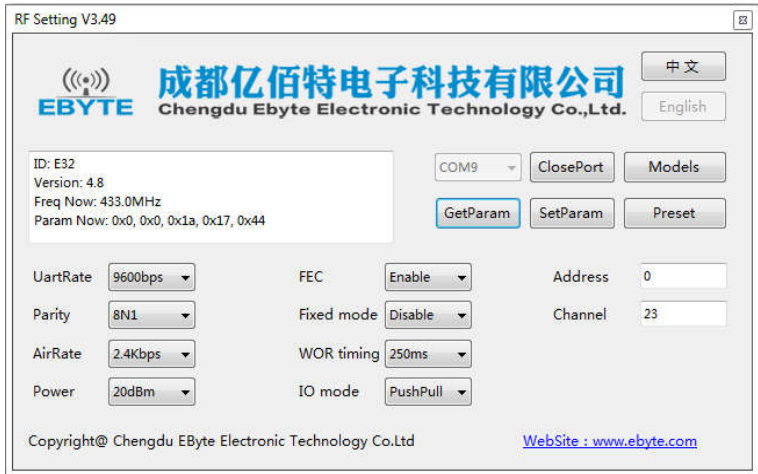


Рис. 1.6. Вікно з параметрами, які отримані від модуля

Поле «Power» призначене для встановлення потужності передавача. Можливе встановлення чотирьох рівнів потужності: 10 дВт (10 мВт), 14 дВт (25 мВт), 17 дВт (50 мВт) та 20 дВт (100 мВт).

У полі «FEC» встановлюється режим контролю кадрів на перевірку їхньої цілісності. Якщо цей режим вимкнений, то знижується час затримки у каналі радіозв'язку.

Поле «Fixed Mode» призначене для вибору режиму фіксованого обміну даними. Це прозорий режим для фіксованих вузлів.

Поле «WOR timing» призначене для встановлення значення часу на пробудження модуля, якщо встановлений режим енергозбереження. Для більших значень цього параметра знижується вживання енергії, але збільшується час затримки на приймання та оброблення даних.

У полі «IO mode» вказується режим керування послідовним пор-

том з боку модуля. За умовчанням встановлений режим із внутрішнім керуванням (PushPull) лініями інтерфейсу *UART/TTL*. Можливе встановлення режиму *OpenDrane*, що є по суті зовнішнім керуванням лініями інтерфейсу *UART/TTL*.

Поле «*Adress*» потрібно для призначення внутрішньої адреси вузлу у мережі *LoRaWAN*. Доступні адреси займають діапазон від 0 до 65535. За умовчанням модуль має адресу, яка дорівнює значенню «0».

Поле «*Channel*» призначене для встановлення номера потрібного каналу, який визначає несучу частоту сигналу. За умовчанням встановлений канал №23, якому відповідає частота 430 МГц. Весь діапазон частот має 32 канали із кроком 1 МГц від 410 до 441 МГц.

Далі необхідно запустити програму *COM Port Toolkit* для моніторингу послідовного порту та спостерігати за отриманими радіомодемом повідомленнями від ПЛК, тобто значення на його аналоговому вході, тобто, температури. Якщо повідомлення не отримуються або приходять спотвореними перевірте налаштування послідовного порту у контролері та у програмі моніторингу. Потрібно, щоб усі налаштування у модемі, ПЛК та програмі збігалися.

1.5.3 Програмування ПЛК150 для роботи у режимі головного пристрою за протоколом ModBus

Друге завдання полягає у налаштуванні зв'язку контролера з програмою симулювання підлеглого пристрою за протоколом *ModBus*, яка запущена на ПК. Усі пари *ModBus*-вузлів взаємодіють за допомогою радіомодемів.

Для виконання другого завдання створіть у середовищі *CoDeSys* новий проєкт та визначте необхідний таргет-файл та мову реалізації *POU PLC_PRG*. Як цільову платформу оберіть ПЛК *ОВЕН150-1.L*, а мову програмування оберіть *CFC*. Контролер буде лише опитувати підлеглий пристрій, тому вікно *POU PLC_PRG* залиште порожнім. Далі перейдіть у вкладення «*Ресурси*» та оберіть ЛКМ утиліту «*Конфігурація ПЛК*». За допомогою контекстного меню (натисканням на ПКМ), додайте програмний модуль *ModBus (Master) [VAR]*, а у ньому змініть інтерфейс *RS-232* на *RS-485* (для . Налаштуйте параметри інтерфейсу

обміну, а саме, встановить швидкість – «9600», налаштуйте формат кадру – «8-n-1» та режим обміну – «RTU».

Далі створіть віртуальну модель підлеглого пристрою – **Universal Modbus device [VAR]** та налаштуйте його параметри – адресу пристрою, наприклад, «16». Інші параметри можна залишити без змін, як це зображено на рис. 1.7 [2, 3, 5].

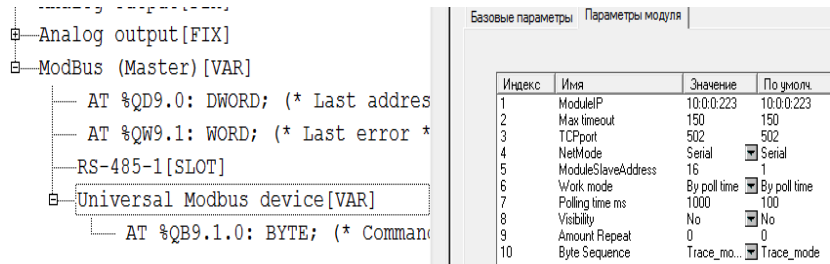


Рис. 1.7. Налаштування модуля підлеглого пристрою

Далі створіть мережні змінні. Це будуть вхідний та вихідний регістри. Для змінних вкажіть номери регістрів та функції, які застосовуються до цих регістрів, як це показано на рис. 1.8:

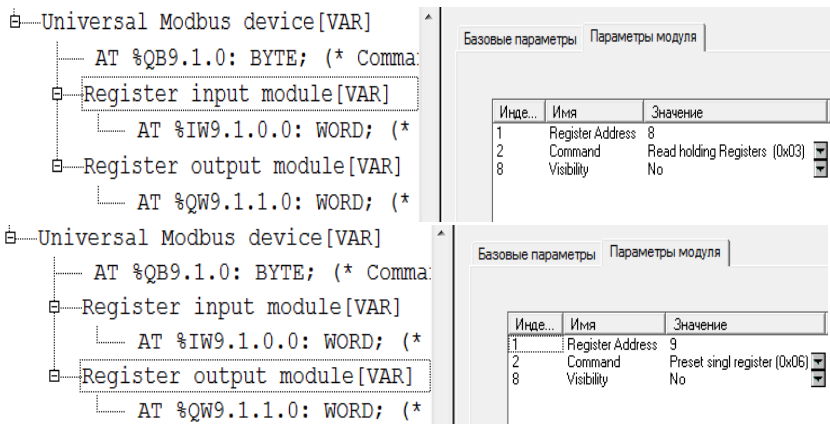


Рис. 1.8. Налаштування мережних змінних

Далі, скопіюйте проєкт та завантажте код проєкту до ПЛК. Активуйте виконання програми користувача у ПЛК. Не відключайтеся

середовищем *CoDeSys* від ПЛК для спостереження за змінними у режимі *Online*. Для зручності спостережень створіть візуалізацію.

1.5.4 Налаштування програми ModSim32 для формування відповідей на запити від головного пристрою

У програмі *ModSim32* створіть та налаштуйте вікно із двома *holding*-регістрами (з адресами «8» та «9»). Адресу підлеглого *ModBus*-пристрою встановіть рівною значенню «16». Режим та параметри зв'язку встановіть наступними: «9600», формат кадру – «8-n-1» та режим обміну даними – «RTU». Налаштуйте регістр №8 на автоматичне змінення значення з інкрементом «1» із циклом змінення 2 с. Отримайте у програмі-симуляторі значення від ПЛК у регістрі №9.

1.6 Завдання для самостійного виконання

Для самостійного виконання пропонується виконати завдання, в яких виконується опитування трьох віддалених модулів (*RTU*) з боку ПЛК150 за протоколом *ModBus*. Перший модуль вимірює температуру та вологість за допомогою датчика *DHT11*, другий – вимірює температуру та відповідний опір за допомогою термометра опору *Pt100*, третій – вимірює температуру за допомогою цифрового датчика температури *DS18B20*. Усі модулі підключені послідовним інтерфейсом (*RS-485* або *RS-232*) до радіомодемів *E32-DTU-433L20* (промислового виконання у металевому корпусі на *DIN*-рейку 35 мм).

Параметри налаштування головного пристрою для зчитування вимірюваних параметрів мають значення, що наведені далі. Модуль з датчиком *DHT11* має адресу 128, а регістри для зчитування параметрів є вхідними (тип «*input*») та мають номери 2 та 3 (адреси 0x0001 та 0x0002). Значення температури та вологості зберігаються у регістрах у вигляді цілого числа, яке на приймальному боці потрібно розділити на число 10. Таким чином, отримуємо реальне значення вимірюваних параметрів. Модуль з датчиком *Pt100* має адресу 64, а регістри для зчитування параметрів є типу «*holding*» та мають номери 1 та 2 (адреси 0x0000 та 0x0001). У першому регістрі зберігається значення температури, у другому – відповідний опір. Зчитані значення потрібно поділити на число 10. Модуль з датчиком температури *DS18B20* має адресу

144, а реєстри для зчитування параметру є типу «*holding*» та мають номери 1 та 2 (адреси 0x0000 та 0x0001).

Зауважимо, що радіомодеми мають налаштування інтерфейсу за умовчанням, а номер радіоканалу у всіх – 23₁₀.

Необхідно розробити програму користувача для *ПЛК150* для зчитування даних від модулів за допомогою радіомодемів на зразок наведеної у п.1.5.3. або у джерелах [2] та [3]. Для зручності зробіть візуалізацію.

1.7 Оформлення результатів проведення заняття

Складіть звіт про проведене заняття, де наведіть схему з'єднань, умови обміну даними. У звіті повинні бути наведені скріншоти з екрану монітора та демонстрацією результатів виконаного завдання.

1.8 Контрольні запитання

- 1) Надайте коротку характеристику властивостей модуля SX1278.
- 2) Наведіть структурну схему модуля SX1278.
- 3) Які режими роботи можна налаштувати для модуля SX1278?
- 4) Як пов'язані потужність передавача, швидкість обміну та довжина лінії зв'язку з якістю зв'язку радіоканалом?
- 5) Які параметри потрібно налаштувати для прозорого обміну фіксованим каналом зв'язку?

Практичне завдання 2

ВИВЧЕННЯ ПРИНЦИПІВ ПОБУДОВИ ГЕТЕРОГЕННИХ РСУ З КЛІЄНТ-СЕРВЕРНОЮ АРХІТЕКТУРОЮ НА ОСНОВІ ВИКОРИСТАННЯ ШЛЮЗУ WIFI–ETHERNET–RS232

2.1 Мета виконання практичного завдання

На виконання цього завдання у відповідності до робочої програми відводиться 2 ак. години аудиторного та 2 ак. години часу самостійної роботи студента (СРС).

Мета завдання:

- закріпити на практиці отримані теоретичні відомості про будову та роботу гетерогенних РСУ із клієнт-серверною архітектурою;
- ознайомитися з технічними характеристиками, функціоналом та можливостями шлюзів із послідовними інтерфейсами на прикладі використання пристрою *HLK-RM04* виробництва компанії *Hi-Link Co* (Китай) [6];
- навчитися конфігурувати шлюз *HLK-RM04* для сумісної роботи засобів РСУ з різними фізичними інтерфейсами без протоколу та за протоколами *ModBus RTU* та *ModBus TCP*;
- розробити та налаштувати РСУ для моніторингу віддаленого модулю вимірювання температури та вологості з боку ПЛК та ПК.

2.2 Апаратне забезпечення практичного завдання

Виконання завдання проводиться на будь-якому робочому місці із ПК та стендом, на якому встановлено контролер *ПЛК150*. Шлюз *HLK-RM04* (модуль з інтерфейсами *WIFI-Ethernet-RS232*) підключений до локальної мережі *Ethernet*. Пристрій встановлений на *DIN*-рейку на стенді та може бути з'єднаний з будь-яким ПЛК для зв'язку з іншим ПЛК або віддаленим модулем. Віддалений модуль (типу *RTU*) з'єднується зі шлюзом інтерфейсом *RS-232*.

Схема варіантів з'єднань ПК, ПЛК, віддаленого модулю та мережного обладнання зображена на рис. 2.1.

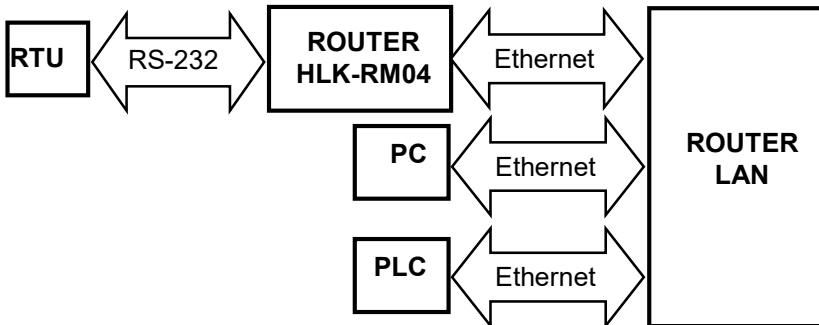


Рис. 2.1. Схема мережних з'єднань

2.3 Програмне забезпечення практичного завдання

Конфігурування модуля *HLK-RM04* здійснюється за допомогою вбудованого у нього *WEB*-сервера по локальній мережі або за допомогою програми *HLK-RM04_CONFIG_en* (якщо модуль підключений до ПК послідовним інтерфейсом *RS-232*). Для доступу до сторінки *WEB*-сервера потрібно у будь-якому браузері ввести *IP*-адресу, яка зазначена на пристрої, та ввести на запит логін (*admin*) та пароль (*admin*). Якщо *IP*-адреса пристрою невідома, то її можна визначити за допомогою програми-сканера локальної мережі *HLK-RM04_Discover*. Ця програма здійснює пошук шлюзу у локальній мережі. Зазначені програми доступні до завантаження за посиланнями:

- а) <http://www.hlktech.net/inc/lib/download/download.php?DIId=23> ,
- б) <http://www.hlktech.net/inc/lib/download/download.php?DIId=24> .

Тестування шлюзу можливе за допомогою тестової програми *USR-TCP232-Test*, яка транслює дані між послідовним інтерфейсом та інтерфейсом *Ethernet*. Програма доступна до завантаження за посиланням <https://www.pusr.com/support/downloads/usr-tcp232-test-V13> . У цьому випадку послідовний інтерфейс шлюзу за допомогою нульмодемного кабелю з'єднати з *COM*-портом ПК. Якщо використаний ПІ, то з'єднання здійснюється за допомогою віртуального *COM*-порту, який створюється ОС *Windows* на ПК. Прослуховування послідовного інтерфейсу здійснюється за допомогою програми *COM Port Toolkit*. Якщо необхідно прослухати локальну мережу *Ethernet*, то можна використати програму *TCP Port Toolkit*.

Для формування запитів до підлеглої пристрою (до *RTU*) за протоколом *ModBus* використовується або програма *Terring Modbus Tools*, або контролер *ПЛК150* із програмою користувача з доданим програмним модулем *ModBus-master*. З іншого боку лінії зв'язку функцію підлеглої пристрою виконує програма-симулятор *ModSim32*, яка теж може бути підключена до послідовного порту ПК або до інтерфейсу *Ethernet* [2, 3].

2.4 Короткі теоретичні відомості

Для інтегрування у локальну мережу пристроїв, які не мають ін-

терфейсу *Ethernet* використовують допоміжні пристрої. Це можуть бути або перетворювачі інтерфейсів, або, більш функціональні, *WiFi-Ethernet-Serial* пристрої з можливістю маршрутизації та підтримкою клієнт-серверного обміну. Модуль *HLK-RM04* є пристроєм, який виконує усі згадані вище функції. На рис. 2.2 зображено схему комунікаційних зв'язків, які забезпечує пристрій *HLK-RM04*. Але повною мірою можна застосувати у модулі усі канали, якщо у нього встановлений конфігураційний режим *Default*. В інших випадках модуль забезпечує обмін даними по схемі «точка-точка», тобто використовує лише два інтерфейси із трьох (див. рис. 2.2).

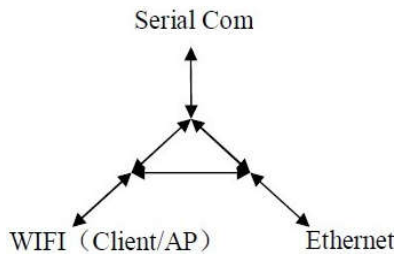


Рис. 2.2. Схема комунікаційних зв'язків *HLK-RM04*

Вбудоване програмне забезпечення модуля підтримує підключення пристроїв з послідовним інтерфейсом (*RS-232* або *RS-485*) та має вбудований стек протоколів *TCP/IP*, що дозволяє створювати шлюз між ними. Також модуль може працювати у режимі бездротової точки доступу або станції-клієнта *WiFi*-мережі. Це надає можливість передавання даних від пристроїв із послідовним інтерфейсом мережею *Ethernet* або зворотно.

У табл. 2.1 наведені основні технічні характеристики модуля *HLK-RM04*. У бездротовому з'єднанні модуль підтримує три режими: клієнт, точка доступу або маршрутизатор. Модуль підтримує усі стандарти шифрування даних (*WPA-PSK/WPA2-PSK*). Модуль забезпечує чотири режими трансляції даних між інтерфейсами: режим «*Default*», режим «*Serial_to_Ethernet*», режим «*Serial_to_WIFI-client*» та режим «*Serial_to_WIFI-AP*».

Схема роботи модуля у режимі «*Serial_to_Ethernet*» зображена на рис. 2.3. У цьому режимі модуль виконує функцію моста між послідо-

вним інтерфейсом та мережею *Ethernet* (*WAN*). Причому модуль у мережі *Ethernet* може отримати *IP*-адресу як клієнт сервісу *DHCP*, який забезпечує сервер мережі (інший роутер), або може використати статичну *IP*-адресу. У цьому режимі другий інтерфейс *Ethernet* (*LAN*) буде заблокований.

Таблиця 2.1 – Технічні характеристики модуля *HLK-RM04*

Мережні стандарти	Бездротові: IEEE 802.11 n/g/b Дротові: IEEE 802.3, IEEE 802.3u
Швидкість передавання даних	Макс. до 150 Mbps (IEEE 802.11 n)
Кількість каналів та частота сигналу	14 каналів на частоті 2.4...2.4835 ГГц
Потужність передавача	10...15 DBm
Інтерфейси	2 – <i>Ethernet</i> (<i>WAN</i> та <i>LAN</i>) 2 – <i>Serial</i> (<i>RS-232</i> або <i>RS-485</i>) 1 – <i>USB</i> (<i>host/slave</i>)

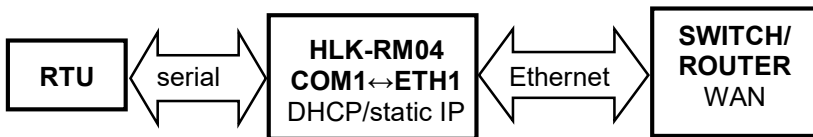


Рис. 2.3. Схема з'єднань у режимі «*Serial_To_Ethernet*»

У режимі «*Serial_to_WIFI-client*» модуль працює клієнтом мережі (див. рис. 2.4). У цьому режимі обидва інтерфейси *Ethernet* модуля заблоковані. Модуль конфігурується на отримання динамічної *IP*-адреси або може використати статичну *IP*-адресу. Шифрування даних у бездротовому режимі обміну використовує стандартні методи. Для автоматичного підключення до мережі *WIFI* у конфігурацію модуля необхідно додати *SSID* мережі та пароль для доступу до неї.

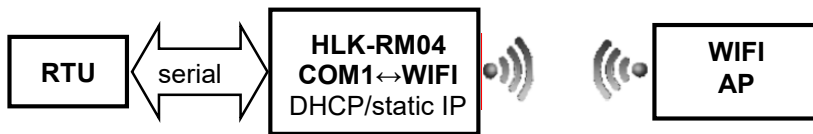


Рис. 2.4. Схема з'єднань у режимі «*Serial_to_WIFI-client*»

У режимі «*Serial_to_WIFI-AP*» модуль виконує функцію точки доступу (*AP*) (див. рис. 2.5). Відповідно, до модуля підключаються інші *WIFI*-клієнти. Усі клієнти отримують динамічні *IP*-адреси у сервісі *DHCP*, який забезпечує модуль. При цьому обидва інтерфейси *Ethernet* заблоковані. Модуль створює власну мережу *WIFI*.

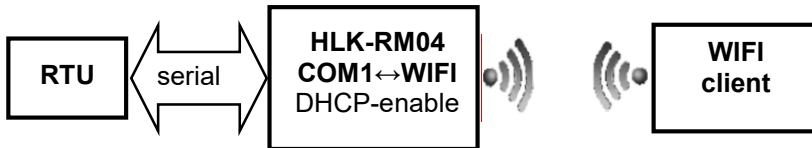


Рис. 2.5. Схема з'єднань у режимі «*Serial_To_WIFI-AP*»

У режимі «*Default*» *WIFI* активовано, а модуль функціонує у режимі точки доступу (*AP*) (див. рис. 2.6). Також активовано обидва інтерфейси *Ethernet*, причому інтерфейс *Eth_1* з'єднаний з глобальною мережею *WAN*, а інтерфейс *Eth_2* – з локальною мережею *LAN*. Мережею *WIFI* до модуля можуть підключитись клієнти мережі. Модуль у мережі *WAN* має динамічну *IP*-адресу. У мережах *LAN* та *WIFI* модуль виконує сервіс *DHCP*.

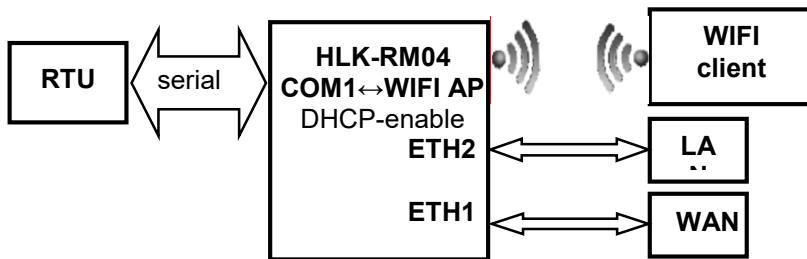


Рис. 2.6. Схема з'єднань у режимі «*Default*»

Модуль підтримує чотири режими обміну даними «*Serial_To_Ethernet*»: *TCP Server*, *TCP Client*, *UDP Server* та *UDP Client*.

У з'єднанні *TCP Server* модуль прослуховує визначений порт, наприклад, стандартний порт 502 протоколу *ModBus-TCP*, та очікує з'єднання з клієнтом. Як тільки сервер отримує запит на з'єднання, він передає усі дані, які він отримує від послідовного інтерфейсу. Тобто, на послідовному інтерфейсі може знаходитись підлеглий пристрій за про-

токолом *ModBus*. Також дані передаються зворотно до клієнта.

У режимі *TCP Client* модуль з'єднаний з визначеним портом. Усі дані передаються з боку *TCP*-сервера безпосередньо до послідовного інтерфейсу та зворотно. У разі порушення зв'язку модуль запускає функцію відновлення підключення.

У режимі *UDP Server* модуль відкриває локальний порт. Дані, що надійшли до порту, модуль спрямовує до послідовного порту та записує на порт віддаленого клієнта. Модуль може фіксувати лише останню інформацію про віддалене з'єднання. Дані, які отримані модулем від послідовного інтерфейсу спрямовуються безпосередньо на порт віддаленого *IP*-адреса.

У режимі *UDP Client* модуль безпосередньо спрямовує на визначений порт та *IP*-адресу віддаленого пристрою (сервера). Дані від сервера спрямовуються до послідовного порту.

Конфігурування модуля може здійснюватись або за допомогою вбудованого до нього *WEB*-сервера, або, використовуючи *AT*-команди у будь-якій термінальній програмі, наприклад, *HyperTerminal*. Конфігурування модуля за допомогою *WEB*-сервера потребує ідентифікації користувача. Конфігурування за допомогою *AT*-команд можливе після перемикання модуля у режим отримання *AT*-команд.

Якщо відома *IP*-адреса модуля *HLK-RM04*, то у будь-якому браузері можливе відкриття сторінки *WEB*-сервера для конфігурування модуля. На рис. 2.7. зображено сторінку конфігурування модуля. Зона сторінки для налаштувань модуля поділена на три частини: зону визначення режиму мережного обміну, зону налаштування послідовного порту та частину із командами підтвердження або відхилення встановлених параметрів.

У даному випадку (див. рис. 2.7) модуль працює у режимі створення мосту від послідовного інтерфейсу до локальної мережі *Ethernet* з активованим сервісом *DHCP*. Параметри послідовного інтерфейсу мають стандартні налаштування (9600, «8-n-1»). Мережний обмін здійснюється за протоколом *TCP* через порт 502 у режимі обміну даними *TCP Server*.

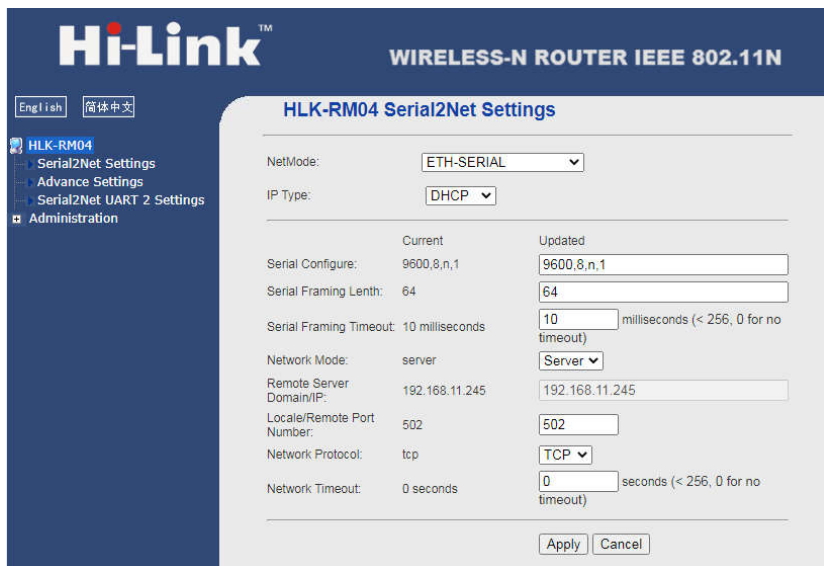


Рис. 2.7. Сторінка конфігурування модуля *HLK-RM04*

2.5 Послідовність виконання завдання

Виконання завдання складається з декількох етапів.

Спочатку необхідно перевірити налаштування модуля та провести тестування його функції за допомогою програми *USR-TCP232-Test*. Для цього послідовний інтерфейс модуля необхідно з'єднати з ПК, а модуль підключити до локальної мережі *Ethernet*. Це підготовчий етап.

Далі виконується перша частина завдання, яка полягає у налаштуванні обміну даними між ПЛК та ПК за допомогою шлюзу *HLK-RM04*. Програма користувача у контролері циклічно відправляє повідомлення про вимірне значення температури всередині теплового об'єкта. Програма користувача для контролера подібна до наданої у п.1.5.1 (див. стор. 13). Контролер *ПЛК150* з'єднується зі шлюзом послідовним інтерфейсом *RS-232*. Для отримання даних на ПК потрібно запустити програму прослуховування інтерфейсу *Ethernet* за *IP*-адресою та портом, які були раніше визначені.

Друга частина завдання – це підключення модуля вимірювання

температури та вологості до *HLK-RM04*. У цьому випадку застосовується протокол *ModBus TCP*, тому для відправлення запитів використаний контролер *ПЛК150* із програмою користувача, до якої входить програмний модуль *ModBus-Master* із комунікаційним модулем для зв'язку по інтерфейсу *Ethernet* та за протоколом *ModBus TCP* [2, 3].

2.5.1 Налаштування модуля *HLK-RM04* для зв'язку із ПЛК та ПК

Запустіть програму *HLK-RM04_Discover* для пошуку модуля у локальній мережі та визначення його *IP*-адреси. Конфігурування модуля буде виконуватись за допомогою *WEB*-сервера через будь-яку програму-браузер для доступу до його *WEB*-сторінки.

Отже, за допомогою браузера підключиться до *WEB*-сервера модуля за визначеною *IP*-адресою. На запит *WEB*-сервера вкажіть логін (*admin*) та пароль (*admin*). Для застосування модуля *HLK-RM04* у мережі налаштуйте його для роботи у режимі «*Serial to Ethernet*», якщо цей режим не встановлено. Перевірте або налаштуйте параметри послідовного інтерфейсу, тобто швидкість – 9600, формат кадру – «8-n-1». Встановіть режим роботи модулю у мережі як *TCP*-сервер та призначте порт обміну «502». Усі потрібні налаштування відповідають зображенню на рис. 2.7.

Далі запустіть програму *USR-TCP232-Test* для тестування модуля. Вікно програми, у якому вже встановленні параметри зв'язку, зображене на рис. 2.8.

Надамо деякі пояснення щодо зображення на рис. 2.8. Так, ліворуч знаходиться зона налаштувань послідовного порту ПК (*COMSettings*), з яким з'єднаний модуль *HLK-RM04*. Параметри налаштувань послідовного порту ПК відповідають налаштуванням послідовного порту у модулі (9600, «8-n-1»). У зоні налаштувань послідовного інтерфейсу знаходиться інтерактивна кнопка *Open/Close* для відкриття або закриття доступу програми до порту. На рис. 2.8 зображено, що послідовний порт відкритий та очікує повідомлення від модуля *HLK-RM04*. Нижче знаходяться зони для налаштування режиму відображення отриманих та переданих повідомлень. У цих зонах є активні кнопки Save... та Clear... для отриманих повідомлень і кнопки Load... та

Clear... для переданих повідомлень. Зверху знаходиться поле (COM port data receive) для отриманих повідомлень від ПК (це, наприклад, отримане повідомлення «Hello HLK-M04»). Знизу знаходиться поле для введення повідомлень (це, наприклад, повідомлення «HELLO WORLD! I am HLK-M04»). Кнопка Send слугує для відправлення повідомлень.

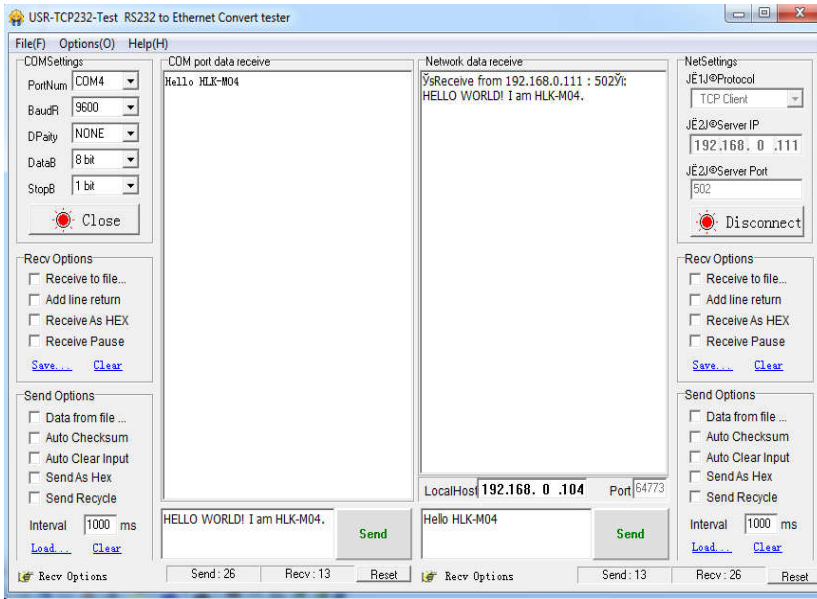


Рис. 2.8. Вікно з налаштуваннями тестової програми

Праворуч знаходиться зона налаштувань мережного інтерфейсу ПК (NetSettings). Тут встановлено протокол обміну у режимі TCP-клієнта та визначено порт 502 для локального хоста, тобто для ПК. Вказано IP-адресу сервера, тобто модуля HLK-RM04 (192.168.0.111). Також вказано IP-адресу клієнта, тобто локального хоста – ПК (192.168.0.104). Для запуску або припинення роботи локального хоста є інтерактивна кнопка Connect/Disconnect. Інші зони праворуч подібні до описаних вище відносно послідовного порту. Внизу вікна з обох боків знаходиться інформаційний рядок, у якому відображаються чисельні характеристики процесу обміну даними та кнопки Reset.

Ця частина завдання вважається виконаною, якщо повідомлення передаються від ПК до модуля та зворотно. Вимкніть модуль та від'єднайте кабель, який з'єднує його з ПК.

2.5.2 Програмування ПЛК150 для циклічного передавання повідомлень

Для виконання цього завдання необхідно змінити схему з'єднань. Попередньо вимкніть живлення усіх пристроїв. Далі з'єднайте порт *RS232 Debug* контролера *ПЛК150* та послідовний порт модуля спеціальним кабелем. З одного боку на кабелі встановлений конектор *RJ-12*, який підключається до ПЛК. З іншого боку кабелю встановлений стандартний конектор для послідовних портів *DB-9 (M)*, який підключається до ПК. Можливе використання додатково ПП *RS-232/USB*.

У завданні використовується програма, яка відсилає текстове повідомлення «TEST1» (див. попереднє практичне завдання, п.1.5.1). Програму потрібно доопрацювати, щоб вона надсилала значення, яке виміряне на 4-му аналоговому вході контролера. Необхідно визначити необхідний номер послідовного порту ПЛК. Так, для порту *RS232 Debug* внутрішній номер дорівнює значенню «4» [2, 3].

Отже, доопрацюйте, скопіюйте проєкт та завантажте його код до ПЛК. Далі запустіть програму користувача на виконання. За допомогою програми *TCP Port Toolkit* для прослуховування мережі *Ethernet* для *IP*-адреси сервера спостерігайте за отриманими повідомленнями. Вікно програми *TCP Port Toolkit* з отриманими повідомленнями зображено на рис. 2.9. На рядку внизу вікна програми вказано, що повідомлення надходять від хоста, у якого *IP*-адреса (192.168.0.111) через порт 502 за протоколом *TCP*. Ця адреса належить модулю *HLK-RM04*.

Налаштування роботи програми *TCP Port Toolkit* є нескладним та інтуїтивно зрозумілим. Достатньо вказати *IP*-адресу і порт та запустити процес для отримання даних від сервера. У цьому випадку програма виконує функцію *TCP*-клієнта.

2.5.3 Програмування ПЛК150 для роботи у режимі головного пристрою

Більшість пристроїв класу *RTU* за протоколом *ModBus* можуть бу-

ти лише підлеглими та не мають інтерфейсу *Ethernet*. Тому у мережі обов'язково повинен бути головний пристрій, який відправляє запити до таких модулів та шлюз, який транслює ці запити на послідовний інтерфейс. Так, у даному випадку модуль *RTU* з'єднаний зі шлюзом *HLK-RM04* послідовним інтерфейсом *RS-232*. Усі запити, які надходять локальною мережею, шлюз у прозорому режимі спрямовує на свій послідовний інтерфейс, а потім отриману відповідь від модуля *RTU* спрямовує у локальну мережу *Ethernet*. Тут шлюз виконує функцію сервера за протоколом *ModBus TCP* на порту «502». Функцію клієнта (головного пристрою) виконує контролер або комп'ютер.

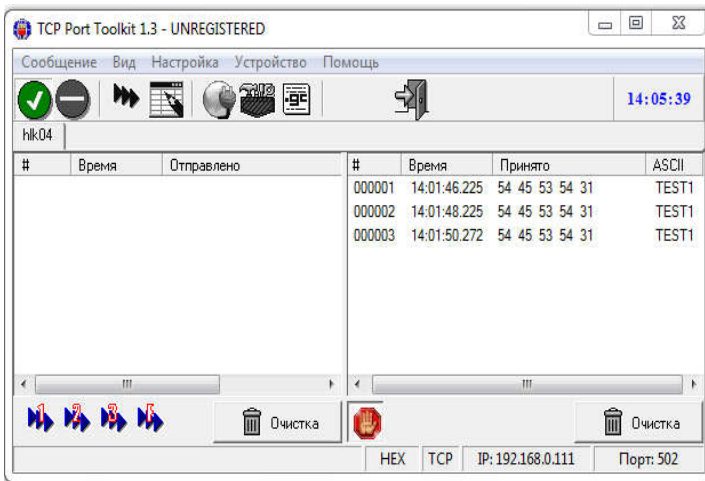


Рис. 2.9. Вікно програми для прослуховування локальної мережі

Отже, створіть у середовищі *CoDeSys* новий проєкт та визначте необхідний таргет-файл та мову реалізації *POU PLC_PRG*. Як цільову платформу оберіть ПЛК *OBEN150-I.L*, а мову програмування виберіть *CFC*. Контролер буде лише опитувати підлеглий пристрій, тому вікно *POU PLC_PRG* залиште порожнім. Далі перейдіть до вкладення «Ресурси» та відкрийте утиліту «Конфігурація ПЛК». У ній, за допомогою ПКМ викличте контекстне меню, додайте програмний модуль *ModBus (Master) [VAR]* [2, 3, 5].

Далі створіть віртуальну модель підлеглого пристрою – **Universal Modbus device [VAR]** та вкажіть його *ModBus*-адресу (32). Встановить роботу комунікаційного модуля за протоколом *TCP* (4-ий рядок у вкладенні з параметрами модуля) та *IP*-адресу шлюзу, яка визначена раніше у попередніх завданнях (192.168.0.111). Важливо, що формат *IP*-адреси у даному випадку як роздільник використовує символ «:», тобто адреса у першому рядку параметрів підлеглого пристрою (тобто, шлюзу) буде такою – 192:168:0:111. Важливо встановити параметр **Byte Sequence** у десятому рядку у значення **Native**. Це потрібно для правильного перетворення отриманих даних відповідно до стандарту *IEEE754* значення параметра типу *Float*, яке надходить до ПЛК двома послідовними регістрами за протоколом *ModBus*. Також змініть частоту опитування, наприклад, одне опитування за одну секунду. Інші параметри можна залишити без змін, як це зображено на рис. 2.10.

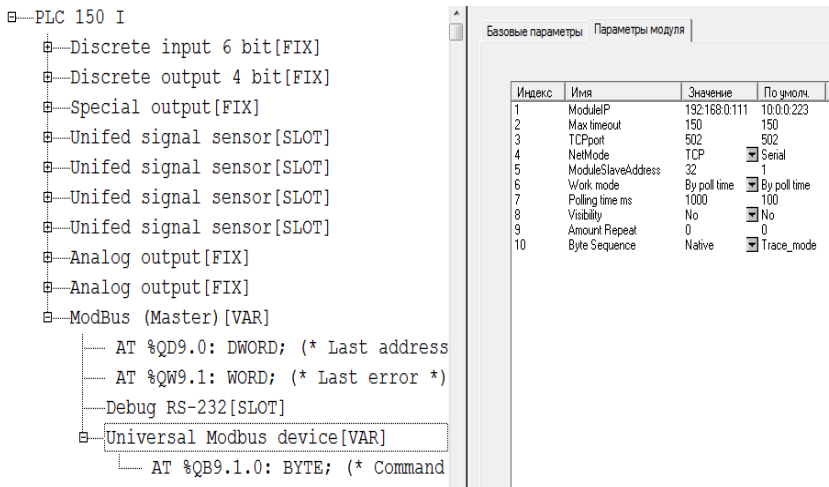
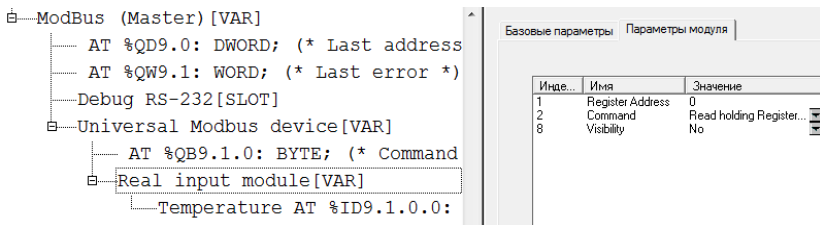


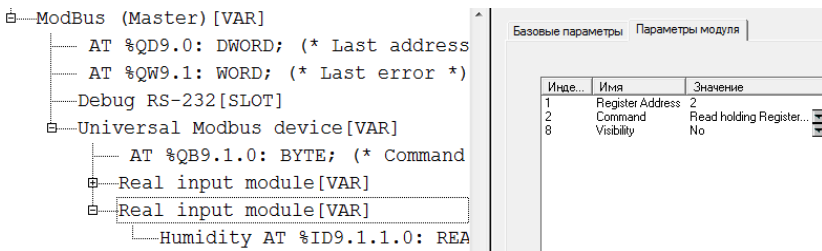
Рис. 2.10. Налаштування модуля підлеглого пристрою

Далі створіть мережні змінні відповідно до ресурсів модуля *RTU*. Цей модуль побудований на базі мікроконтролера *ESP-01 (ESP-8266)* із *WIFI*-модулем та послідовним інтерфейсом *RS-232*. Внутрішнє програмне забезпечення мікроконтролера виконує функцію опитування ком-

бінованого датчику типу *DHT11*, тобто модуль виконує функцію вимірювача температури та вологості. Внутрішня структура його пам'яті за протоколом *ModBus* складається усього з чотирьох реєстрів, по два на кожен параметр. Адреси реєстрів такі: канал вимірювання температури займає реєстри з адресами «0» та «1», канал вологості займає реєстри «2» та «3». Крім того, внутрішнє програмне забезпечення має вбудований *WEB*-сервер зі сторінкою для відправлення до браузера значень вказаних параметрів (для перевірки). Таким чином, до конфігурації ПЛК необхідно додати два модуля типу **Real input module[VAR]** для приймання значень у форматі *Float* (дійсні числа одиничної точності за стандартом *IEEE754*) та вказати адреси для цих каналів. Для змінних вкажіть номери реєстрів та функції, які застосовуються до цих реєстрів, як це показано на рис. 2.11 *а* та *б*.



а



б

Рис. 2.11. Налаштування мережних змінних

Далі, скопіюйте проєкт та завантажте код проєкту до ПЛК. Активуйте виконання програми користувача у ПЛК. Не відключайтеся середовищем *CoDeSys* від ПЛК для спостереження за змінними у ре-

жимі *Online*. Для зручності спостережень додайте до проєкту візуалізацію для відображення значень поточних параметрів. Контролюйте правильність отриманих значень параметрів у браузері за допомогою *WEB*-серверу модуля вимірювання температури та вологості за допомогою каналу *WIFI*.

2.6 Завдання для самостійного виконання

Для самостійного виконання пропонується виконати таке завдання. Потрібно налаштувати опитування підлеглого пристрою ПЛК (*Slave*), у якого порт *RS232 Debug* з'єднаний з послідовним портом *RS232* модуля *HLK-RM04*, з боку головного пристрою ПЛК (*Master*). Для з'єднання ПЛК та модуля *HLK-RM04* використайте кабель із першого завдання (п. 2.5.2). Необхідно отримати значення змінного резистора на четвертому аналоговому каналі підлеглого ПЛК за протоколом *ModBus TCP*. На головному контролері зробіть візуалізацію для відображення отриманого значення.

2.7 Оформлення результатів проведення заняття

Складіть звіт про проведене заняття, де наведіть схеми комунікаційних з'єднань, умови обміну даними, режими роботи роутера. У звіті повинні бути наведені скріншоти з екрану монітора та демонстрацією результатів виконання завдань.

2.8 Контрольні запитання

- 1) Надайте коротку характеристику шлюзу *HLK-RM04*.
- 2) Наведіть структурну схему шлюзу *HLK-RM04*.
- 3) Які режими роботи можна налаштувати у шлюзі *HLK-RM04*?
- 4) Які параметри потрібно налаштувати для прозорого обміну даними скрізь шлюз?
- 5) Наведіть приклади використання шлюзу *HLK-RM04*.

Практичне завдання 3

ВИВЧЕННЯ ПРИНЦИПІВ ПОБУДОВИ ГЕТЕРОГЕННИХ РСУ З КЛІЄНТ-СЕРВЕРНОЮ АРХІТЕКТУРОЮ НА ОСНОВІ ВИКОРИСТАННЯ ШЛЮЗУ ETHERNET–RS485/RS232

3.1 Мета виконання практичного завдання

На виконання цього завдання у відповідності до робочої програми відводиться 2 ак. години аудиторного та 2 ак. години часу самостійної роботи студента (СРС).

Мета завдання:

- закріпити на практиці отримані теоретичні відомості про будову та роботу гетерогенних РСУ із клієнт-серверною архітектурою;
- ознайомитися з технічними характеристиками, функціоналом та можливостями шлюзів із послідовними інтерфейсами на прикладі використання пристрою *E810-DTU* виробництва компанії *EBYTE Co* (Китай) [7];
- навчитися конфігурувати шлюз *E810-DTU* для сумісної роботи засобів РСУ з різними фізичними інтерфейсами без протоколу та за протоколами *ModBus RTU* та *ModBus TCP*;
- розробити та налаштувати РСУ для моніторингу та керування дискретними виходами модулю з боку ПЛК та ПК.

3.2 Апаратне забезпечення практичного завдання

Виконання завдання проводиться на будь-якому робочому місці з ПК та стендом, на якому встановлено контролер *ПЛК150*. Шлюз *E810-DTU* (модуль з інтерфейсами *Ethernet–RS-485/RS-232*) підключений до локальної мережі *Ethernet*. Пристрій встановлений на *DIN*-рейку на переносному стенді та може бути з'єднаний з будь-яким ПЛК для зв'язку з іншим ПЛК, ПК або модулем. Модуль керування дискретними виходами типу *RTU* з'єднується зі шлюзом інтерфейсом *RS-485*.

Схема варіантів з'єднань ПК, ПЛК, віддаленого модулю та мережного обладнання зображена на рис. 3.1.

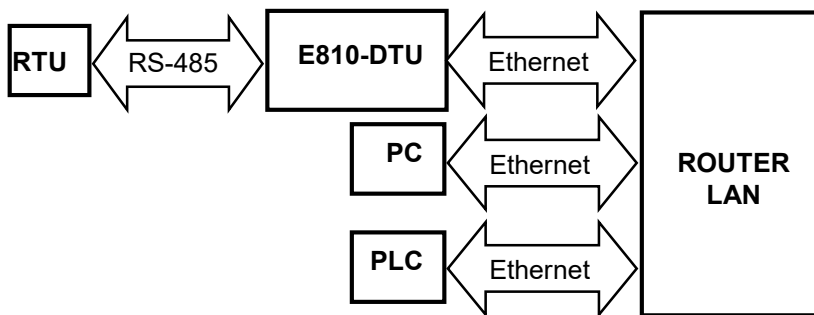


Рис. 3.1. Схема мережних з'єднань

3.3 Програмне забезпечення практичного завдання

Конфігурування модуля *E810-DTU* здійснюється за допомогою вбудованого у нього *WEB*-сервера по локальній мережі або за допомогою програми *EBYTE Configuration Tools*. Програма конфігурування модуля доступна для завантаження за посиланням <https://www.ebyte.com/en/pdf-down.aspx?id=908>. За допомогою цієї програми можна провести пошук модулів у локальній мережі та визначити їхню *IP*-адресу. Якщо проводити конфігурування за допомогою сторінки *WEB*-сервера потрібно у будь-якому браузері ввести *IP*-адресу, яка визначена раніше, та ввести на запит логін (*admin*) та пароль (*admin*).

Тестування шлюзу *E810-DTU* можливе за допомогою тестової програми *TCP/UDP Net Assistant*, яка трансліює дані між послідовним інтерфейсом та локальною мережею *Ethernet* (програма доступна для завантаження за адресою <https://www.ebyte.com/en/pdf-down.aspx?id=765>). У цьому випадку послідовний інтерфейс пристрою потрібно з'єднати з *COM*-портом ПК за допомогою нульмодемного кабелю. Якщо використаний перетворювач інтерфейсів, то з'єднання здійснюється за допомогою віртуального *COM*-порта, який створюється ОС *Windows* на ПК.

Прослуховування послідовного інтерфейсу здійснюється за допомогою програми *COM Port Toolkit*. Якщо необхідно прослухати локальну мережу *Ethernet*, то можна використати програму *TCP Port Toolkit*.

Для формування запитів до підлеглого пристрою (*RTU*) за протоколом *ModBus* використовується або програма *Terring Modbus Tools* або контролер *ПЛК150* із програмою користувача та з доданим програмним модулем *ModBus-master*. З іншого боку лінії зв'язку функцію підлеглого пристрою виконує програма-симулятор *ModSim32*, яка теж може бути підключена до послідовного порту або до інтерфейсу *Ethernet* пристрою. Модуль керування дискретними виходами підтримує протокол *ModBus*.

3.4 Короткі теоретичні відомості

Для інтегрування у локальну мережу пристроїв, які не мають інтерфейсу *Ethernet* використовують допоміжні пристрої. Це можуть бути або перетворювачі інтерфейсів, або, більш функціональні, шлюзи *Ethernet-Serial* з можливістю маршрутизації та підтримкою клієнт-серверного обміну. Модуль *E810-DTU* є пристроєм, який виконує усі згадані вище функції. Особливістю модуля є забезпечення одночасної роботи двох сокетів (А та В) у різних режимах та можливість їхнього підключення до мережі *Ethernet*.

Вбудоване програмне забезпечення модуля підтримує одночасне підключення пристроїв до обох послідовних інтерфейсів (*RS-232* або *RS-485*) та має вбудований стек протоколів *TCP/IP*, що дозволяє створювати шлюз між послідовними інтерфейсами та мережею *Ethernet*. Модуль може створювати власну мережу та забезпечувати сервіс *DHCP* або може використовувати статичне адресування.

Модуль підтримує чотири режимі перетворення даних «*Serial To Ethernet*» на обох сокетах: *TCP Server*, *TCP Client*, *UDP Server* та *UDP Client*.

У з'єднанні *TCP Server* модуль прослуховує визначений порт, наприклад, стандартний порт протоколу *ModBus-TCP* (502), та очікує з'єднання з клієнтом. Як тільки сервер отримує запит на з'єднання, він передає усі дані, які він отримує від послідовного інтерфейсу. Тобто, на послідовному інтерфейсі може знаходитись підлеглий пристрій за протоколом *ModBus*. Також дані передаються зворотно до клієнта.

У режимі *TCP Client* модуль з'єднаний з визначеним портом. Усі дані передаються з боку *TCP*-сервера безпосередньо до послідовного інтерфейсу та зворотно. У разі порушення зв'язку модуль запускає функцію відновлення підключення.

У режимі *UDP Server* модуль відкриває локальний порт. Дані, що надійшли до порту, модуль спрямовує до послідовного порту та записує на порт віддаленого клієнта. Модуль може фіксувати лише останню інформацію про віддалене з'єднання. Дані, які отримані модулем від послідовного інтерфейсу спрямовуються безпосередньо на порт віддаленого *IP*-адреса.

У режимі *UDP Client* модуль безпосередньо спрямовує на визначений порт та *IP*-адресу. Дані від сервера спрямовуються до послідовного порту.

Конфігурування модуля може здійснюватись або за допомогою вбудованого до нього *WEB*-сервера, або, за допомогою *AT*-команд у будь-якій термінальній програмі, наприклад, *HyperTerminal*. Конфігурування модуля за допомогою *WEB*-сервера потребує ідентифікації користувача. Конфігурування за допомогою *AT*-команд можливе після перемикання модуля у режим *AT*-команд. У цьому випадку модуль підключається до ПК послідовним інтерфейсом.

За допомогою програми *EBYTE Configuration Tools* необхідно визначити *IP*-адресу модуля. Після запуску програми змініть мову інтерфейсу на англійську та введіть адресу локального хоста, тобто ПК. У цілому, інтерфейс програми інтуїтивно зрозумілий. Далі, для пошуку модуля у локальній мережі потрібно ЛКМ натиснути на команду «*Search*». Якщо модуль знайдений, то відкриється вікно, яке зображене на рис. 3.2. У рядку таблиці будуть наведені основні відомості про знайдений модуль. За *IP*-адресою у браузері необхідно відкрити сторінку *WEB*-сервера для конфігурування пристрою. На рис. 3.3 зображене вікно головної сторінки *WEB*-сервера із основними поточними налаштуваннями (вкладення *Current State*).



Рис. 3.2. Вікно програми налаштування модуля *E810-DTU*

Далі потрібно відкрити вкладення **IP Setting** для отримання відомостей про мережні налаштування модуля. У цьому вкладенні вказані тип маршрутизації (статична або динамічна адреса), маска мережі, адреса маршрутизатора та адреси *DNS*-серверів. Залишить налаштування без змін, як це зображено на рис. 3.4

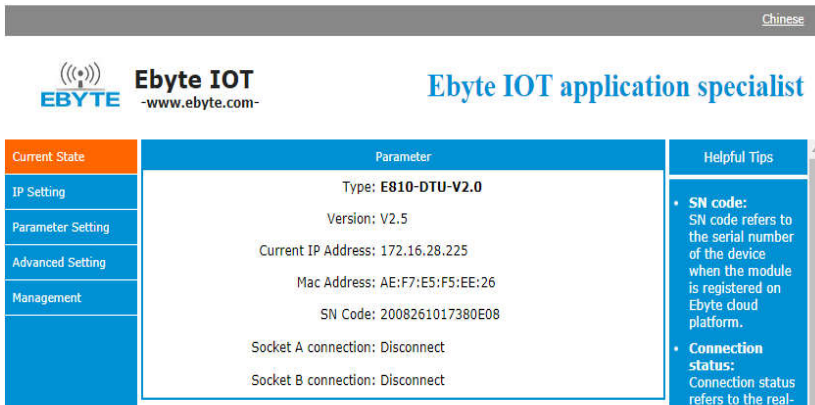


Рис. 3.3. Вікно головної сторінки *WEB*-сервера модуля *E810-DTU*

На рис. 3.5 та 3.6 зображені вкладення **Parameter Setting** із налаштуванням послідовного інтерфейсу та сокетів модуля для роботи у мережі *Ethernet*. На рис. 3.5 **Socket A** відповідає налаштуванням модуля для транслявання даних у режимі сервера (*TCP Server*) від послідовного порту за протоколом *ModBus* з підлеглим пристроєм. На рис. 3.6 **Socket B** налаштований на роботу модуля у режимі клієнта (*TCP Client*). Таке налаштування забезпечує транслявання даних від мережі *Ethernet* до послідовного інтерфейсу. Але цей режим застосовують значно рідше.

Current State	Parameter	Helpful Tips
IP Setting	IP address access mode: STATIC IP ▼ The machine's IP: <input type="text" value="172"/> . <input type="text" value="16"/> . <input type="text" value="28"/> . <input type="text" value="225"/> Subnet mask: <input type="text" value="255"/> . <input type="text" value="255"/> . <input type="text" value="255"/> . <input type="text" value="0"/> The gateway address: <input type="text" value="172"/> . <input type="text" value="16"/> . <input type="text" value="28"/> . <input type="text" value="1"/> DNS server: <input type="text" value="8"/> . <input type="text" value="8"/> . <input type="text" value="8"/> . <input type="text" value="8"/> Standby DNS server: <input type="text" value="172"/> . <input type="text" value="16"/> . <input type="text" value="28"/> . <input type="text" value="1"/> <input type="button" value="save setting"/> <input type="button" value="cancel"/>	<ul style="list-style-type: none"> • IP address access mode: Static IP means static IP address;DHCP means dynamically obtaining IP address, subnet mask and gateway. • The machine's IP address: IP address of the
Parameter Setting		
Advanced Setting		
Management		

Рис. 3.4. Вікно з відкритим вкладенням IP Setting

Current State	Parameter	Helpful Tips
IP Setting	Baud rate: <input type="text" value="9600"/> (300~3000000)bps Data bits: 8 ▼ bit Calibration bit: None ▼ Stop bit: 1 ▼ bit Flow control: NFC ▼ Serial port packing time: <input type="text" value="10"/> (0, 10~255)ms Serial port packing length: <input type="text" value="1460"/> (0, 4~1460)chars Modbus to tcp switch: ON ▼	<ul style="list-style-type: none"> • Local interface 1~65535. When a module does TCP client, a local port of 0 means random local ports are used. • Remote port 1-65535. • Packing time/length Default:10/1000 When set to 0/0, use automatic packaging mechanism. You can also set it to a non-zero value.
Parameter Setting	Socket A Parameter Work mode: TCP Server ▼ Maximum number of TCP server connections: 3 ▼ beyond connection numbers KICK ▼ Local/remote port: <input type="text" value="502"/> <input type="text" value="8887"/> (0~65535) Reconnect time: <input type="text" value="2"/> (0, 2~65535)s Clear cache function: <input type="checkbox"/> Keep alive connections: <input checked="" type="checkbox"/> Detection time: <input type="text" value="10"/> (0 close,2~7200)s Detection of interval: <input type="text" value="5"/> (2~7200) s Number of detection: <input type="text" value="30"/> (2~255)times	
Advanced Setting		
Management		

Рис. 3.5. Вікно з відкритим вкладенням Parameter Setting (Socket A)

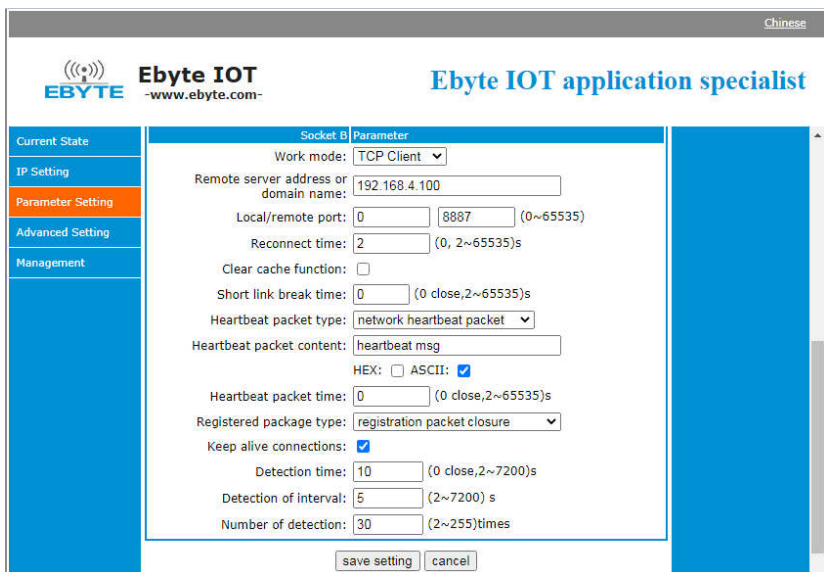


Рис. 3.6. Вікно з відкритим вкладенням Parameter Setting (Socket B)

Схема роботи модуля у режимі «*TCP Server*» для **Socket A** зображена на рис. 3.1. У цьому режимі модуль виконує функцію шлюзу між послідовним інтерфейсом та мережею *Ethernet*. Модуль у мережі *Ethernet* може отримати *IP*-адресу як клієнт сервісу *DHCP*, який забезпечує сервер мережі (інший роутер), або може використати статичну *IP*-адресу.

До послідовного інтерфейсу *RS-485* шлюзу підключений підлеглий модуль для керування дискретними виходами. Керування виходами модуля за протоколом *ModBus* може здійснюватись або апаратно (контролер *ПЛК150*), або програмно (програма *ModSim32*) через локальну мережу *Ethernet*. Параметри послідовного інтерфейсу мають стандартні налаштування (9600, «8-n-1»). Мережний обмін здійснюється за протоколом *TCP* через порт 502 у режимі *TCP Server*.

3.5 Послідовність виконання завдання

Виконання завдання складається з декількох етапів.

Спочатку необхідно перевірити налаштування модуля та, якщо

потрібно, провести тестування його функцій за допомогою програми *USR-TCP232-Test*. Цей підготовчий етап виконується подібно до викладеного у п.2.5.1.

Далі виконується перша частина завдання, яка полягає у налаштуванні обміну даними між ПЛК та ПК за допомогою модуля *E810-DTU*. Програма користувача у контролері циклічно відправляє повідомлення про виміряне значення температури всередині теплового об'єкта. Програма користувача для контролера подібна до наданої у п.1.5.1 (див. стор. 13). Контролер *ПЛК150* з'єднується з модулем *E810-DTU* послідовним інтерфейсом *RS-232* за допомогою спеціального кабелю. Для отримання даних на ПК потрібно запустити програму прослуховування інтерфейсу *Ethernet* за *IP*-адресою та портом, які були раніше визначені для *E810-DTU* у локальній мережі (див. рис. 3.2).

Друга частина завдання – це підключення до модуля керування дискретними виходами. У цьому випадку застосовується протокол *ModBus TCP*, тому для відправлення запитів використаний контролер *ПЛК150* із програмою користувача, до якої входить програмний модуль *ModBus-Master*. Комунікаційний модуль контролера повинен бути налаштований на роботу по інтерфейсу *Ethernet* за протоколом *ModBus TCP* [2, 3].

3.5.1 Налаштування модуля *E810-DTU* для зв'язку ПЛК та ПК

Запустіть програму *EBYTE Configuration Tools* для пошуку модуля у мережі та визначення його *IP*-адреси. Проведіть налаштування модуля за допомогою *WEB*-сервера через будь-яку програму-браузер для доступу до його *WEB*-сторінки. Встановіть налаштування відповідно до рис. 3.4 та 3.5. Тобто параметри послідовного інтерфейсу швидкість – 9600, формат кадру – «8-n-1». Встановіть режим роботи модулю у мережі *Ethernet* як серверу за протоколом *TCP* та призначте порт обміну «502».

За допомогою програми *USR-TCP232-Test* перевірте роботу шлюзу *E810-DTU*. Необхідні відомості наведені у п. 2.5.1 та налаштування подібні зображенню на рис. 2.8. Ця частина вважається виконаною, якщо повідомлення передаються від ПК до шлюзу та зворотно. Вимкніть шлюз та від'єднайте кабель, який з'єднує його з ПК.

3.5.2 Програмування ПЛК150 для циклічного передавання повідомлень

Для виконання цього завдання необхідно змінити схему з'єднань. Попередньо вимкніть живлення усіх пристроїв. Далі з'єднайте порт *RS232 Debug* контролера *ПЛК150* та послідовний порт шлюзу *E810-DTU* спеціальним кабелем. Цей кабель вже застосовувався при виконанні завдання у п.2.5.2, де наведено його опис.

Для виконання цього завдання надається програма, яка подібна до наведеної у п.1.5.1, стор.13). Але, програму потрібно доопрацювати так, щоб вона відсилала значення, яке виміряне на 3-му аналоговому вході контролера (термопара типу «К»). Також необхідно визначити необхідний номер послідовного порту ПЛК. Так, для порту *RS232 Debug* внутрішній номер порту дорівнює значенню «4».

Отже, доопрацюйте, скопіюйте проєкт та завантажте його код до ПЛК. Далі запустіть програму користувача на виконання. За допомогою програми *TCP Port Toolkit* для прослуховування мережі *Ethernet* для IP-адреси сервера спостерігайте за отриманими повідомленнями, як це зображено на рис. 2.9 на стор. 33.

3.5.3 Програмування ПЛК150 для роботи у режимі головного пристрою

Більшість пристроїв класу *RTU* за протоколом *ModBus* можуть бути лише підлеглими та не мають інтерфейсу *Ethernet*. Тому у мережі обов'язково повинен бути головний пристрій, який відправляє запити до таких модулів та шлюз, який транслює ці запити на послідовний інтерфейс. Так, у даному випадку модуль *RTU* з'єднаний зі шлюзом *E810-DTU* послідовним інтерфейсом *RS-485*. Усі запити, які надходять локальною мережею шлюз спрямовує на послідовний інтерфейс у прозорому режимі, а потім отриману відповідь від модуля *RTU* спрямовує у локальну мережу *Ethernet*. Тут шлюз виконує функцію сервера за протоколом *ModBus TCP* на порту «502». Функцію клієнта виконує ПЛК або ПК.

Отже, створіть у середовищі *CoDeSys* новий проєкт та визначте необхідний таргет-файл та мову реалізації *POU PLC_PRG*. Як цільову

платформу оберіть ПЛК *OBEH150-I.L*, а мову програмування виберіть *CFC*. Контролер буде лише опитувати підлеглий пристрій, тому вікно *POU PLC_PRG* залиште порожнім. Далі перейдіть до вкладення «*Ресурси*» та оберіть натисненням на ЛКМ утиліту «*Конфігурація ПЛК*». У ній, за допомогою натиснення на ПКМ викличте контекстне меню, додайте програмний модуль *ModBus (Master) [VAR]*.

Далі створіть віртуальну модель підлеглого пристрою – *Universal Modbus device [VAR]* та встановіть його *ModBus*-адресу (32). Налаштуйте роботу комунікаційного модуля за протоколом *TCP* (4-ий рядок у вкладенні з параметрами модуля) та *IP*-адресу шлюзу, яка визначена раніше у попередніх завданнях (172.16.28.225). Важливо, що формат *IP*-адреси у даному випадку як роздільник використовує символ «*;*», тобто адреса у першому рядку параметрів модуля буде такою – 172:16:28:225. Також змініть частоту опитування, наприклад, одно опитування за одну секунду. Інші параметри можна залишити без змін, як це зображено на рис. 2.10, за виключенням *IP*-адреси.

Далі створіть мережні змінні відповідно до ресурсів модуля *RTU*. Цей модуль побудований на базі мікроконтролера із послідовним інтерфейсом *RS-485*. Внутрішнє програмне забезпечення мікроконтролера виконує функцію керування чотирма дискретними виходами (електромеханічними реле). Структура його пам'яті за протоколом *ModBus* складається усього з чотирьох регістрів, по одному на кожен канал. У модулі встановлена адреса, яка має значення «32». Адреси регістрів такі: перший канал має адресу 0x0001, наступні канали мають, відповідно, адреси 0x0002, 0x0003 та 0x0004. Модуль підтримує лише виконання *ModBus*-функції з кодом 0x06, тобто запис значення до одиничного регістра. Керування станом виходів здійснюється за допомогою відправлення коду команди керування виходом до потрібного регістру, який відповідає визначеній функції. У табл. 3.1 наведені коди команд керування виходами модуля. Зазначимо, що потрібний код команди складається з двох байтів: старший вказує на функцію керування виходом, молодший – на час затримки її виконання. Код – це сукупне числове значення двох байтів, яке відповідає за розміром одному регістру.

Таблиця 3.1 – Функції дискретних виходів та коди команд керування

Команди керування виходами (функції керування)	Коди керування виходами	
	Код команди керування виходами	Значення параметру затримки (с)
OPEN	0x01	0x00
CLOSE	0x02	0x00
TOGGLE (<i>Self-locking</i>)	0x03	0x00
LATCH (<i>Inter-locking</i>)	0x04	0x00
MOMENTARY (<i>Non-locking</i>)	0x05	0x00
DELAY	0x06	0x00-0xff

Наведемо деякі пояснення щодо табл. 3.1. Так у першому стовпчику вказані функції, які може виконати модуль щодо керування дискретними виходами. Функції OPEN та CLOSE виконують просте вмикання та вимикання реле. Для вмикання реле без затримки потрібно до певного регістру записати необхідний код команди, наприклад, число 0x0100 (256₁₀). Відповідно, вимикання реле буде здійснюватись за кодом 0x0200 (512₁₀). Функції TOGGLE та LATCH перемикають стан реле після отримання чергового запиту на записування значення до регістру із внутрішнім або зовнішнім «підтягуванням». Для керування перемиканням стану реле без запам'ятовування (функція MOMENTARY) потрібно відправити код 0x0500 (1280₁₀). Це призводить до вмикання реле лише на одну секунду. Насамкінець, відправлення коду 0x060A (1546₁₀=1536+10) відповідає команді DELAY. У цьому випадку модуль активує реле на 10 с.

Таким чином, для керування модулем потрібно додати до головного пристрою у конфігурацію ПЛК модуль підлеглого пристрою (див. рис. 3.7). Далі додайте мережні змінні типу Register Output module[VAR] для відправлення до віддаленого модуля необхідних кодів команд для керування виходами. Для змінних вкажіть відповідні номери регістрів [2, 3].

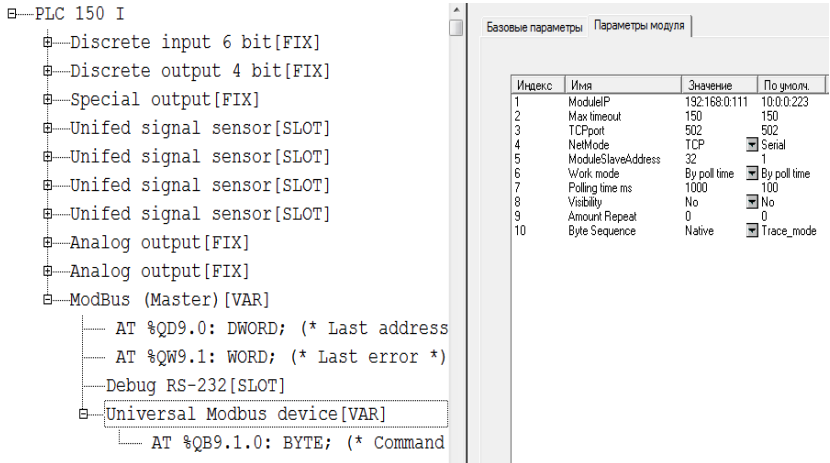


Рис. 3.7. Налаштування модуля підлеглого пристрою

Далі, скопіюйте проект та завантажте його код до ПЛК. Активуйте виконання програми користувача у ПЛК. Спостерігайте за станом змінних у режимі *Online*. Для зручного спостереження та керування виходами додайте до проекту візуалізацію для відображення значень стану виходів та запису кодів команд керування виходами. Приклад візуалізації зображений на рис. 3.8.

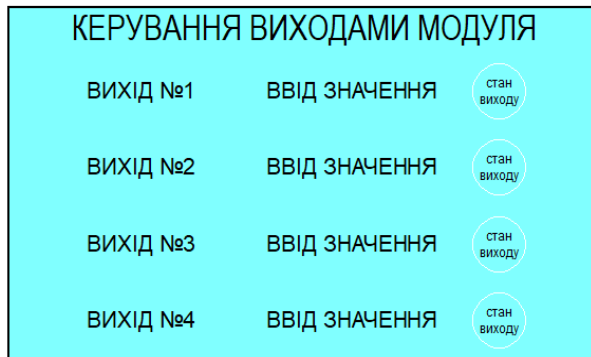


Рис. 3.8. Візуалізація для керування виходами модуля RTU

3.6 Завдання для самостійного виконання

Для самостійного виконання пропонується виконати таке завдання. Налаштувати опитування підлеглого ПЛК (*Slave*), у якого порт *RS232*

Debug з'єднаний із послідовним портом *RS232* шлюзу *E810-DTU*, з боку головного ПЛК (*Master*) за протоколом *ModBus TCP*. Для з'єднання підлеглого ПЛК та шлюзу необхідно використати кабель із першого завдання (див. п. 3.5.2). Необхідно отримати значення температури (*Pt100*) на другому аналоговому каналі підлеглого ПЛК. На головному контролері зробіть візуалізацію із відображенням отриманого значення.

3.7 Оформлення результатів проведення заняття

Складіть звіт про проведене заняття, де наведіть схеми комунікаційних з'єднань, умови обміну даними, режими роботи шлюзу. У звіті повинні бути наведені скріншоти з екрану монітора та демонстрацією результатів виконання завдань.

3.8 Контрольні запитання

- 1) Надайте коротку характеристику шлюзу *E810-DTU*.
- 2) Наведіть структурну схему шлюзу *E810-DTU*.
- 3) Які режими роботи можна налаштувати у шлюзі *E810-DTU*?
- 4) Які параметри потрібно налаштувати для прозорого обміну даними скрізь шлюз?
- 5) Наведіть приклади використання шлюзу *E810-DTU*.

Практичне завдання 4

ВИВЧЕННЯ ПРИНЦИПІВ ПОБУДОВИ ГЕТЕРОГЕННИХ РСУ З КЛІЄНТ-СЕРВЕРНОЮ АРХІТЕКТУРОЮ НА ОСНОВІ ВИКОРИСТАННЯ ШЛЮЗУ ETHERNET-SERIAL

4.1 Мета виконання практичного завдання

На виконання цього завдання у відповідності до робочої програми відводиться 2 ак. години аудиторного та 2 ак. години часу самостійної роботи студента (СРС).

Мета роботи:

- закріпити на практиці отримані теоретичні відомості про будову та роботу гетерогенних РСУ із клієнт-серверною архітектурою;

- ознайомитися з технічними характеристиками, функціоналом та можливостями комунікаційного шлюзу *Ethernet–Serial EKON134* [8];
- навчитися конфігурувати комунікаційний шлюз *EKON134* для сумісної роботи засобів РСУ із різними фізичними інтерфейсами без протоколу та за протоколом *ModBus RTU*;
- розробити та налаштувати РСУ для керування дискретними виходами модуля та для опитування модуля вимірювання температури і вологості з боку ПК за допомогою використання програми-симулятора роботи сенсорної панелі оператора *СПЗхх* [9].

4.2 Апаратне забезпечення практичного завдання

Виконання завдання можливе на будь-якому робочому місці із ПК та стендом, на якому встановлено шлюз. Комунікаційний шлюз *EKON134* змонтований на переносному стенді, на який додатково встановлено модуль *RTU*. Шлюз має чотири послідовних інтерфейси та мережний інтерфейс *Ethernet*. Модуль керування дискретними виходами з'єднується зі шлюзом інтерфейсом *RS-485* (порт P4). До порту P2 інтерфейсом *RS-232* підключений віддалений модуль для вимірювання температури та вологості за допомогою датчика *DHT11*. Для тестування роботи шлюзу у автономному режимі його послідовні порти P1 та P3 з'єднані нульмодемним кабелем із з'єднувачами *RJ45*. Шлюз з'єднаний з локальною мережею *Ethernet* (можливе з'єднання з ПК за допомогою *cross-over* кабелю).

Схема з'єднань ПЛК, віддалених модулів та мережного обладнання зображена на рис. 4.1.

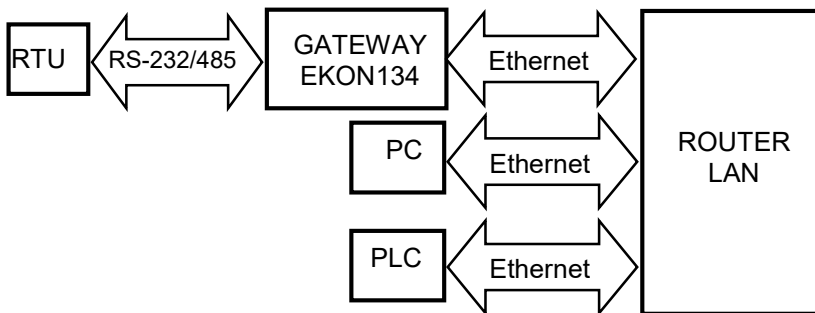


Рис. 4.1. Схема мережних з'єднань

4.3 Програмне забезпечення практичного завдання

Конфігурування шлюзу *ЕКОН134* здійснюється за допомогою вбудованого *WEB*-сервера по локальній мережі *Ethernet*. Для доступу до *WEB*-сервера потрібно у будь-якому браузері ввести *IP*-адресу (або мережне ім'я пристрою, це ім'я – <http://ekon134>), яка зазначена на пристрої, та ввести на його запит логін (*admin*). Пароль для доступу не потрібний. Це для випадку роботи шлюзу у режимі статичної *IP*-адреси. Якщо у мережі працює сервіс *DHCP* та у шлюзі встановлено цей режим, то *IP*-адресу можна визначити за допомогою програми-сканера локальної мережі *Advanced IP Scanner*. Ця програма здійснює сканування локальної мережі та надає інформацію про *IP*-адреси, *MAC*-адреси та стан хостів мережі. Для завантаження програма доступна за посиланням <https://www.advanced-ip-scanner.com/ru/>.

Сторінка *WEB*-сервера для налаштування шлюзу *ЕКОН134* зображена на рис. 4.2. На цій сторінці відкрите вкладення *Сеть* з основними налаштуваннями пристрою у мережі *Ethernet*.

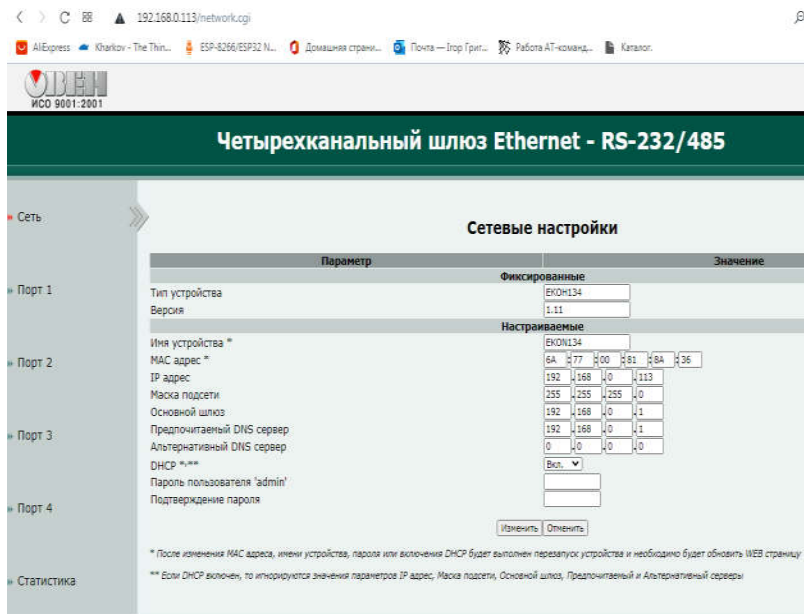


Рис. 4.2. Сторінка *WEB*-сервера для налаштування шлюзу *ЕКОН134*

Для доступу до послідовних інтерфейсів на шлюзі *ЕКОН134* використовується спеціальна програма для емуляції віртуальних портів «Конфигуратор виртуальных портов» (КВИП), яка доступна для завантаження за посиланням https://owen.ua/uploads/69/cvp_setup.zip із сайту www.owen.ua від компанії-виробника пристроїв *ВО ОВЕН (Україна)*. Вікно програми із основними налаштуваннями шлюзу зображено на рис. 4.3. Програма запущена на локальному ПК із *IP*-адресою 192.168.0.102 та працює через локальний порт 502. Причому, у пристрої активований режим *DHCP*. На рис. 4.3 зображено, що у шлюзі активовано два порти – *P2* та *P4*, для яких створено програмою віртуальні *COM*-порти *COM1* та *COM2*. Доступ до цих віртуальних портів здійснюється за *IP*-адресою 192.168.0.102 через мережний порт 502. Порти 50000, 50001, 50002 та 50003 є внутрішніми портами шлюзу, а порти *COM1* та *COM2* є віртуальними портами, які створені в ПК.

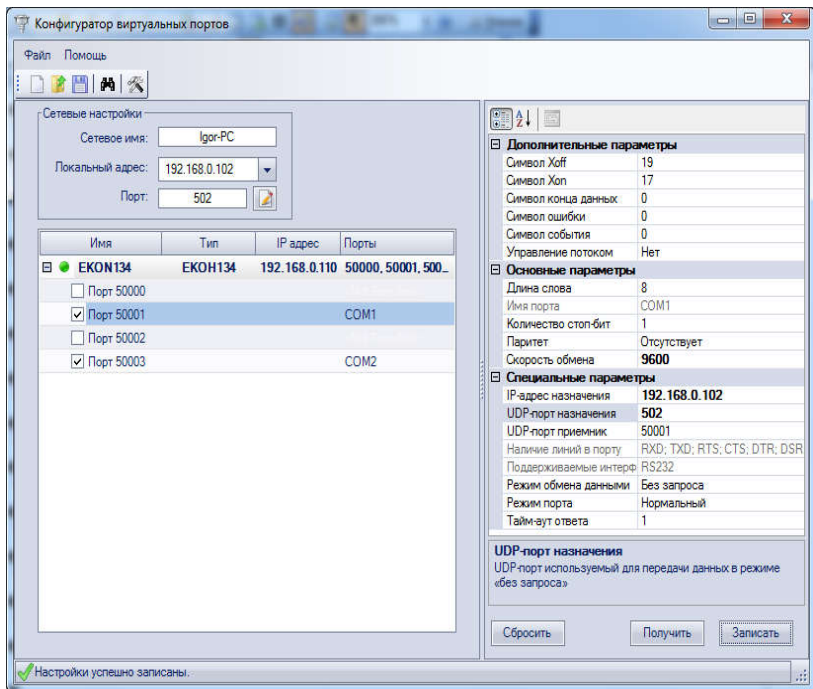


Рис. 4.3. Вікно програми «Конфигуратор виртуальных портов»

На рис. 4.3 зображено вікно налаштувань послідовного порту P4 для роботи по інтерфейсу *RS-485* на пристрої *ЕКОН134*, який має внутрішній порт 50003. Налаштування послідовного порту мають стандартні параметри – швидкість 9600, формат кадру «8-п-1». Додатково в налаштуваннях вказано *IP*-адресу та порт, до яких транслюються пакети, це 192.168.0.102:502. Інші налаштування відносяться до особливостей роботи пристрою *ЕКОН134*.

Для тестування шлюзу використовується програма прослуховування послідовних портів *COM Port Toolkit*.

Для опитування віддалених модулів, які підключені до послідовних портів шлюзу P2 та P4, використана програма конфігурування панелі оператора *СПЗхх* та емулятор роботи панелі оператора.

4.4 Короткі теоретичні відомості

Для інтегрування у локальну мережу пристроїв, які не мають інтерфейсу *Ethernet*, але мають послідовний інтерфейс (*RS-232/485*), використовують допоміжні пристрої. Це можуть бути або перетворювачі інтерфейсів, або, більш функціональні, шлюзи *Ethernet-Serial* з можливістю маршрутизації та підтримкою клієнт-серверного обміну. Пристрій *ЕКОН134* є шлюзом, який забезпечує трансляцію даних між чотирма послідовними інтерфейсами та інтерфейсом *Ethernet*. Пристрій оснащений одним інтерфейсом *Ethernet* та чотирма послідовними портами (порти P1 ... P4) [8].

Порти P1 та P4 пристрою можуть бути налаштовані для обміну даними по інтерфейсам *RS-232* або *RS-485*. Порти P2 та P3 підтримують роботу пристрою лише з інтерфейсом *RS-232*. Крім того, порт P3 призначений для проведення сервісних робіт зі шлюзом. Вибір режиму роботи портів та шлюзу визначається апаратно, за допомогою *DIP*-перемикачів, які встановлені на верхній панелі пристрою. Пристрій оснащений світлодіодною індикацією, яка сигналізує про проходження даних по послідовним інтерфейсам та по інтерфейсу *Ethernet*.

На рис. 4.4 зображена спрощена структурна схема пристрою. Більш докладно властивості та принцип дії шлюзу наведені у інструкції з експлуатації [8].

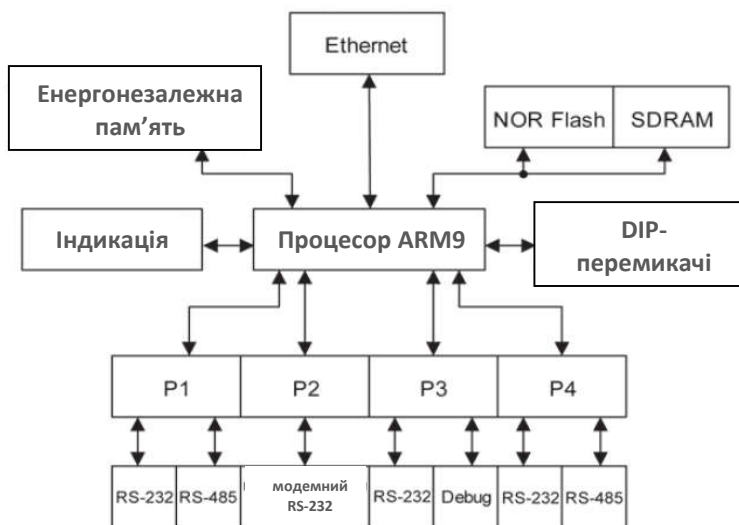


Рис. 4. 4. Спрощена структурна схема шлюзу *ЕКОИ134*

4.5 Послідовність виконання завдання

Виконання завдання складається з двох етапів.

Спочатку необхідно провести налаштування модуля та перевірити його роботу. Для цього необхідно з'єднати послідовні порти P1 та P3 пристрою нульмодемним кабелем. Далі необхідно налаштувати пристрій для передавання даних між зазначеними портами. Обмін даними буде спостерігатись за допомогою запущених двох екземплярів програми *COM Port Toolkit*, кожен з яких підключений до віртуального *COM*-порту, який створений програмою *КВП*. Повідомлення відправляються від одного екземпляра програми до іншого та, відповідно, теж відбувається у зворотному напрямку.

Далі шлюз налаштовується для обміну даними за протоколом *ModBus* між ПК та модулями *RTU*. Для взаємодії використовується емулятор сенсорної панелі оператора *СП300*, який запущений на ПК.

4.5.1 Налаштування шлюзу *ЕКОИ-134* для сумісної роботи з ПК

Запустить браузер для доступу до *WEB*-сервера пристрою *ЕКОИ-134* за ім'ям хосту <http://ekon134> або за *IP*-адресою, яка визначена за до-

помогою програми *Advanced IP Scanner*. На сторінці *WEB*-сервера пристрою проведіть необхідні налаштування відповідно до зображень на рис. 4.2. Далі необхідно налаштувати послідовні порти P1 та P3 для сумісного обміну даними. На рис. 4.5 зображено налаштування порту P1 у *WEB*-сервері. У цьому випадку параметри послідовного інтерфейсу мають такі значення – швидкість 115200, формат кадру «8-н-1».

Параметр	Значение
Основные параметры	
Скорость, бит/с	115200
Биты данных	8
Стоповые биты	1
Четность	Нет
Управление потоком	RTS/CTS
Дополнительные параметры	
Символ Xon (hex)	11
Символ Xoff (hex)	13
Специальные параметры	
Режим обмена данными	Без запроса
Тайм-аут ответа (мсек)	1
Интерфейс	RS-232
Режим порта	Нормальный
UDP-порт - приемник	50000
IP-адрес назначения	192.168.0.102
UDP-порт назначения	502

Изменить Отменить

Рис. 4.5. Налаштування порту P1

Подібно до рис. 4.5 необхідно налаштувати порт P3. Усі налаштування необхідно записати до конфігураційного файлу програми КВП за допомогою натиснення ЛКМ на кнопку *Изменить*.

Далі необхідно запустити програму *COM Port Toolkit* у двох екземплярах. На рис. 4.6 зображено вікно першого екземпляру програми *COM Port Toolkit*, яка підключена до першого *COM*-порту. Налаштування інтерфейсу зазначені у рядку статусу програми, який знаходиться знизу, під інформаційними полями для відправлення та приймання повідомлень.

Після проведених налаштувань проведіть тестування шлюзу. Для цього відправте будь-яке повідомлення від одного екземпляру програми до іншого, а також у зворотному напрямку. Налаштуйте одночасне

циклічне передавання з обох боків через шлюз (встановіть циклічність передавання – одне повідомлення за одну секунду). У програмі *КВП* спостерігайте за роботою віртуальних портів шлюзу. Після зупинення обміну визначте у *WEB*-сервері пристрою статистику щодо переданих даних. Для цього відкрийте вкладення *Статистика* у *WEB*-сервері. Зверніть увагу на збіжність кількості пакетів, які відправлені з порту **P1** та прийняті портом **P3**. Це говорить, що шлюз здійснював обмін даними без втрати пакетів. Насамкінець, проведіть скидання статистичних даних за допомогою відповідної кнопки на вкладенні. На завершення цього завдання відключіть віртуальні порти **P1** та **P3**, але попередньо закрийте обидва екземпляри програми *COM Port Toolkit*.

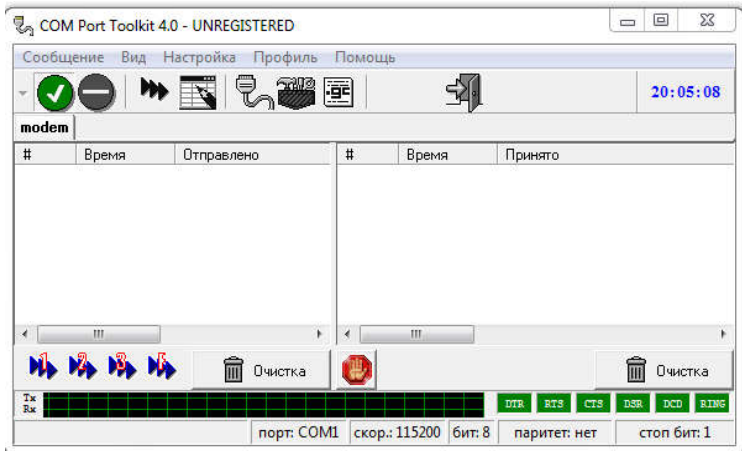


Рис. 4.6. Вікно програми *COM Port Toolkit*

4.5.2 Налаштування шлюзу *ЕКОН-134* для опитування віддалених модулів з боку ПК (емулятора панелі оператора *СПЗхх*)

Для виконання цього завдання необхідно провести налаштування інтерфейсів портів **P2** та **P4**. Зверніть увагу на положення *DIP*-перемикача, який перемикає режим роботи четвертого порту пристрою на обмін по інтерфейсу *RS-485*. Згідно інструкції [8] перемикач *DIP2* повинен бути у положенні «ON». Далі, у *WEB*-сервері проведіть налаштування параметрів обміну портів відповідно до параметрів обміну, які встановленні на віддалених модулях. Обидва модулі мають такі на-

лаштування: швидкість – 9600, формат кадру – «8-п-1». Модуль дискретних виходів має інтерфейс *RS-232*, тому він підключений до порту P2. Модуль для вимірювання температури та вологості має інтерфейс *RS-485*, тому він підключений до порту P4. На рис. 4.7 зображено налаштування порту P4 у *WEB*-сервері пристрою. Зверніть увагу, що фізичний порт мережі *Ethernet* для отримання пакетів, тобто ПК, має стандартне значення для протоколу *ModBus*, тобто, 502. Проведіть відповідне конфігурування віртуальних портів P2 та P4 у програмі *КВІІ*. Налаштування віртуального порту 50003, який відповідає фізичному інтерфейсу *RS-485* на порту P4 повинні бути подібними до налаштувань віддаленого модуля.

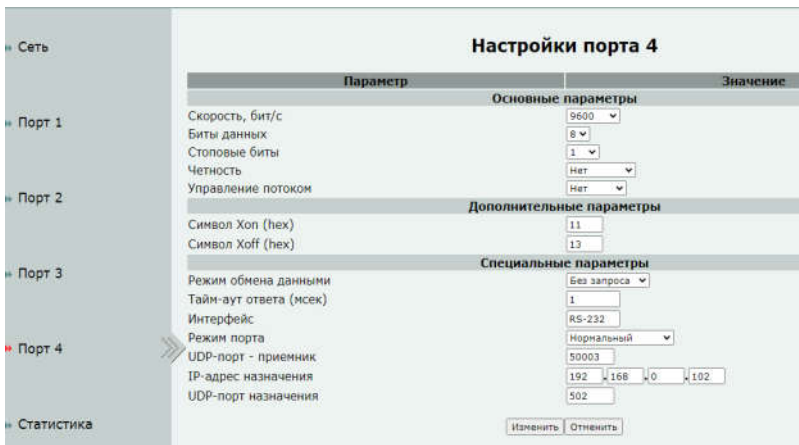


Рис. 4.7. Вікно з налаштуваннями порту P4 у *WEB*-сервері

4.5.3 Конфігурування панелі *СПЗхх* для зв'язку з віддаленими модулями за допомогою шлюзу *ЕКОН-134*

Запустіть програму конфігурування панелі оператора «*Конфигуратор СПЗхх*». Програма конфігурування призначена для налаштування панелі оператора *СПЗхх ОВЕН*. Програма працює під керуванням ОС *Windows* та дозволяє формувати і зберігати призначені для користувача екрани, які відображаються на дисплеї панелі у процесі її експлуатації та налаштовувати обмін між панеллю та модулями даними, які відображені у екранах користувача [9].

Панель оператора *СПЗхх ОВЕН* відноситься до пристроїв людино-машинного інтерфейсу та призначена для відображення та редагування значень параметрів, які отримані через послідовні інтерфейси від ПЛК або віддалених модулів. Конфігурація панелі є сукупністю значень параметрів, які визначають роботу пристрою. Проект конфігурації панелі зберігається у спеціальному файлі з розширенням *.twp.

Панель оператора призначена для виконання таких функцій:

- відображення стану об'єкта керування у режимі реального часу, з використанням графічних піктограм (індикатори, графіки, лінійки, умовні позначення обладнання, тощо);
- відображення інтерактивних (сенсорних) елементів, за допомогою яких оператор здійснює безпосереднє керування функціонуванням об'єкта та технологічним процесом;
- управління функціонуванням ПЛК та/або інших пристроїв;
- запис та читання значень регістрів ПЛК та/або інших пристроїв, до яких підключена панель.

Отже, у програмі конфігурування панелі створіть новий проект із режимом роботи панелі оператора як головного пристрою. У програмі конфігурування передбачено початкове покрокове налаштування створеного проекту за допомогою *Мастера настройки*. На першому екрані діалогу з майстром виберіть тип панелі – СП310 (або СП307), на другому екрані виберіть режими роботи панелі по обом її портам. Для обох портів виберіть режим роботи панелі, це режим *Modbus RTU Master*. Для кожного порту встановить параметри інтерфейсів. Це будуть такі параметри, які відповідають параметрам модулів: швидкість – 9600, формат кадру – «8-n-1». На третьому екрані введіть ім'я проекту та натисніть ЛКМ на кнопку ГОТОВО. В результаті з'явиться стандартне дерево проекту з одним порожнім екраном. Збережіть проект на жорсткому диску ПК. Тепер все готово для створення людино-машинного інтерфейсу, тобто екранів із інтерактивними графічними елементами та графічними об'єктами.

Робота з програмою полягає у створенні екранів, виборі графічних елементів та налаштуванні їх властивостей. Для цього використовуйте

довідкову систему програми для конфігурування панелі оператора СПЗхх. Далі створить екран з елементами управління та відображення. На рис.4.8 зображений варіант оформлення одного екрану панелі.

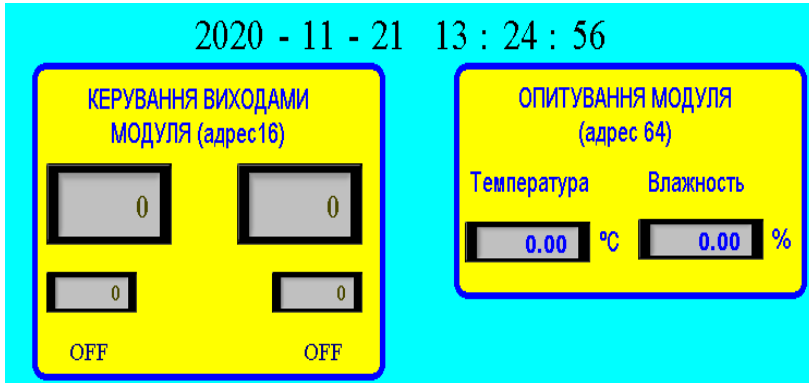


Рис. 4.8. Екран панелі оператора

На екрані зображені графічні елементи (ГЕ) та графічні об'єкти (ГО) різного функціонального призначення:

- поточні дата та час;
- графічні елементи «прямокутник із закругленими кутами», де знаходяться інші графічні елементи, які відносяться до модулів;
- статичний текст, який не пов'язаний з мережними елементами, наприклад, це інформаційні табло: «КЕРУВАННЯ ВИХОДАМИ МОДУЛЯ (адреса 16)» та «ОПИТУВАННЯ МОДУЛЯ (адреса 64)»;
- цифрові дисплеї для відображення поточних значень стану виходів модуля дискретних виходів та модуля вимірювання температури та вологості;
- дисплеї для введення цифрових значень для керування виходами модуля;
- індикатори для відображення стану виходів модуля дискретних виходів.

Аби не заглиблюватися в особливості редагування ГЕ та ГО з точки зору дизайну, зупинимося на принципах їхнього конфігурування з точки зору мережної взаємодії, тобто обміну даними між пристроями.

Основними параметрами для зв'язку графічного елементу із мережними даними є адреса пристрою у мережі, номер та тип регістра. Згідно до зображення на рис. 4.8 *ModBus*-адреса модуля дискретних виходів має значення «16», а адреса модуля вимірювання температури та вологості має значення «64».

Адреси регістрів у модулі дискретних виходів мають такі значення: вихідний канал №1 – 0x0001, вихідний канал №2 – 0x0002. Принцип дії модуля подібний до того, що наведено у п.3.5.3. Коди команд керування відповідають інформації, що наведена у табл. 3.1. Відмінність полягає у кількості дискретних виходів, тобто у кількості регістрів. Ще є відмінність у часі самостійного вимикання виходу за командою *MOMENTARY (Non-locking)*, який дорівнює 0,5 с. Можливе зчитування стану обох каналів у вигляді так званої маски виходів.

У модулі вимірювання температури та вологості є лише чотири регістри, у яких зберігаються вказані параметри у форматі числа із плаваючою комою одиничної точності. Це регістри з адресами від 0x0000 до 0x0003. Перші два регістри зберігають значення температури, наступні – значення вологості.

Розглянемо приклад налаштування відображення у цифровому вигляді значення регістра. Це буде регістр для зберігання значення температури. Налаштування вкладення *Регистр елемента ГЕ «Цифрової дисплей»* зображені на рис. 4.9. У вкладенні *Дисплей* необхідно встановити позначку у полі визначення формату відображення (*Float*) та вказати потрібну розрядність, наприклад загальну кількість знаків – 5, а кількість дробових знаків – 2.

Налаштування ГЕ *«Ввод данных»* подібно до налаштувань ГЕ *«Цифрової дисплей»*. Додатково можливо налаштувати опцію зчитування значення регістра, до якого записуються дані.

Для відображення стану виходів модуля можливе використання ГЕ *«Динамический текст»*. Цей ГЕ дозволяє відображати текст у залежності від значення регістра стану виходу. У випадку керування виходами будь-яке не нульове значення буде відобразитися текстом *«ON»*, у іншому випадку – буде текст *«OFF»*.

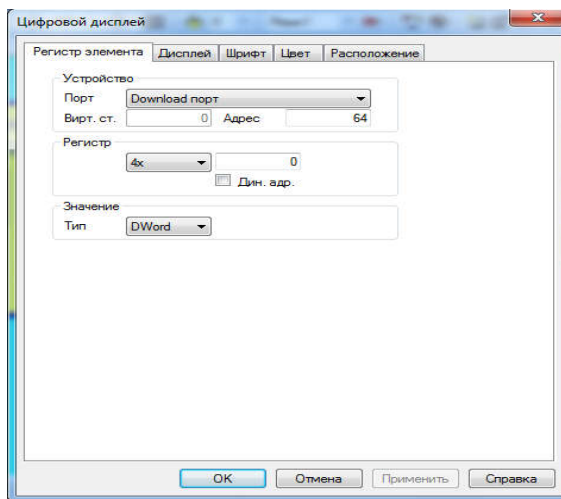


Рис. 4.9. Вікно налаштування ГЕ «Цифровой дисплей»

Конфігурування панелі оператора є допоміжним завданням для перевірки роботи шлюзу. Тому розглянуті лише необхідні ГЕ для перевірки роботи шлюзу з точки зору опитування регістрів у модулях.

Після закінчення створення усіх елементів екрану збережіть проєкт на жорсткому диску. Для перевірки працездатності макета РСУ використовуйте емулятор панелі у режим **ONLINE** (без наявності панелі). Для цього натисніть на відповідну піктограму у панелі інструментів програми-конфігуратора. Виберіть потрібні *COM*-порти ПК, які створені програмою *КВП*. Якщо на екрані панелі відображається повідомлення про відсутність зв'язку з пристроєм, перевірте номери та налаштування *COM*-портів у ПК. У разі вдалого підключення перевірте працездатність РСУ. Емулятор панелі одночасно опитує два модулі по різних інтерфейсам. За необхідністю використайте довідкову систему програми для конфігурування панелі оператора.

4.6 Завдання для самостійного виконання

Для самостійного виконання пропонується завдання, у якому шлюз використовується для опитування віддалених модулів з боку *ПЛК150* за протоколом *Modbus TCP*. У проєкті для ПЛК зробіть візуалізацію із відображенням значень параметрів одного модуля та можли-

вістю керувати виходами іншого модуля. Візуалізацію зробіть подібною до зображеної на рис. 4.8.

4.7 Оформлення результатів проведення заняття

Складіть звіт про проведене заняття, де наведіть схеми комунікаційних з'єднань, умови обміну даними, режими роботи шлюза. У звіті повинні бути наведені скріншоти з екрану монітора та демонстрацією результатів виконання завдань.

4.8 Контрольні запитання

- 1) Надайте коротку характеристику та наведіть функціональні можливості шлюзу *ЕКОН-134 ОВЕН*.
- 2) Наведіть структурну схему шлюзу *ЕКОН-134*.
- 3) Які режими роботи можна налаштувати у шлюзі *ЕКОН-134*?
- 4) Які параметри можливо додатково налаштувати для обміну даними скрізь шлюз *ЕКОН-134*?
- 5) Наведіть приклади використання шлюзу *ЕКОН-134*.

Практичне завдання №5

ВИКОРИСТАННЯ ПРОТОКОЛУ MQTT У РСУ ІЗ МОДЕЛЛЮ ОБМІНУ ДАНИМИ «ВИДАВЕЦЬ – ПЕРЕДПЛАТНИК»

5.1 Мета проведення практичного завдання

На виконання цього завдання у відповідності до робочої програми відводиться 2 ак. години аудиторного та 2 ак. години часу самостійної роботи студента (СРС).

Мета виконання завдання:

- закріпити на практиці отримані теоретичні відомості про основні принципи організації обміну даними між пристроями за протоколом *MQTT* [10] у РСУ (а також, у технологіях *IoT* та *IIoT*);
- ознайомитись з принципами налаштування протоколу *MQTT* у пристроях з його підтримкою без використання хмарних технологій;

- ознайомитись з порядком налаштування клієнта та брокера у системі обміну даними «М2М»;

- підключити за протоколом *MQTT* віддалений модуль для вимірювання температури та вологості до клієнта за допомогою брокера.

5.2 Апаратне забезпечення практичного завдання

Завдання може виконуватися на усіх робочих місцях з ПК. Джерелом інформації є модуль, який побудований на мікропроцесорному контролері серії *ESP8266*, до якого підключений датчик температури та вологості *DHT11*. Модуль з'єднаний із локальною мережею безпроводним зв'язком *WiFi* за допомогою *WiFi*-роутера.

5.3 Програмне забезпечення практичного завдання

Програмне забезпечення для виконання завдання складається з додатку, яке реалізує функцію брокера та додатку, яке реалізує функцію клієнта. Модуль, який опитує програма-клієнт, має внутрішнє програмне забезпечення, яке реалізує протокол *MQTT*, опитує датчик вимірювання температури та вологості та публікує виміряні параметри.

5.4 Короткі теоретичні відомості

Протокол *Message Queuing Telemetry Transport (MQTT)* [11] у останній час набув популярності завдяки використанню у технологіях *IoT* та *IIoT* для зв'язку як споживчих, так і промислових пристроїв у складі РСУ.

Протокол *MQTT* для обміну використовує модель обміну «Видавець – Передплатник» («*Publisher – Subscriber*»), яка дозволяє для мереж із обмеженою перепускною здатністю та нестабільним каналом зв'язку обмінюватися даними між двома пристроями. Його перевагою є невелике навантаження на ресурси процесора та простота використання. Крім того, протокол дуже зручно використовувати у автономних пристроях із обмеженим часом роботи. У таких умовах протокол є зручним транспортом для потокового обміну даними між пристроями у мережах з низькою перепускною спроможністю та/або з великим часом затримки у технології *IoT* та *IIoT*. Протокол працює як надбудова над стеком протоколів *TCP/IP*, причому для нього неважливо середовище

передавання даних. Це може бути класична мережа *Ethernet* або безпроводні канали зв'язку *Bluetooth* та *WiFi*.

Комунікаційна схема системи обміну даними, яка використовує протокол *MQTT*, складається із брокера та з одного або декількох клієнтів. Видавець не потребує ніяких налаштувань щодо кількості та розміщення передплатників, які отримують повідомлення. Крім того, передплатнику не потрібне налаштування на конкретного видавника. Усі функції перенаправлення повідомлень виконує програма-брокер. У системі може діяти декілька видавників, що розповсюджують повідомлення та декілька передплатників.

Основні особливості протоколу *MQTT*:

- це асинхронний протокол;
- повідомлення мають компактний розмір;
- працює в умовах нестабільного зв'язку на лінії передавання даних;
- підтримується декілька рівнів якості обслуговування (*QoS*);
- просте інтегрування нових пристроїв (масштабування).

Протокол *MQTT* працює на прикладному рівні поверх стеку протоколів *TCP/IP* та використовує за умовчанням порт 1883 (якщо підключення за протоколом *SSL*, то використовується порт 8883).

Обмін повідомленнями у протоколі *MQTT* здійснюється між клієнтом (*Client*), який може бути видавником або передплатником (*Publisher/Subscriber*) повідомлень, та брокером (*Broker*) повідомлень (наприклад, це програми *Domoticz MQTT* або *Mosquitto MQTT*).

Видавець спрямовує повідомлення до *MQTT*-брокера, в яких вказано топик (*Topic*). Передплатники можуть отримувати повідомлення від множини видавників в залежності від передплати на відповідні топики.

Пристрої за протоколом *MQTT* формують визначені типи повідомлень для взаємодії з брокером. Основні з типів повідомлень такі:

- *Connect* – встановити з'єднання з брокером;
- *Disconnect* – розірвати з'єднання з брокером;
- *Publish* – опублікувати дані в топик на брокері;

- *Subscribe* – передплатити підписку на топик у брокері;
- *Unsubscribe* – відмовитись від підписки від топика.

На рис. 5.1 зображено схему взаємодії між передплатником, видавцем та брокером.

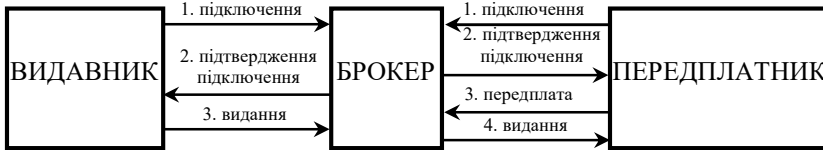


Рис. 5.1 .Архітектура клієнт-серверного обміну за протоколом *MQTT*

Існує регламентована семантика щодо оформлення топиків. Так, топіки це перелік символів із кодової таблиці *UTF-8*. Структура топиків має плоску ієрархічну структуру дерева, що спрощує їхню організацію та доступ до даних. Топіки можуть мати декілька рівнів, які розділені символом «/».

Приклад топика, в якому датчик температури знаходиться у приміщенні будівлі та публікує дані брокеру: `/home/living-space/living-room1/temperature`.

Передплатник може також отримувати повідомлення одночасно від декількох топиків за допомогою створення так званої *wildcard*. Ці карти також можуть бути однорівневими та багаторівневими.

Однорівнева карта *wildcard* використовує як роздільник символ «+». Наприклад, для отримання даних про температуру у всіх однотипних приміщеннях топик буде таким: `/home/living-space+/temperature`. В результаті передплатник отримає такі дані:

- `/home/living-space/living-room1/temperature;`
- `/home/living-space/living-room2/temperature;`
- `/home/living-space/living-room3/temperature.`

Для багаторівневої карти *wildcard* застосовується символ «#». Приклад використання такої карти – це отримання даних від різних датчиків в усіх приміщеннях будівлі: `/home/living-space/#`. В результаті передплатник отримає такі дані:

- /home/living-space/living-room1/temperature;
- /home/living-space/living-room1/light1;
- /home/living-space/living-room1/light2;
- /home/living-space/living-room1/humidity;
- /home/living-space/living-room2/temperature;
- /home/living-space/living-room2/light1.

Структура повідомлень у протоколі *MQTT* складається із трьох частин:

1. Фіксований заголовок (присутній у всіх повідомленнях).
2. Змінний заголовок (присутній лише у певних повідомленнях).
3. Дані, тобто «навантаження» (присутній лише у певних повідомленнях).

На рис. 5.2 зображено структуру фіксованого заголовка пакетів у протоколі *MQTT*.

Bit	7	6	5	4	3	2	1	0
Byte 1	Message Type				Flags specific to each MQTT packet			
Byte 2	Remaining Length							

Рис. 5.2. Структура фіксованого заголовка пакетів у протоколі *MQTT*

Поля фіксованого заголовка повідомлень мають такі призначення:

- **Message Type** – розміром чотири біти (від 7-го до 4-го першого байта), це тип повідомлення, наприклад: **CONNECT**, **SUBSCRIBE**, тощо;

- **Flags specific to each MQTT packet** – ці чотири біти (від 3-го до 0-го першого байта) призначені для допоміжних флагів, у залежності від типу повідомлення;

- **Remaining Length** – вказує на довжину поточного повідомлення (змінний заголовок + дані), може займати від одного до чотирьох байтів.

Усього у протоколі *MQTT* існує 15-ть типів повідомлень, перелік яких наведений у табл. 5.1. У таблиці прийняти скорочення: так «К», це клієнт, а «С» – це сервер.

Таблиця 5.1 – Типи повідомлень та їхні коди у протоколі *MQTT*

Тип повідомлення	Код	Напрямок слідування	Опис
RESERVED	0000 (0)	не діє	зарезервовано
CONNECT	0001 (1)	К→С	Запит клієнта на підключення до серверу
CONNACK	0010 (2)	К←С	Підтвердження успішного підключення
PUBLISH	0011 (3)	К←С, К→С	Публікування повідомлення
PUBACK	0100 (4)	К←С, К→С	Підтвердження публікування
PUBREC	0101 (5)	К←С, К→С	Публікування отримано
PUBREL	0110 (6)	К←С, К→С	Дозвіл на видалення повідомлення
PUBCOMP	0111 (7)	К←С, К→С	Публікування завершено
SUBSCRIBE	1000 (8)	К→С	Запит на передплату
SUBACK	1001 (9)	К←С	Запит на передплату прийнято
UNSUBSCRIBE	1010 (10)	К→С	Запит на відмову від передплати
UNSUBACK	1011 (11)	К←С	Запит на відмову прийнято
PINGREQ	1100 (12)	К→С	Запит типу PING
PINGRESP	1101 (13)	К←С	Відповідь типу PING
DISCONNECT	1110 (14)	К→С	Повідомлення про відключення від сервера
RESERVED	1111 (15)	не діє	зарезервовано

Наступне поле першого байта (див. рис. 5.2) складається з чотирьох флагів. Це такі флаги: DUP, QoS, QoS та Retain:

- DUP – цей флаг встановлюється, якщо клієнт або брокер здійснюють повторне відправлення пакету. Застосовується у повідомленнях типів PUBLISH, SUBSCRIBE, UNSUBSCRIBE, PUBREL. Якщо флаг

встановлений, то змінний заголовок повинен мати **Message ID** (ідентифікатор повідомлення);

- **QoS** – якість обслуговування, може приймати такі значення – 0, 1 або 2;

- **Retain** – у процесі публікації повідомлення зі встановленим флагом, брокер його збереже.

У змінному заголовку повідомлення наведені такі відомості про пакет:

- **Packet identifier** – ідентифікатор пакета, який є у всіх повідомленнях, за виключенням: **CONNECT**, **CONNACK**, **PUBLISH** (з $QoS < 1$), **PINGREQ**, **PINGRESP**, **DISCONNECT**;

- **Protocol name** – назва протоколу (лише у повідомленнях типу **CONNECT**);

- **Protocol version** – версія протоколу (лише у повідомленнях типу **CONNECT**);

- **Connect flags** – флаги, які вказують на поведінку клієнта під час з'єднання.

Флаги, які вказують на поведінку під час з'єднання розподілені у восьмому байті у такому порядку (від старшого до молодшого бітів):

- **User name** – за наявністю цього флагу у «навантаженні» має бути ім'я користувача для аутентифікації клієнта;

- **Password** – за наявністю цього флагу у «навантаженні» має бути вказаний пароль для аутентифікації клієнта;

- **Will Retain** – якщо цей флаг встановлений (1), то брокер зберігає так зване **Will Message**;

- **Will QoS** – якість обслуговування для **Will Message**, якщо він встановлений флаги **Will Flag**, **Will QoS** та **Will retain** є обов'язковими;

- **Will Flag** – якщо цей флаг встановлений, то після відключення клієнта від брокера без відправлення повідомлення **DISCONNECT** (це можливе у випадку випадкового роз'єднання, тощо), брокер надішле повідомлення усім підключеним клієнтам за допомогою **Will Message**;

- **Clean Session** – очищення сесії. Якщо флаг не встановлений

(0), то брокер збереже сесію, усі передплати клієнта, а також відправить клієнту усі повідомлення з якістю QoS1 та QoS2, які були отримані брокером під час роз'єднання з клієнтом, у випадку повторного з'єднання. Відповідно, якщо флаг встановлений (1), то під час повторного підключення клієнта йому буде необхідно знову оформити передплату на топіки.

Таблиця 5.2 – Коди повідомлень для пакету CONNECT

Код	Повернутий код	Пояснення щодо коду
0	0x00 Connection Accepted	З'єднання прийнято
1	0x01 Connection Refused, unacceptable protocol version	Брокер не підтримує версію протоколу, яку використовує клієнт
2	0x02 Connection Refused, identifier rejected	У переліку відсутній Client ID, якому дозволено підключення
3	0x03 Connection Refused, Server unavailable	З'єднання встановлено, але сервіс <i>MQTT</i> не доступний
4	0x04 Connection Refused, bad user name or password	Невірний логін або пароль
5	0x04 Connection Refused, not authorized	Доступ до з'єднання заборонений

Флаг **Session Present** застосовується у повідомленні із типом **CONNACK**. Якщо брокер приймає підключення зі встановленим (1) флагом **Clean Session**, то він призначає статус не встановлено (0) для біту **Session Present**. Якщо брокер приймає підключення з невстановленим (0) флагом **Clean Session**, то статус біта **Session Prese** залежить від того, зберігав брокер раніше сесію з цим клієнтом. Так, якщо зберігав, то біт **Session Prese** буде встановлений (1), якщо брокер попередню сесію не зберіг, то біт **Session Prese** буде скинутий (0). Тобто, параметр дозволяє клієнту визначити статус попередньої сесії у брокера.

Флаг **Connect Return code** потрібен для визначення ситуації, коли брокер за будь-яких причин не може прийняти правильно сформоване повідомлення **CONNECT** від клієнта. У цьому випадку у повідом-

ленні CONNACK у другому байті буде встановлено потрібний код згідно з табл. 5.2.

Завершає повідомлення дані, тобто «навантаження». Зміст та формат даних, що передаються у повідомленнях визначаються у додатку. Розмір даних може бути розрахований шляхом віднімання із параметра Remaining Length довжини змінного заголовка.

Щодо якості сервісу у протоколі *MQTT* (QoS) регламентовано три рівня. Перший рівень якості сервісу це QoS 0 «*At most once*». На цьому рівні видавець відправляє повідомлення брокеру і не чекає від нього підтвердження, тобто «відправив та забув».

На рис. 5.3 зображена схема реалізації першого рівня якості сервісу QoS 1 «*At most once*».



Рис. 5.3. Схема реалізації першого рівня якості сервісу

Другий рівень якості сервісу це QoS 1 «*At least once*» (див. рис. 5.4). Цей рівень гарантує, що повідомлення буде обов'язково доставлено брокеру, але існує ймовірність дублювання повідомлень від видавця. Після отримання дубліката повідомлення, брокер знову робить його розсилання передплатникам. Крім того, брокер надсилає підтвердження про отримання повідомлення. Якщо видавець не отримав повідомлення типу PUBACK (див. табл.6.1) від брокера, він вдруге відправляє пакет, при цьому встановлюється (1) флаг DUP.

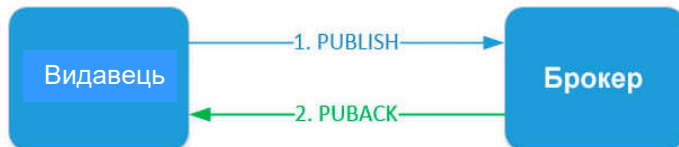


Рис. 5.4. Схема реалізації другого рівня якості сервісу

Третій рівень якості сервісу це QoS 2 «*Exactly once*» (див. рис.5.5). На цьому рівні гарантується доставляння повідомлень передплатнику та виключається можливе дублювання вже відправлених повідомлень. Алгоритм забезпечення третього рівня якості сервісу буде таким. Видавець відправляє повідомлення брокеру. У цьому вказується унікальний ідентифікатор пакету Packet ID, QoS=2 та DUP=0. Видавець зберігає це повідомлення не підтвердженням доки, доти не отримує від брокера повідомлення PUBREC (див. табл.6.1). Брокер відповідає цим повідомленням, якщо у повідомленні Packet ID подібний до отриманого від видавника. Після отримання повідомлення PUBREC видавець відправляє повідомлення PUBREL з подібним Packet ID. До моменту отримання цього повідомлення брокер зберігає його копію. Після отримання повідомлення PUBREL брокер видаляє копію повідомлення та відправляє повідомлення про закінчення транзакції PUBCOMP.

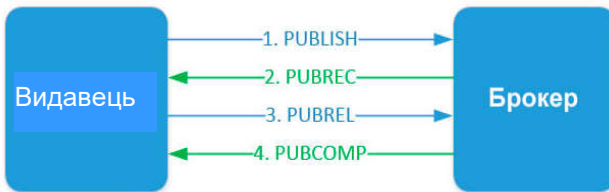


Рис. 5.5. Схема реалізації третього рівня якості сервісу

У протоколі *MQTT* реалізовано такі методи захисту інформації:

- аутентифікація клієнтів. У цьому методі пакет **CONNECT** може мати поля **USERNAME** и **PASSWORD**;
- контроль доступу за допомогою **Client ID**;
- підключення до брокера за протоколом *TLS/SSL*.

Приклади обладнання для *IIoT* та *IoT*, де може бути застосований протокол *MQTT*:

- контролери, які програмують за допомогою *WEB*-сервіса;
- модулі віддаленого введення та виведення;
- шлюзи;

- платформи та сервери;
- вимірювачі параметрів.

Як програму-клієнт для практичного завдання пропонується використати програму *MQTTBox*, яку можна завантажити за посиланням: <https://www.microsoft.com/en-us/p/mqttbox/9nblggh55jzg?activetab=pivot:overviewtab> .

Як програму-брокер рекомендовано використати програму *Mosquitto*, яку можна завантажити за посиланням: <https://mosquitto.org/files/binary/win64/mosquitto-2.0.8-install-windows-x64.exe> .

Для тестування роботи пристрою пропонується використати програму *MQTT-Explorer*, яку можна завантажити за посиланням: <https://www.microsoft.com/uk-ua/p/mqtt-explorer/9pp8sfm082wd> .

5.5 Послідовність виконання завдання

Виконання завдання складається з декількох етапів. Спочатку потрібно налаштувати пристрій (*Видавець*) для роботи за протоколом *MQTT*. Далі потрібно налаштувати брокер (програма *Mosquitto*), яка може бути встановлена як сервіс на ПК. Насамкінець необхідно налаштувати клієнта (*Передплатник*). Це програма *MQTTBox*, яка може бути встановлена як додаток на ПК або на мобільний телефон. Додаток для телефону можна завантажити з *Internet*. Але це необов'язково, тому що програма *MQTTBox* може виконувати функцію локального клієнту для перевірки з'єднання з видавцем.

5.5.1 Налаштування пристрою, який виконує функцію видавця

Функцію видавця буде виконувати модуль, який побудований на мікроконтролері *ESP-01 (ESP-8266)* із *WIFI*-модулем. Внутрішнє програмне забезпечення мікроконтролера виконує функцію опитування комбінованого датчику типу *DHT11*, тобто модуль виконує функцію вимірювача температури та вологості. Цей модуль, якщо він підключений до локальної мережі потрібно знайти, тобто визначити його *IP*-адресу за допомогою програми *Advanced IP Scanner*. Програма раніше вже була використана під час виконання попереднього практичного завдання.

Отже, якщо *IP*-адреса віддаленого модуля визначена, то можна за допомогою браузера відкрити його *WEB*-сторінку. На рис. 5.6 зображена головна сторінка *WEB*-сервера (вкладення *Main*), на якій вказані основні мережні налаштування модуля та параметри мікроконтролера.

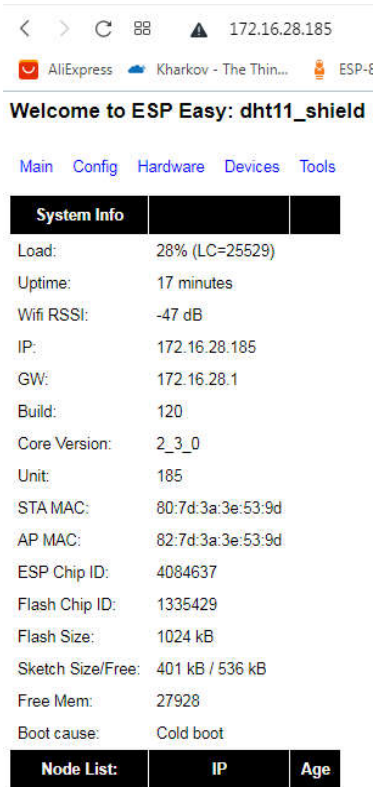


Рис. 5.6. Головна сторінка *WEB*-сервера модуля

а порт 1883 (який закріплений за протоколом *MQTT*). Також потрібно вказати ім'я з'єднання, але пароль не встановлюйте. Інші параметри залиште без змін. Натисканням ЛКМ на кнопку *Submit* підтвердить налаштування, що призведе до їх запам'ятовування та перезавантаження модуля. Далі потрібно вказати топіки протоколу *MQTT*. Ці налаштування знаходяться у іншому вкладенні (*Tools*). Так у вкладенні *Tools* у

Для налаштування модуля для роботи за протоколом *MQTT* потрібно перейти до вкладення *Config*. У цьому вкладенні усі налаштування потрібно залишити без змін, крім параметра *Protocol*. Якщо натиснути ЛКМ на кнопку із символом «?», то відкриється вікно у браузері із поясненнями щодо налаштування протоколу *MQTT* у пристрої. Для встановлення потрібного протоколу ЛКМ натисніть на поле із позначкою «▼». У переліку, що випадає, потрібно обрати варіант *Domoticz MQTT*.

Далі потрібно вказати *IP*-адресу мережного пристрою, де встановлено брокер протоколу *MQTT*. Це може бути або контролер, або ПК. Введіть *IP*-адресу та порт на ПК, з якого здійснюється доступ до даних модуля. Наприклад, це буде адреса 172.16.28.100,

рядку **System** потрібно ЛКМ активувати опцію **Advanced**. Потім потрібно занести до полів **Subscribe Template** та **Publish Template** топіки. Наприклад, для передплатника це буде топик **domoticz/out**, відповідно для видавця це буде топик **domoticz/in**.

Насамкінець, необхідно визначити ідентифікатор параметра (**IDX**), що передається, та активувати опцію передавання даних. Для цього потрібно перейти до вкладення **Devices**. У вкладенні визначте значення ідентифікатора **IDX/Var** та поставте позначку напроти рядка **Send Data**. Підтвердить активування передавання даних натисканням ЛКМ на кнопку **Submit**.

5.5.2 Налаштування клієнта для отримання повідомлень за протоколом MQTT

Після завантаження програми **MQTTBox** із сайту її встановлення на ПК не потрібно. Достатньо лише запустити її виконавчий файл або ярлик на робочому столі ОС **Windows**. Взагалі, після скачування програма розміщується у директорії зі шляхом доступу: **C:\Users\Igor\AppData\Local\Programs\MQTTBox\0.2.1\MQTTBox.exe**.

Після запуску програми відкриється її робоче вікно, як це зображене на рис. 5.7.

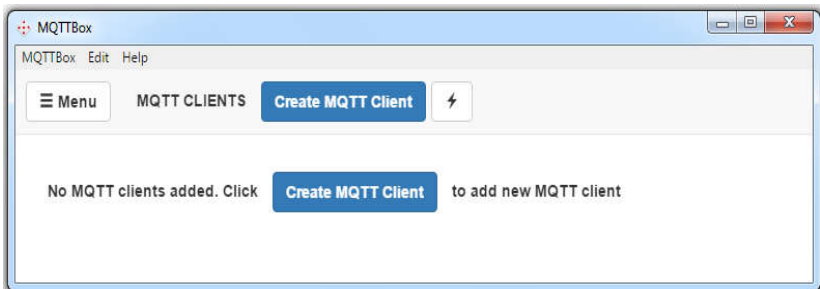


Рис. 5.7. Стартове вікно програми **MQTTBox**

На рис. 5.7 у робочому вікні програми зазначено, що у програмі немає налаштованих клієнтів. Також у робочому вікні програми є активна кнопка для створення клієнтів (**Create MQTT Client**). Програма пропонує додати клієнта за допомогою відповідної кнопки. Після натискання ЛКМ на цю кнопку з'явиться вікно, яке зображене на рис. 5.8.

На зображенні, що на рис. 5.8, представлені поля для налаштувань (ліворуч) та поля для позначок (праворуч), які встановлені (крім позначки *Will Retain*) за умовчанням. Заповніть поля бланка для налаштувань відповідно до зображення на рис. 5.9. Підтвердіть введенні параметри налаштувань клієнта натисненням ЛКМ на кнопку *Save*.

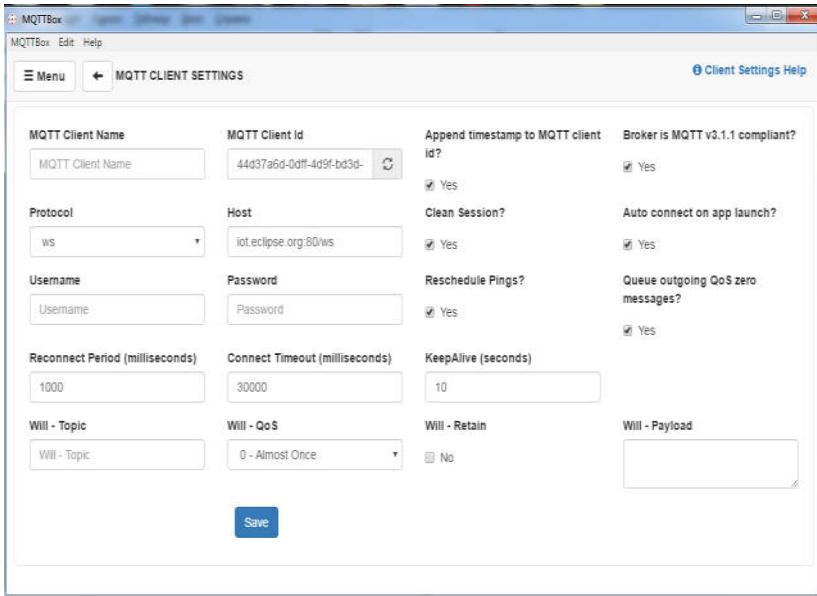


Рис. 5.8. Вікно для налаштувань клієнта за протоколом *MQTT*

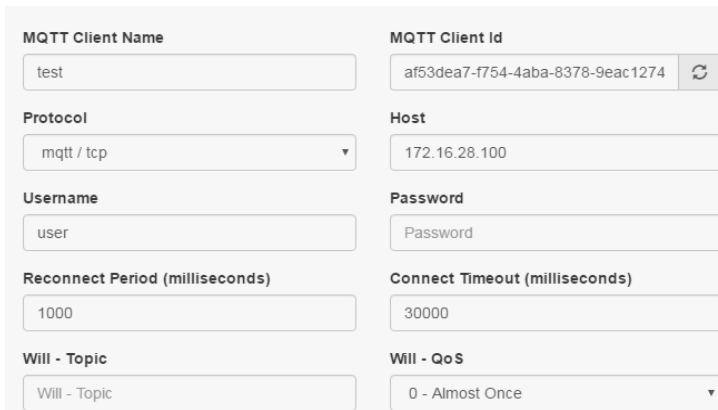


Рис. 5.9. Налаштування клієнта у програмі *MQTTBox*

Бланк налаштувань клієнта, який зображений на рис. 5.9, містить усі складові повідомлень, тобто ім'я клієнта, ім'я користувача, IP-адресу хоста (тобто, ПК), тощо. Усі флаги, які є у повідомленні за протоколом *MQTT* встановлені за умовчанням. Також за умовчанням встановлено рівень якості сервісу, це *QoS 0 «Almost once»*. За потребою можна налаштувати параметри *Will*-карти, але у цьому завданні це робити не потрібно.

Якщо, усі налаштування зроблені правильно, то програма-брокер автоматично з'єднається з клієнтом, що буде відображено повідомленням *Connected* на зеленому фоні, як це зображено на рис.5.10. Також у меню програми-брокера з'являться кнопки *Add Publisher* та *Add Subscriber* для створення вікон для відображення журналу протоколу видавця та передплатника з ім'ям топіків: *Topic to Publish* (вікно має контур синього кольору) – *domoticz/out* та *Topic to Subscriber* (вікно має контур жовтого кольору) – *domoticz/in*.

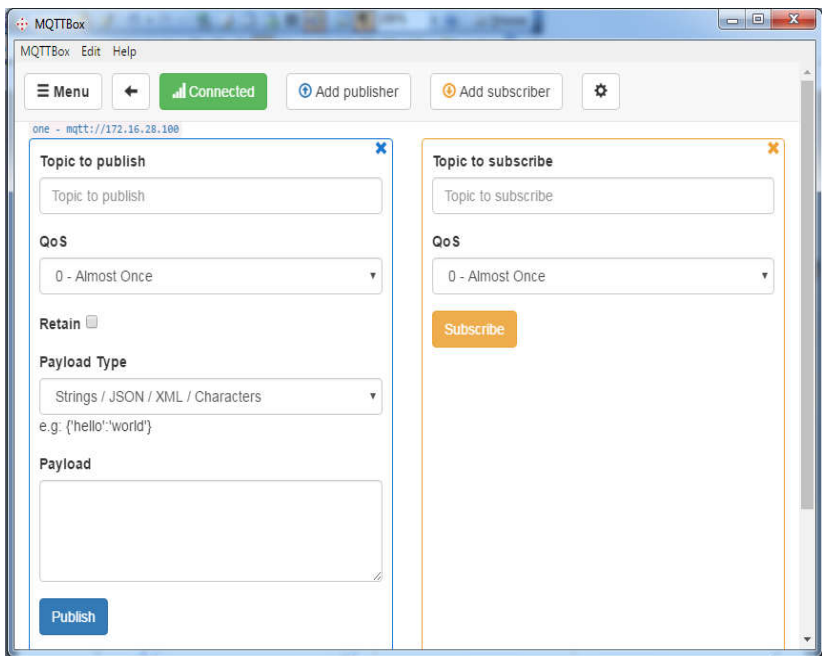


Рис. 5.10. Робочий простір клієнта *MQTTBox* із топіками

Для завершення налаштувань потрібно внести вказані топіки у відповідні поля. Зауважимо, що віддалений модуль не може опрацьовувати вхідні повідомлення, тобто, виходячи із вбудованого програмного забезпечення, він може бути лише видавником. Якщо функцію передплатника модуль виконувати не може, тому топік для видання можна залишити порожнім. Для обох топіків встановлено найпростіший рівень якості сервісу, це QoS 0 «Almost once». Після завершення налаштувань активувати роботу брокера можна натисканням на відповідні кнопки всередині вікон передплатника (Subscribe) та видавця (Publish). Для зупинення роботи брокера потрібно також натиснути на відповідні кнопки.

На рис. 5.11 зображені вікна видавця та передплатника активованого брокера із отриманими повідомленнями.

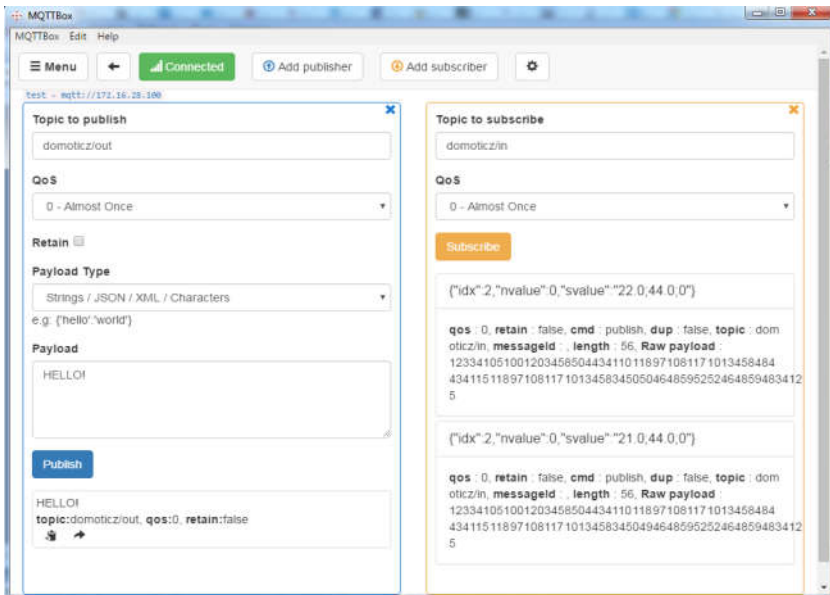


Рис. 5.11. Робочий простір активованого клієнта *MQTTBox*

На рис. 5.11 у вікні передплатника у повідомленні вказано такі параметри:

- ім'я клієнта – test;

- ім'я топіка – *domoticz/in*;
- *idx* – ідентифікатор змінної (2), дорівнює значенню, яке вказано у налаштуваннях віддаленого модуля, див. п. 5.5.1;
- так зване «навантаження», тобто, корисна інформація. Це значення параметрів, які отримує програма-брокер;
- у полі повідомлення вказані параметри роботи протоколу *MQTT* та власне код повідомлення (опис див. у п. 5.4).

Якщо дані надходять до брокера та клієнта-передплатника, то на цьому етапі можна вважати практичне завдання виконаним.

Зауважимо, що програма *MQTTBox* призначена для тестування пристроїв, які підтримують протокол *MQTT*. Тому вона має досить обмежений функціонал. Але її використання дозволяє ознайомитися із основами протоколу та тестувати віддалені пристрої. Крім того, програма може імітувати роботу видавця, тобто з нею може з'єднатись клієнт-передплатник. Опис роботи програми *MQTTBox* можна отримати за посиланням: <http://workswithweb.com/html/mqttbox/getstarted.html>.

5.6 Завдання для самостійного виконання

Для самостійного опрацювання пропонується налаштувати програму *MQTTBox* для підключення до віддаленого модуля. У джерелі [12] наведені відомості щодо налаштування модуля та підключення до нього клієнтів, наприклад, додатка *Domoticz*. Взагалі, модуль має дуже розвинуте внутрішнє програмне забезпечення. Він може виконувати свої функції за декількома протоколами та із використанням різних інтерфейсів. Так, крім протоколу *MQTT*, у модулі підтримані протокол *Modbus* та шина *CAN* з використанням не тільки *Ethernet*, але з використанням безпроводних каналів зв'язку (*WIFI*) та послідовних інтерфейсів (*RS485* та *RS232*). Модуль керує виходами із підтримкою різних режимів: просте вмикання/вимикання, затримку вмикання/вимикання та короткочасне (0.5 с) вмикання із наступним вимиканням. В модулі передбачене локальне керування виходами за допомогою звичайних кнопок без фіксування.

Для конфігурування модуля використовується *WEB*-сервер. Для

підключення до *WEB*-сервера необхідно у будь-якому браузері ввести *IP*-адресу модуля, яку потрібно попередньо визначити. Якщо пристрій використаний вперше, то його *IP*-адреса має таке значення – 192.168.1.100. Логін та пароль – admin. При цьому в модулі вимкнена функція *DHCP*. Але, якщо модуль було раніше налаштовано, то його адресу можливо визначити за допомогою програми-сканера локальної мережі *Advanced IP Scanner* (див. п. 4.3, стор. 51).

Якщо адреса пристрою визначена, то після її введення у браузері відкриється вікно із запитом логіна та пароля, а далі головна сторінка *WEB*-сервера, яка зображена на рис. 5.12.

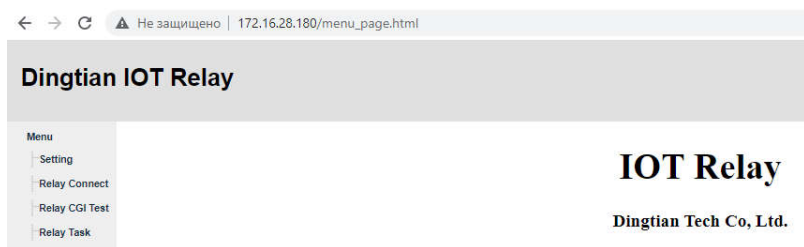


Рис. 5.12. Головна сторінка *WEB*-сервера модуля

Мережні налаштування модуля наведені у вкладенні *Setting*. Відповідне зображення наведене на рис. 5.13.

Setting

Hardware Version	V1.4-AT
Software Version	V2.17.79
Build Date	2021-02-19 20:04:36
Model	Dingtian IOT RELAY-2
Serial Number	2413
Date Time	01.01.1970, 08:21:52
NTP Server	pool.ntp.org
Hostname	Dingtian-Relay2413
Hostname+Suffix	Dingtian-Relay <input type="text"/> <input type="button" value="+"/> <input type="button" value="SN"/> <input type="button" value="v"/>
HTTP Server Port	80
DHCP	<input type="button" value="No"/> <input type="button" value="v"/>
IP	172.16.28.180
Netmask	255.255.255.0
Gateway	172.16.28.253
DNS	172.16.28.253
MAC	bc:34:88:00:08:be
WIFI AP IP	0.0.0.0
WIFI STA IP	0.0.0.0

Рис. 5.13. Вкладення *Setting* *WEB*-сервера модуля

У вкладенні **Setting** вказаний серійний номер (Serial Number), який має значення «2413». Цей номер потрібен при формуванні топіків та запитів на публікування та передплату.

Для підключення до модуля потрібно відкрити вкладення **Relay Connect**. Зображення із відкритим вкладенням показано на рис. 5.14.

Relay

Channel	Protocol	Addr	Baud	Databits	Stopbits	Parity
RS485	Modbus-RTU	1	115200bps	8bit	1bit	None
CAN	Dinglian String	ID	Speed			
		1	125Kbps			
ETH-UDP1	Dinglian Binary	Remote Address		Remote Port	Local Port	
		192.168.1.9		60000	60000	
ETH-UDP2	Dinglian String	Remote Address		Remote Port	Local Port	
		192.168.1.9		60001	60001	
ETH-TCP Server	Modbus-TCP				Local Port	
					502	
ETH-TCP Client	Modbus-RTU Over TCP	Remote Address		Remote Port		
		192.168.1.9		502		
ETH-MQTT	MQTT	Broker Address		Broker Port	Broker Username	Broker Password
		172.16.28.100		1883	mqtt	123

Other	
Relay Password	0 0-9999(0 no password)
Keep Alive Second	30 1-120 second(0 close)
Jogging Time	10 1-255 (1=100ms)
Power Failure Recovery Relay	No
Input Control Relay	Yes

Button Type	
Momentary	Momentary

Save

Рис. 5.14. Вкладення **Relay Connect** *WEB*-сервера модуля

У вкладенні **Relay Connect** важливою є інформація щодо налаштувань клієнта (відмічено контуром на рис. 5.14). Це стосується рядка **ETH-MQTT**, де вказано *IP*-адресу та порт брокера (172.16.28.100:1883), ім'я брокера (mqtt) та пароль для підключення (123). Брокер встановлено на комп'ютері із зазначеною адресою. Потрібно цю адресу вказати при створенні клієнта у програмі *MQTTBox* на будь-якому комп'ютері. Налаштування клієнта будуть подібними до зображення на рис. 5.9, але із відповідними зміненнями (див. рис. 5.14).

Далі, до клієнта необхідно додати топіки для видання та передплати, як це показано на рис. 5.10. Але, щодо топіків та повідомлень, необхідно звернутися до опису пристрою, яке наведено у джерелі [12].

Топіки для публікування стану та інших параметрів пристрою будуть відправлятися із періодом 30 с (приклад наведений для пристрою із серійним номером «100»).

Публікування стану реле та дискретних входів відображається у таких топіках (з різним відображенням):

- /dingtian/relay100/out/r1~8 – можливі значення: ON або OFF;
- /dingtian/relay100/out/i1~8 – можливі значення: ON або OFF;
- /dingtian/relay100/out/relay1~8 – можливі значення: {"idx": "1", "status": "OFF"};
- /dingtian/relay100/out/input1~8 – можливі значення: {"idx": "1", "status": "HIGH"} або {"idx": "1", "status": "LOW"}.

Пристрій також публікує значення інших параметрів: IP-адресу, серійний номер пристрою, MAC-адресу, кількість реле та дискретних входів на борту пристрою. Це буде відповідати таким топікам:

- /dingtian/relay100/out/ip, значення: 192.168.1.100;
- /dingtian/relay100/out/sn, значення: 100;
- /dingtian/relay100/out/mac, значення: bc:34:88:00:00:ff;
- /dingtian/relay100/out/input_cnt, значення: 4;
- /dingtian/relay100/out/relay_cnt, значення: 4.

Якщо потрібно отримати усю перелічену вище інформацію, то у топіку потрібно використати спеціальний символ для багаторівневої карти *wildcard* – «#». Топік буде таким: /dingtian/relay100/out/# .

Для отримання виданих даних від пристрою у програмі *MQTTBox* потрібно зробити передплату на відповідні топіки. Тобто, по відношенню до програми топіки пристрою будуть вихідними.

У пристрої налаштовано передплатника на отримання команд керування від іншого клієнта (програми *MQTTBox*). Типи команд керування відносяться до параметру «type». Це такі команди: ON/OFF, DELAY та JOGGING. Для цього передбачені такі топіки:

- /dingtian/relay100/in/control – можливі значення: {"type": "ON/OFF", "idx": "1", "status": "ON", "time": "0", "pass": "0"};
- /dingtian/relay100/in/r1~8 – можливі значення: ON або OFF.

Для першого топіку можливе відправлення таких повідомлень (вмикання обох реле на 20 с):

- {"type":"delay","idx":'1',"status":"on","time":"20","pass":"0"} ;
- {"type":"delay","idx":'2',"status":"on","time":"20","pass":"0"} .

На рис. 5.15 та 5.16 зображені вікна із створеними у програмі *MQTTBox* видавцем та передплатником. У цих вікнах на топик з керування виходами пристрою (*/dingtian/relay100/in/control*) відправлені повідомлення на їхнє вмикання на 20 с. Причому, на цьому ж топіку відображено код повідомлення у строковому (рис. 5.15, поз. 1) та цифровому (рис. 5.15, поз. 2) вигляді. На рис. 5.16 зображено фрагмент (*MAC-адреса*) публікування повідомлень від пристрою за топіком */dingtian/relay100/out/#*.

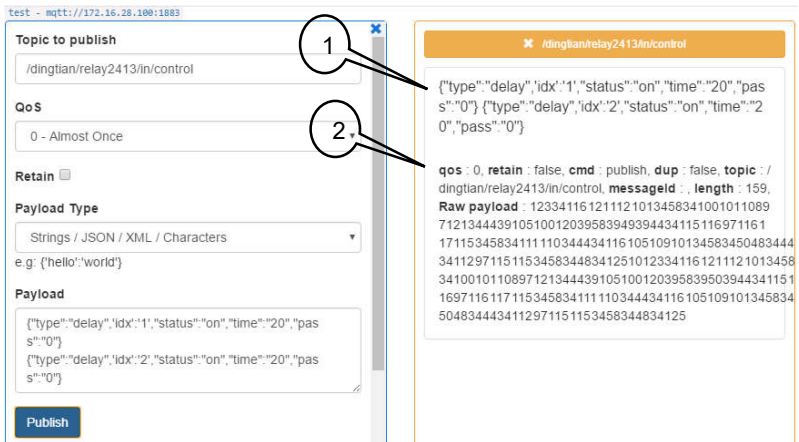


Рис. 5.15. Вікно клієнта-видавця та клієнта-передплатника



Рис. 5.16. Фрагмент повідомлень від модуля

5.7 Оформлення результатів проведення заняття

Складіть звіт про проведене заняття, де наведіть схеми комунікаційних з'єднань, умови обміну даними. У звіті повинні бути наведені скріншоти з екрану монітора та демонстрацією результатів виконання завдань.

5.8 Контрольні запитання

- 1) Надайте коротку характеристику та призначення протоколу *MQTT*.
- 2) Визначте складові повідомлення у протоколі *MQTT*
- 3) Наведіть приклади використання протоколу *MQTT* у технологіях *IoT* та *IIoT*.
- 4) Яким чином реалізована система якості сервісу у протоколі *MQTT*?

Практичне завдання 6

ВИКОРИСТАННЯ ХМАРНИХ ТЕХНОЛОГІЙ ДЛЯ ДИСТАНЦІЙНОГО МОНІТОРИНГУ ТА КЕРУВАННЯ ВІДДАЛЕНИМИ ТЕХНОЛОГІЧНИМИ ОБ'ЄКТАМИ

6.1 Мета проведення практичного завдання

На виконання цього завдання у відповідності до робочої програми відводиться 2 ак. години аудиторного та 2 ак. години часу самостійної роботи студента (СРС).

Мета виконання завдання:

- закріпити на практиці отримані теоретичні відомості про використання хмарних технологій для дистанційного моніторингу та керування віддаленими технологічними об'єктами;
- засвоїти порядок використання хмарних технологій на прикладі сервісу *OwenCloud* [12];
- розробити та налаштувати РСУ для дистанційного моніторингу

температури за допомогою сервісу *OwenCloud* та застосуванням протоколу *ModBus*.

6.2 Апаратне забезпечення практичного завдання

Завдання виконується на робочому місці із ПК та стендом, на якому встановлено контролер *ПЛК150* із *WIFI-Serial* шлюзом на базі мікроконтролера *ESP-01* або із *GSM*-модемом. Шлюз (або модем) з'єднаний за допомогою інтерфейсу *RS-232* з ПЛК та надає можливість підключити контролер до відкритої *WIFI*-мережі (або до сервісу *GPRS*) із доступом до *Internet*. Комп'ютер є клієнтом сервісу *OwenCloud*, до якого відправляє запити через *Internet*. Сервер *OwenCloud* у свою чергу виконує функцію головного пристрою по відношенню до підключеного до нього пристрою із протоколом *ModBus*. Сервіс забезпечує функції опитування пристроїв та відправлення відповідей на запити користувачів. Схема варіантів з'єднань ПК, ПЛК та комунікаційних пристроїв зображена на рис. 6.1.

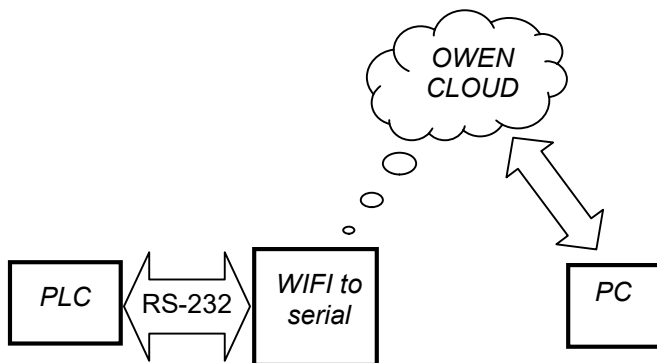


Рис. 6.1. Комунікаційна схема застосування сервісу *OwenCloud*

6.3 Програмне забезпечення практичного завдання

Основним програмним компонентом, який застосований під час виконання завдання є хмарний сервіс *OwenCloud*. Цей сервіс доступний для віддаленого доступу до даних у пристроях автоматизації за допомогою мережі *Internet*. Виконання завдання неможливе при наявності *proxy-server*, який активований на каналі доступу до *Internet* у надавача

послуг для користувачів НТУ «ХПІ». Це означає, що доступ з боку сервісу до хоста (*ПЛК150*) у локальній мережі вільний лише за протоколом *HTTP*, який використовує порт «80». Порт «502» протоколу *Mod-Bus* заблокований надавачом *Internet*-послуг. Тому вільний доступ до контролера з боку сервісу можливий лише завдяки його підключенню до *Internet* за допомогою *WIFI-Serial* шлюзу, який підключається до вільної *WIFI*-мережі із *SSID* «*ntu_khpi*». Пароль доступу до мережі не потрібний. Інше програмне забезпечення розміщено локально на ПК. Це браузер для доступу до сервісу та середовище *CoDeSys V2.3*. для програмування ПЛК [5].

6.4 Короткі теоретичні відомості

Хмарний сервіс *OwenCloud* застосовується для віддаленого моніторингу, управління та зберігання архівів даних пристроїв, які застосовують у системах автоматизації. Пристрої підключаються до сервісу послідовним інтерфейсом (із застосуванням мережних шлюзів) або через локальну мережу, яка має доступ до *Internet*. Якщо використаний контролер із інтерфейсом *Ethernet*, наприклад, *ПЛК150*, то потрібен лише доступ до *Internet*. Якщо використаний пристрій лише із послідовним інтерфейсом (наприклад, *RS-485*), то потрібно застосувати спеціальний мережний шлюз *PM210* або *PE210* виробництва компанії *OBEH*. Можливе використання віддалених модулів із інтерфейсом *Ethernet* (серії *Mx210 OBEH*).

Користувач отримує доступ до сервісу за допомогою *WEB*-інтерфейсу або мобільного додатка. В обох випадках необхідно підключення до мережі *Internet*. *WEB*-версія сервісу доступна за адресою: <https://cloud.owen.ua>. Мобільний клієнт для ОС *Android* доступний для скачування у сервісі *GooglePlay*.

Сервіс надає користувачам наступний базовий функціонал:

- збір даних з підключених пристроїв;
- зберігання розрахованих даних протягом 90 днів;
- відображення даних у вигляді мнемосхем, графіків та таблиць;
- відображення пристроїв на карті;

- віддалене управління пристроями;
- збереження та завантаження конфігурацій пристроїв;
- аварійні повідомлення по електронній пошті та через месенджер *Telegram*, *push*-повідомлення для мобільного додатка;
- інтеграція зі *SCADA*-системами за допомогою безкоштовного *OPC*-сервера *OWEN* або за протоколом *OPC UA*;
- відкритий *API* для інтеграції з іншими інформаційними системами (базами даних, тощо).

Сервіс підтримує підключення пристроїв, які визначені у керівництві [10] до використання *OwenCloud*. Усі пристрої повинні виконувати функцію підлеглого пристрою на послідовних інтерфейсах та *Ethernet* за протоколами *ModBus RTU/ASCII*, *OWEN* та *ModBus TCP*.

Більш детальну інформацію щодо використання та налаштування сервісу можна отримати за посиланням:

https://owen.ua/uploads/108/owencloud_rukovodstvo_polzovatelya.pdf.

6.5 Послідовність виконання завдання

Виконання завдання складається з декількох етапів. Спочатку на сервісі створюється обліковий запис для додавання пристрою та отримання «токена» (ідентифікатора пристрою). Далі створюється проєкт для *ПЛК150* із додаванням програмного модуля сервісу *OwenCloud* та додаванням мережних змінних. Потім змінні імпортуються до сервісу. Зауважимо, що сервіс *OwenCloud* підтримають контролери лінійки *ПЛК1xx* із версією вбудованого ПЗ мікроконтролера v2.17.0 та *target*-файла v2.12. Також сервіс підтримують контролери *ПЛК110[M02]* із версією вбудованого ПЗ мікроконтролера v0.3.66 та *target*-файла v3.12.

6.5.1 Створення облікового запису на сервісі.

Для створення облікового запису у браузері відкрийте сторінку сервісу за адресою <https://cloud.owen.ua>. Одразу на сайті буде запропоновано авторизуватися або зареєструватися. Також є можливість перегляду роботи сервісу у демо-режимі.

Після реєстрування необхідно додати до облікового запису при-

стрій. Для цього у режимі адміністрування облікового запису за допомогою кнопки *Добавление прибора*. У наступному вікні заповнити обов'язкові поля. До поля *IMEI* додати *MAC*-адресу ПЛК, до поля *Тип прибора* – обрати *ПЛК через Modbus TCP*. На рис. 6.2 зображено приклад налаштування вказаних полів.

Тип прибора*	ПЛК через Modbus TCP
Идентификатор*	6A:77:00:00:85:45 <small>Введите MAC-адрес ПЛК, который хотите подключить к OwenCloud. MAC-адрес указан на боковой грани прибора.</small>
Адрес в сети*	1
Заводской номер	
Название прибора*	PLC150_IL
Категории	
Часовой пояс*	GMT+2:00 <small>Время на странице прибора будет смещаться в зависимости от часового пояса.</small>

Отменить Добавить

Рис. 6.2. Налаштування *ModBus*-пристрою для підключення до сервісу *OwenCloud*

Адресу *ModBus*-пристрою можна змінити на будь-яку дозволена згідно протоколу (за умовчанням адреса *ModBus*-пристрою має значення «1»). Заводський номер та *MAC*-адреса контролера зазначений на бирці, яка наклеєна збоку пристрою.

Генерований «токен» буде використаний пізніше, під час налаштування контролера для роботи із сервісом *OwenCloud*.

6.5.2 Налаштування контролера для роботи із сервісом *OwenCloud*.

Налаштування контролера починається із визначення його основних параметрів та версії прошивання. Для цього потрібно запустити середовище програмування *CoDeSys V2.3*, а у ньому створити проєкт із

програмою користувача на мові CFC. Далі потрібно налаштувати комунікаційний шлюз, підключитись до контролера та запустити вбудований у середовище *CoDeSys* ПЛК-браузер. У командний рядок браузера ввести команду `PLCInfo` для отримання основних налаштувань контролера. Контролер на запит надасть інформацію, приклад якої зображений на рис.6.3.

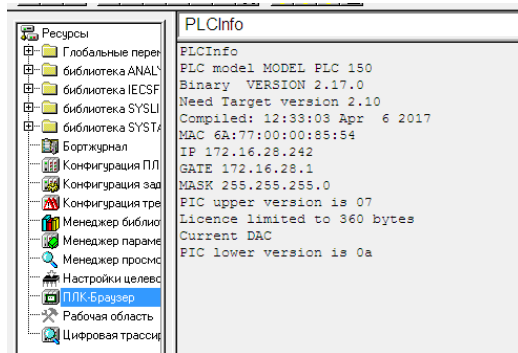


Рис. 6.3. Відповідь контролера на запит у ПЛК-браузері

Якщо внутрішнє програмне забезпечення ПЛК та його таргет-файл відповідають вимогам, які зазначені вище, то потрібно відключитись від контролера. Далі потрібно налаштувати проект для роботи сумісно із сервісом *OwenCloud*. Для цього потрібно до конфігурації контролера додати модуль підлеглого *ModBus*-пристрою та встановити його адресу (за умовчанням це буде адреса «1»). До фіксованого модуля *Modbus [FIX]* додати модуль *Cloud [VAR]*. Далі необхідно додати змінні, наприклад, це буде змінна типу *float* з ім'ям `pt50_net` для зберігання значення опору на аналоговому вході контролера. Значення змінна `pt50_net` отримує від іншої змінної, яка прив'язана до відповідного апаратного входу контролера. Отже, конфігурація контролера буде подібною до зображення на рис. 6.4.

На рис. 6.4 зображене налаштування модуля *Cloud [VAR]*, у рядок `CloudToken` додано «токен», який отриманий під час реєстрування пристрою (MRLYZU0Y). Зрозуміло, що цей «токен» прив'язаний до пристрою, а доступ до пристрою можливий лише за умови знання «логіну» та пароля для реєстрування у сервісі *OwenCloud*.

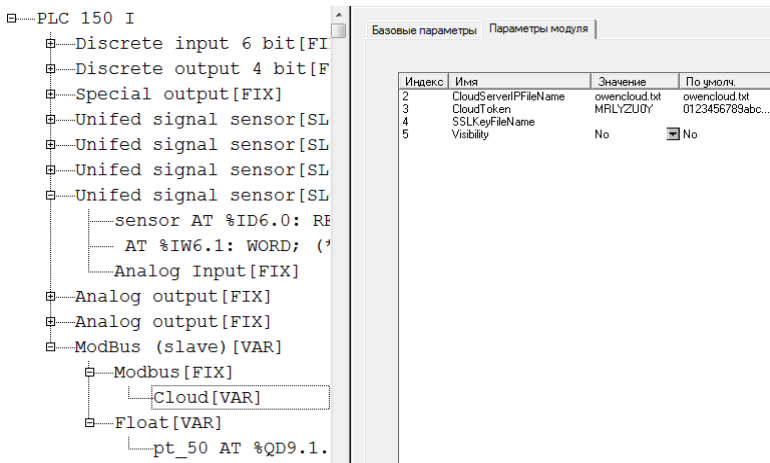



Рис. 6.4. Конфігурація підлеглого *ModBus*-пристрою для роботи із сервісом *OwenCloud*

Після зроблених налаштувань потрібно завантажити код проекту до контролера. При цьому потрібно згенерувати код для завантаження до постійного запам'ятовуючого пристрою ПЛК. На завершення потрібно експортувати змінні із ПЛК до сервісу *OwenCloud*

6.5.3 Додавання змінних до сервісу *OwenCloud*.

Сервіс *OwenCloud* може працювати лише із змінними, які у проекті об'явлені як глобальні. Для зручного налаштування сервісу у ньому передбачена можливість імпорту змінних, які попередньо експортовані із ПЛК. Для цього потрібно у середовищі *CoDeSys* із відкритим проектом створити файл експорту за допомогою команди меню «Проект | Експорт». У відкритому вікні потрібно позначити ресурс «Конфігурація ПЛК» та натиснути на кнопку «OK».

Наступний крок, це додавання файлу експорту до особистого кабінету у сервісі. Для цього потрібно у сервісі *OwenCloud* вибрати опцію «Администрирование» та натиснути на піктограму .

У групі «Параметри Modbus» потрібно обрати функцію «Імпортувати | Загрузить из Codesys v2.3» для завантаження файлу експорту. В результаті буде отримана таблиця змінних, зображення якої показано на рис. 6.5.

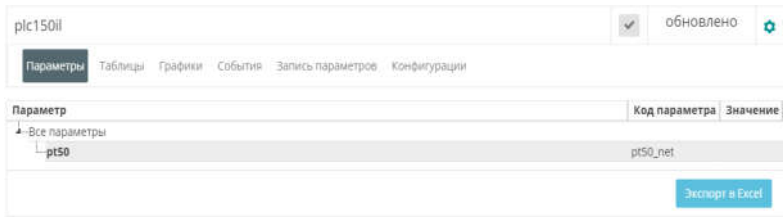


Рис. 6.5. Вікно із доданою змінною після імпорту файлу

Далі потрібно активувати режим перегляду пристрою для відображення значення параметру. Якщо усі налаштування зроблені вірно, то почнеться обмін даними між хмарою та пристроєм. Праворуч зверху відобразиться повідомлення про оновлення значень, тобто змінні отримують актуальні значення.

6.6 Завдання для самостійного виконання

Для самостійного виконання пропонується виконати завдання, в якому за допомогою хмарного сервісу можна керувати станом виходів ПЛК, наприклад, виходом із підключеним нагрівачем із одночасним моніторингом температури всередині теплового об'єкта.

6.7 Оформлення результатів проведення заняття

Складіть звіт про проведене заняття, де наведіть схеми комунікаційних з'єднань, умови обміну даними. У звіті повинні бути наведені скріншоти з екрану монітора та демонстрацією результатів виконання завдань.

6.8 Контрольні запитання

- 1) Наведіть алгоритм налаштування опитування ПЛК за допомогою хмарного сервісу *OwenCloud*.
- 2) Наведіть структурну схему взаємодії ПЛК та ПК за допомогою хмарного сервісу *OwenCloud*.
- 3) Наведіть приклад додавання змінних до хмарного сервісу *OwenCloud*.

ЗМІСТ

Вступ	3
<i>Практичне завдання №1.</i> Принципи побудови РСУ на основі використання безпроводних засобів дистанційного моніторингу та керування	5
<i>Практичне завдання №2.</i> Вивчення принципів побудови гетерогенних РСУ з клієнт-серверною архітектурою на основі використання шлюзу <i>WIFI–Ethernet–RS232</i>	22
<i>Практичне завдання №3.</i> Вивчення принципів побудови гетерогенних РСУ з клієнт-серверною архітектурою на основі використання шлюзу <i>Ethernet–RS232/RS485</i>	37
<i>Практичне завдання №4.</i> Вивчення принципів побудови гетерогенних РСУ з клієнт-серверною архітектурою на основі використання шлюзу <i>Ethernet–Serial</i>	49
<i>Практичне завдання №5.</i> Використання протоколу <i>MQTT</i> у РСУ з моделлю обміну даними «Видавець – Передплатник» ..	62
<i>Практичне завдання №6.</i> Використання хмарних технологій для дистанційного моніторингу та керування віддаленими технологічними об'єктами	83
Список джерел інформації	91

Навчальне видання

Лисаченко Ігор Григорович, **Подустов** Михайло Олексійович,
Бабіченко Анатолій Костянтинович, **Дзевочко** Альона Ігорівна

СУЧАСНІ КОМП'ЮТЕРНО-ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ
У РОЗПОДІЛЕНИХ СИСТЕМАХ УПРАВЛІННЯ

Навчально-методичний посібник
для студентів спеціальності
151 – «Автоматизація та комп'ютерно-інтегровані технології»
денної та заочної (дистанційної) форми навчання

Роботу до видання рекомендував доцент Олексій ДУДНИК

В авторській редакції
План 2021 р., поз. 26

Редактор – Гобельовська Л.П.

Підписано до друку 25.03.2021 р. Формат 60x84/16
Папір офсетний. Гарнітура шкільна. Друк цифровий.
Ум. друк. арк. 5,58. Тираж 50 прим. Зам. №297.

Матеріали надано: видавничим центром НТУ «ХП».
Свідоцтво про державну реєстрацію ДК № 5478 від 21.08.2017 р.
61002, Харків, вул. Кирпичова, 2.

Видавець та виготовлювач ТОВ «Друкарня Мадрид»
Через ФОП Гобельовська Л. П.
61024, м. Харків, вул. Максиміліанівська, 11. Тел.: 0800-33-67-62
Свідоцтво суб'єкта видавничої справи:
Серія ДК №4399 від 27.08.12 р.
www.madrid.in.ua info@madrid.in.ua