

МІНІСТЕРСТВО ОСВІТИ І НАУКИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ
для проведення лабораторних робіт з навчальної дисципліни
«Розподілені системи управління»

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

Харків
2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ
для проведення лабораторних робіт з навчальної дисципліни
«РОЗПОДІЛЕНІ СИСТЕМИ УПРАВЛІННЯ»

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

Затверджено
редакційно-видавничою
радою університету,
протокол № 3 від 30.10.2020 р.

Харків
НТУ «ХПІ»
2020

Методичні вказівки для проведення лабораторних робіт з навчальної дисципліни «Розподілені системи управління» для студентів спеціальності 151 – «Автоматизація та комп'ютерно-інтегровані технології» денної та заочної (дистанційної) форм навчання / уклад. І.Г. Лисаченко, М.О. Подустов, А.К. Бабіченко, А.І. Дзевочко. – Харків: НТУ «ХПІ». – 116 с.

Укладачі: І.Г. Лисаченко
М.О. Подустов
А.К. Бабіченко
А.І. Дзевочко

Рецензент: І. Л. Красніков

Кафедра автоматизації технологічних систем
та екологічного моніторингу

ВСТУП

Методичні вказівки вміщують необхідну інформацію для проведення лабораторних занять зі студентами денної та заочної (дистанційної) форм навчання за спеціальністю 151 – «Автоматизація та комп'ютерно-інтегровані технології» з навчальної дисципліни «Розподілені системи управління». Вони призначені для вивчення студентами основ розроблення та використання розподілених систем управління (PCY).

Елементи PCY складаються з промислового контролера (ПЛК) або панелі оператора (ОП), пристроїв зв'язку з об'єктом (ПЗО) або локальних регуляторів (Р). Усі зазначені пристрої з'єднані за допомогою інтерфейсу *RS-485* та взаємодіють за різними промисловими протоколами. Також пристрої з'єднані безпосередньо або через перетворювач інтерфейсів (ПІ) з персональним комп'ютером (ПК), на якому встановлено спеціальне програмне забезпечення (СПЗ) для програмування ПЛК та конфігурування модулів та ОП. Як об'єкт управління пропонується імітатор нагрівача та охолоджувача (модель теплообмінника). Регульованим параметром у макеті технологічної системи є температура всередині теплообмінника, яку вимірюють за допомогою або термометра опору, або термоелектричного перетворювача – термопари. В лабораторних роботах реалізований ПІД- або двопозиційний закон регулювання технологічного параметра, який здійснює або ПЛК або мікропроцесорний регулятор. У лабораторних роботах використовується метод аналогового управління дискретним виходом, так зване ШІМ-регулювання. У кожній лабораторній роботі наведені відомості про структуру та будову PCY та

необхідне СПЗ, яке потрібно використати для налаштування пристроїв для їхньої сумісної роботи. При цьому як виконавчі механізми використовуються дровий керамічний резистор для нагрівання та вентилятор для охолодження. Метою кожної роботи є організація мережної взаємодії ПЛК з іншими пристроями РСУ (ОП, модулями, тощо) та здійснення управління технологічним процесом (ТП).

Обладнання усіх стендів для виконання лабораторних робіт вироблено компанією ТОВ «ВО ОБЕН» (Україна), яка надала його безкоштовно згідно з програмою підтримки вищих навчальних закладів. Також безкоштовним є СПЗ для конфігурування та налаштування усіх пристроїв виробництва компанії. При використанні в РСУ контролерів серії *ПЛК150* виробництва компанії *ОБЕН* використане середовище *CoDeSys V2.3*, яке теж є безкоштовним.

Лабораторна робота 1

ОСНОВНІ ПРИНЦИПИ ФОРМУВАННЯ ЗАПИТІВ ТА АНАЛІЗ ВІДПОВІДЕЙ У ПРОТОКОЛІ MODBUS

1.1 Мета проведення лабораторної роботи

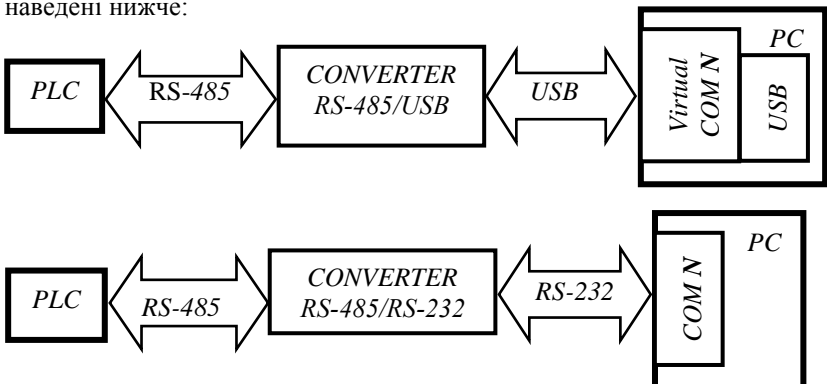
На лабораторну роботу у відповідності до робочої програми відводиться 4 ак. години аудиторного та 4 ак. години часу самостійної роботи студента (СРС).

Мета роботи:

- закріпити на практиці отримані теоретичні відомості про специфікації протоколу *ModBus* на послідовних інтерфейсах *RS-232* і *RS-485*;
- ознайомитися з реалізацією протоколу *ModBus* по послідовному інтерфейсу *RS-485* в контролері *ПЛК150* через конфігурацію його ресурсів;
- навчитись інтерпретувати *ModBus*-запити та *ModBus*-відповіді.

1.2 Апаратне забезпечення лабораторної роботи

Лабораторна робота проводиться на робочих місцях з ПК та стендами з контролерами *ПЛК150*. ПК не мають вбудованого інтерфейса *RS-485*, тому для зв'язку з ПЛК застосовуються ПІ *RS-485/RS-232* або *RS-485/USB*. Схеми варіантів з'єднань ПЛК та ПК наведені нижче:



УВАГА!

Номер та параметри віртуального COM-порту визначаються за допомогою «Диспетчеру пристроїв», який входить до стандартного набору сервісів ОС Windows

1.3 Програмне забезпечення лабораторної роботи

Для зв'язку ПК та ПЛК по послідовному інтерфейсу за допомогою ПП *RS-485/USB* в ОС *Windows* за допомогою драйверу емулюється віртуальний *COM*-порт, з яким без обмежень можуть працювати будь-які програми прослуховування, наприклад *COM Port Toolkit*. Однак ця програма не може інтерпретувати запити та відповідати. Для аналізу *ModBus*-запитів необхідно використовувати спеціальні програми для розкодування. Це програми *CAS Modbus Scanner* та *CAS Modbus RTU Parser*. Також в лабораторній роботі використовують програму-симулятор веденого пристрою *ModSim32* та програму-симулятор ведучого пристрою *OWEN Test Console*. Для реалізації протоколу *ModBus* у контролері *ПЛК150* використовується середовище *CoDeSys V2.3*.

1.4 Короткі теоретичні відомості

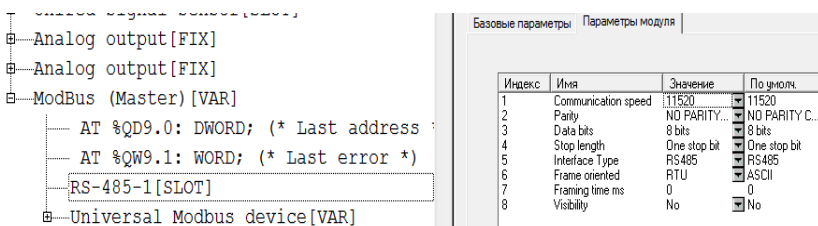
ModBus – це протокол обміну, і розроблений для ПЛК однойменний програмний модуль, який забезпечує роботу ПЛК відповідно до цього протоколу. Для доступу до даних ПЛК із застосуванням протоколу *ModBus* по послідовним інтерфейсам використовують технологію «головний-підлеглий»/«ведучий-ведений» (*master-slave*), при якій лише один пристрій (головний) може ініціювати обмін даними (зробити запит). Інші пристрої (підлегли) передають запитувані головним пристроєм дані або виконують запитувані дії. Для обміну даними за протоколом *ModBus* в ПЛК150 існує два варіанти реалізації: за допомогою бібліотеки *Modbus.lib* або за допомогою конфігурування ресурсів ПЛК. В останньому випадку до стандартної конфігурації ресурсів контролера додаються програмні модулі *ModBus (Master)* або *ModBus (slave)*, за допомогою яких реалізується обмін з ПЛК в статусі ведучого або веденого пристроїв. При цьому в проєкті, створеному в середовищі *CoDeSys* для ПЛК, програма користувача може бути порожньою (реалізується лише обмін даними). До складу зазначених модулів включені інтерфейси для обміну, які налаштовує користувач. Крім того, користувач додає потрібні мережні змінні для обміну даними. Лабораторна робота складається з двох частин: спочатку ПЛК виступає у ролі ведучого пристрою та взаємодіє з імітатором підлеглого пристрою – програмою

ModSim32, потім ПЛК конфігурується як підлеглий пристрій, з яким взаємодіє програма *CAS Modbus Scanner*. Для розкодування запитів використовується програма *CAS Modbus RTU Parser*.

1.5 Послідовність проведення лабораторної роботи

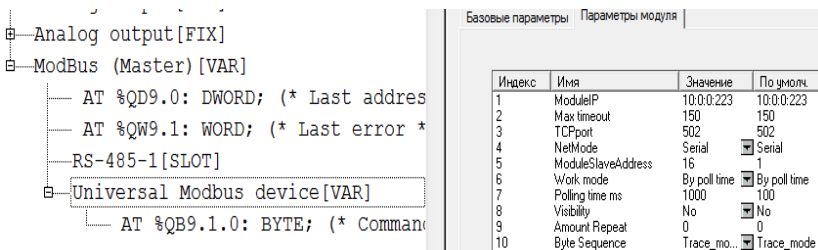
1.5.1 Програмування ПЛК150 для роботи у режимі головного пристрою

Отже, створіть в середовищі *CoDeSys V2.3* новий проект та визначте необхідний таргет-файл і мову реалізації *POU PLC_PRG*. Як цільову платформу оберіть ПЛК *OBEH150-IL*, а мову програмування виберіть *CFC*. Контролер буде лише опитувати підлеглий пристрій, тому *POU PLC_PRG* залиште порожнім. Далі перейдіть до вкладення «Ресурси» та оберіть (ЛКМ) утиліту «Конфігурація ПЛК». У ній за допомогою контекстного меню (ПКМ) додайте програмний модуль *ModBus (Master) [VAR]*, а у ньому змініть слот з інтерфейсом *RS-232* на *RS-485*. Налаштуйте параметри інтерфейсу обміну: встановить швидкість – «115200», налаштуйте формат кадру – «8-н-1» та режим роботи – «RTU», як це показано на рисунку нижче:



Индекс	Имя	Значение	По умолчанию
1	Communication speed	11520	11520
2	Parity	NO PARITY...	NO PARITY C...
3	Data bits	8 bits	8 bits
4	Stop length	One stop bit	One stop bit
5	Interface Type	RS485	RS485
6	Frame oriented	RTU	ASCII
7	Framing time ms	0	0
8	Visibility	No	No

Далі створіть віртуальну модель підлеглого пристрою – *Universal Modbus device [VAR]* та налаштуйте його параметри – тут досить вказати адресу пристрою, наприклад, «16», частоту запитів – 1 раз на секунду, інші параметри можна залишити без змін:



Индекс	Имя	Значение	По умолчанию
1	ModuleIP	10.0.0.223	10.0.0.223
2	Max timeout	150	150
3	TCPport	502	502
4	NetMode	Serial	Serial
5	ModuleSlaveAddress	16	1
6	Work mode	By poll time	By poll time
7	Polling time ms	1000	100
8	Visibility	No	No
9	Amount Repeat	0	0
10	Byte Sequence	Trace_no...	Trace_mode

Далі створіть мережні змінні. Це будуть вхідний та вихідний реєстри. Для змінних вкажіть номери реєстрів та функції, які застосовуються до цих реєстрів, як це показано нижче:

The image shows two screenshots of a configuration interface for a Universal Modbus device. The left side shows a tree view of the configuration, and the right side shows a detailed view of a selected module.

Top Screenshot:

- Tree view:
 - Universal Modbus device[VAR]
 - Register input module[VAR]
 - AT %QB9.1.0: BYTE; (* Command)
 - Register output module[VAR]
 - AT %QW9.1.1.0: WORD; (* Command)

Module parameters table:

Индекс	Имя	Значение
1	Register Address	8
2	Command	Read holding Registers (0x03)
8	Visibility	No

Bottom Screenshot:

- Tree view:
 - Universal Modbus device[VAR]
 - Register input module[VAR]
 - AT %IB9.1.0.0: WORD; (* Command)
 - Register output module[VAR]
 - AT %QW9.1.1.0: WORD; (* Command)

Module parameters table:

Индекс	Имя	Значение
1	Register Address	9
2	Command	Preset single register (0x06)
8	Visibility	No

Далі, скопіюйте проєкт (за допомогою функціональної кнопки F11), встановіть з'єднання з ПЛК за допомогою програмного комунікаційного модуля *Codesys Gateway* та завантажте код проєкту до ПЛК. Активуйте виконання програми користувача в ПЛК за допомогою функціональної кнопки F5. Не відключайтеся середовищем *CoDeSys* від ПЛК для спостереження за змінними у режимі *Online*.

Для приймання запитів від ПЛК запустіть на ПК програму моніторингу COM-порту *COM Port Toolkit*. Налаштуйте інтерфейс обміну та спостерігайте прийняті кадри у вікні програми. Нижче показані прийняті програмою кадри на читання та запис до реєстру.

Скопіюйте до буферу обміну (гарячі клавіші **Ctrl+C**) кадри обох запитів та розкодуйте їх за допомогою програми *CAS Modbus RTU Parser*. Результат розкодування показаний нижче:

1) аналіз кадра запиту на читання реєстра – 10 03 00 07 00 01 36 8A:

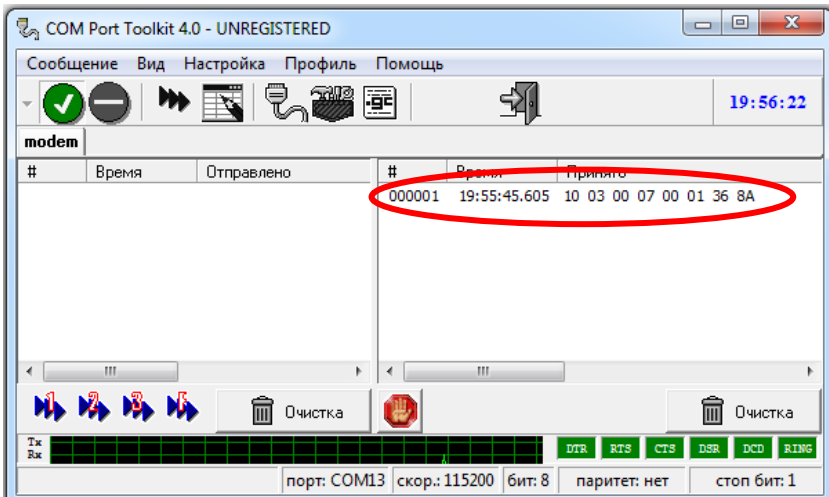
Device Address (адреса пристрою) =16₁₀ =0x10;

Function (код функції) =03₁₀ =0x03 =Read

Holding Registers;

Point Address (адреса реєстру) =40008₁₀;

Point Count (кількість реєстрів) =1.



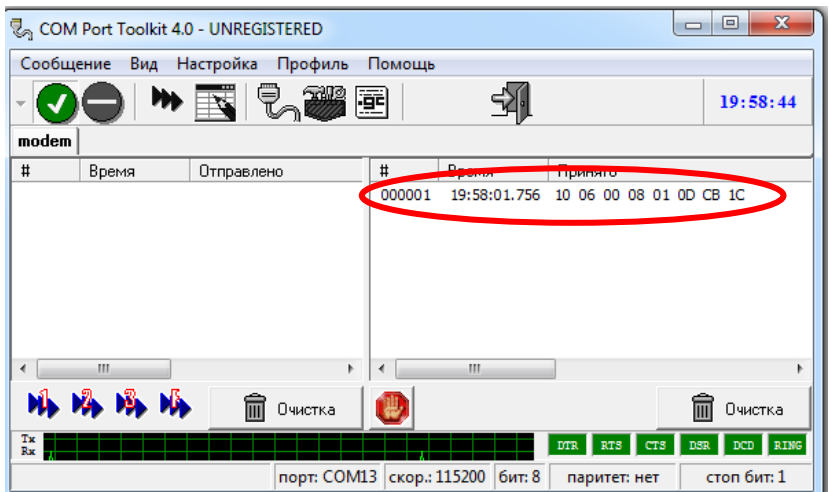
2) аналіз кадра запиту на запис значення до регістру – 10 06 00 08 01 0D CB 1C:

Device Address (адреса пристрою) = 16_{10} = $0x10$;

Function (код функції) = 06_{10} = $0x06$ =Preset Single Register;

Point Address (адреса регістру) = 40009_{10} ;

Required value (значення) = 269_{10} = $0x01 0D$.

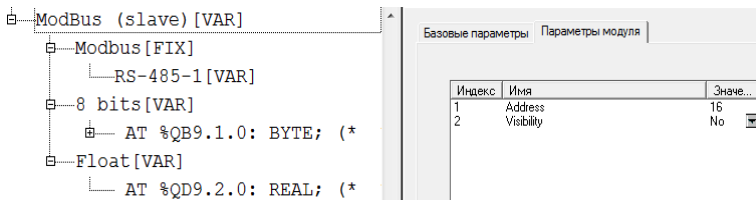


Другий кадр – це запит на запис числа «269» до регістру №9, тобто це друга мережна змінна типу Register output module [VAR].

Останні два байти в обох кадрах є контрольною сумою (CRC) для режиму обміну Modbus-RTU.

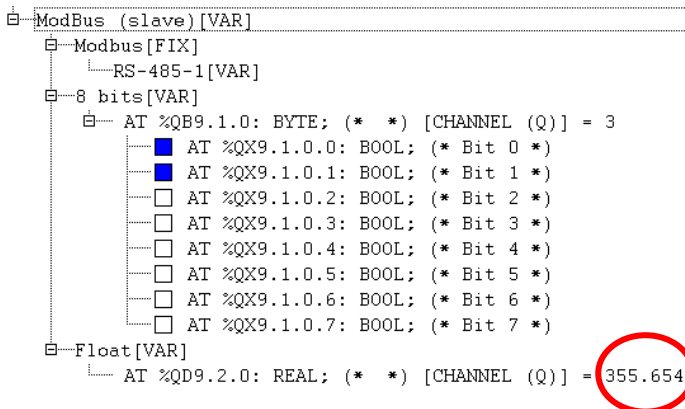
1.5.2 Програмування ПЛК у режимі підлеглого пристрою

Повторіть дії, які викладені у пункті 1.5.1, однак замість модуля головного пристрою додайте до конфігурації програмний модуль підлеглого пристрою – ModBus (slave) [VAR]. У параметрах модуля вкажіть його адресу – «16». Для цього модуля за допомогою контекстного меню додайте комунікаційний інтерфейс RS-485-1 [VAR]. Параметри обміну встановіть подібними до пункту 1.5.1, тобто «115200», «8-n-1». Далі додайте до модуля мережні змінні типів 8 bits [VAR] та Float [VAR]. У результаті буде отримана така конфігурація підлеглого пристрою:



Для читання та запису значень у створені змінні використайте програму CAS Modbus Scanner, а для розкодування запитів – програму CAS Modbus RTU Parser.

Нижче показано стан змінних підлеглого пристрою у конфігурації ресурсів ПЛК.



Тут у змінній типу 8 bits [VAR] встановлені 1-й і 2-й біти (тобто, знаходяться у стані «TRUE»), змінна типу Float [VAR] має значення «355,654». Відзначимо, що нумерація регістрів у пам'яті ПЛК здійснюється за допомогою, так званого «вирівнювання», яке описано в документації до ПЛК150. Відповідно до зазначеного правилом вирівнювання регістр зберігання стану обмоток має адресу «0», а дійсне значення параметру зберігається у регістрах з адресами «2» та «3». Тобто, відповідно до правила вирівнювання пропущений регістр з адресою «1».

Нижче показані запит на перевірку статусу окремих біт (функція «01») та відповідь, в якій статус окремих біт представлений числом у HEX-форматі:

=> Poll: 10 01 00 00 00 08 3E 8D

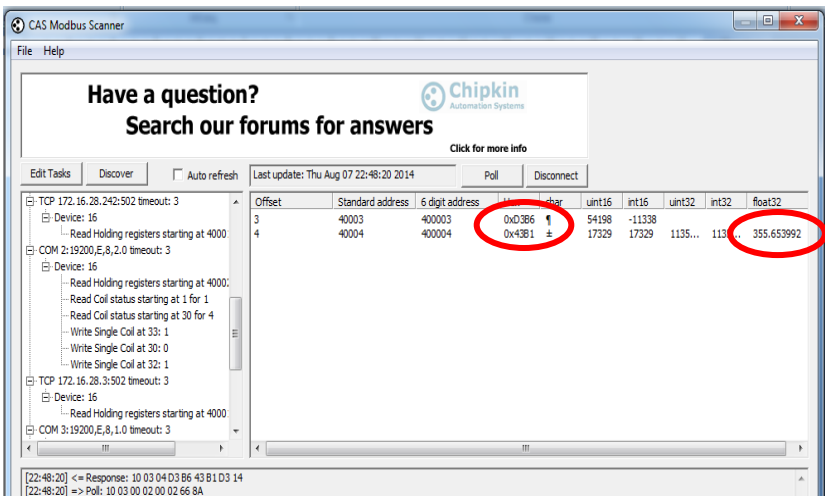
<= Response: 10 01 01 03 14 B5.

Далі показаний запит на встановлення бітів (функція «0F») у положення «TRUE» усіх восьми біт та квитанція про виконання команди:

=> Poll: 10 0F 00 00 00 08 01 FF 7E 15

Write task has completed successfully.

На рисунку нижче показаний запит на читання (функція «03») регістрів у ПЛК та результат читання регістрів з адресами «2» (40003_{10}) та «3» (40004_{10}), тобто регістрів для зберігання числа типу *Float*.



Розкодування запиту та відповіді:

[12:48:20] => Poll: 10 03 00 02 00 02 66 8A

[12:48:20] <= Response: 10 03 04 D3 B6 43 B1 D3 14

за допомогою програми *CAS Modbus RTU Parser* має такий зміст:

Frame Analysis 10 03 04 D3 B6 43 B1 D3 14

The frame has no errors.

Response Analysis

Device Address =16 =0x10

Function =3 =0x03=Read Holding Registers

Point Count =2 =Implied by byte count=4

Point Index =1 Value=54198₁₀=0xD3_B6. (Address

is unknown because the Poll Message was not provided.)

Point Index =2 Value=17329₁₀=0x43_B1. (Address is

unknown because the Poll Message was not provided.)

УВАГА!

**Значення змінної певним чином закодовано у двох регістрах.
Молодший регістр у відповіді першим, а байти всередині
регістра розміщені від старшого до молодшого**

*1.5.3 Перетворення числа формату 32 біт IEEE 754 у десяткове
число*

Щоб записати число у стандарті *IEEE 754* або відновити його,
необхідно знати три параметри:

- S – біт знаку (31-й біт);
- E – зміщення експоненти (30-23 біти);
- M – залишок від мантиси (22-0 біти), без одиниці.

Це цілі числа, які записані у форматі *IEEE 754* у бінарному
форматі.

Наведемо формулу для перетворення у десятковий вид з числа у
форматі *IEEE 754* одиничної точності:

$$F = (-1)^S \cdot 2^{(E-127)} \cdot \left(1 + M / 2^{23}\right)$$

де *F* – число у десятковому форматі. Нижче показані результати

перетворення значень регістрів у число з плаваючою точкою одинарної точності (32 біти).

У таблиці представлено десяткове число 355,654 у 32-х бітному форматі *IEEE754*:

1 біт	8 біт	23 біта	IEEE 754
0	100_0011_1	011_0001_1101_0011_1011_0110	431B_D3B6 _(hex)
0 ₁₀	135 ₁₀	1810432 ₁₀	355,654
знак числа	зміщена експонента	залишок від мантиси	число

1.6 Завдання для самостійної роботи

Для закріплення навичок роботи з ПЛК щодо реалізації *ModBus*-протоколу у режимі головного та підлеглого виконайте наступні завдання.

1.6.1 Програмування ПЛК ОВЕН у режимі головного пристрою

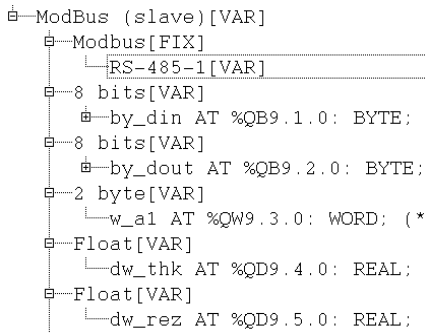
У програмі *ModSim32* створіть та налаштуйте чотири вікна для наступних типів даних: дискретний вхід – 10 елементів, дискретний вихід – 10 елементів, вхідний регістр – 10 елементів та *holding*-регістр – 10 елементів. Адреса *ModBus*-пристрою для усіх типів даних визначить рівним «32». Режим та параметри зв'язку встановіть наступними: «115200», формат кадру – «8-n-1» та режим обміну даними – «RTU».

Отримайте у програмі-симуляторі значення на 3-му і 4-му аналогових входах ПЛК, статус дискретних входів, а також налаштуйте можливість керування дискретними виходами.

Для реалізації даного завдання в ПЛК створіть програмний модуль *Modbus-Master*, налаштуйте комунікаційний модуль, додайте піделементи *Universal Modbus Device...* з відповідними мережними змінними для обміну даними. Далі завантажте проєкт до ПЛК та надайте викладачу результат запису та зчитування даних з симулятора.

1.6.2 Програмування ПЛК150 у режимі підлеглого пристрою

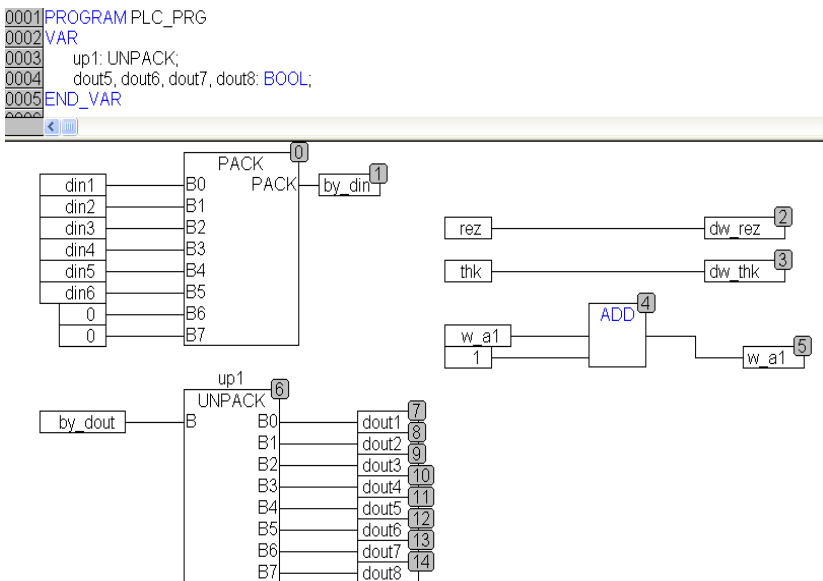
Створіть проєкт з модулем *Modbus-Slave*. Нижче показані регістри *Modbus-slave* пристрою:



Це байти стану дискретних входів `by_din` та стану дискретних виходів `by_dout` фізичних входів та виходів ПЛК. Змінна `w_a1` має тип `WORD` та представляє інкрементуючий лічильник (необхідно реалізувати програмно). Значення температури та змінного резистору зберігаються у змінних з іменами `dw_thk` та `dw_rez` відповідно.

Для читання та запису значень у змінні використовуйте програму *CAS Modbus Scanner*. Далі завантажить проєкт до ПЛК та надайте викладачу результат запису та зчитування даних з програми-сканеру.

В `POU PLC_PRG` на мові `CFC` програма буде такою:



Далі завантажить проєкт до ПЛК та налаштуйте програму *CAS Modbus Scanner* для запису та зчитування даних.

1.7 Оформлення результатів проведення занять і контрольні запитання

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

- 1) Які інтерфейси використовують для реалізації лінії зв'язку із застосуванням протоколу *ModBus*?
- 3) На яких рівнях моделі зв'язку побудований протокол *ModBus*?
- 4) Поясніть принцип обміну даними «головний»-«підлеглий».
- 5) Який формат кадру протоколу *ModBus* на фізичному рівні?
- 6) Який формат кадру протоколу *ModBus* на канальному рівні?
- 7) Який формат кадру протоколу *ModBus* прикладному рівні?
- 8) Як реалізована достовірність на канальному рівні *ModBus*-протоколу?
- 9) Які принципи адресації закладені у протоколі *ModBus*?
- 10) Якого типу дані описані у протоколі *ModBus*?
- 11) Як розділяють кадри в *ASCII*-режимі та *RTU*-режимі?
- 12) Чому в промислових мережах краще використовувати *RTU*-режим?

Лабораторна робота 2

ВЗАЄМОДІЯ КОНТРОЛЕРА ПЛК150 ТА ПАНЕЛІ ОПЕРАТОРА СМІ-1 ПО ІНТЕРФЕЙСУ RS-485 ЗА ПРОТОКОЛОМ MODBUS

2.1 Мета проведення лабораторної роботи

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

Мета роботи:

- закріпити на практиці отримані теоретичні відомості про

специфікації протоколу *ModBus* на послідовному інтерфейсі *RS-485*;

– ознайомитися з реалізацією *ModBus*-протоколу на послідовному інтерфейсі *RS-485* у *ПЛК150* (як головного та підлеглого пристрою) через конфігурування його ресурсів;

– навчитися конфігурувати панель оператора *СМІ-1* для мережної взаємодії з *ПЛК150*.

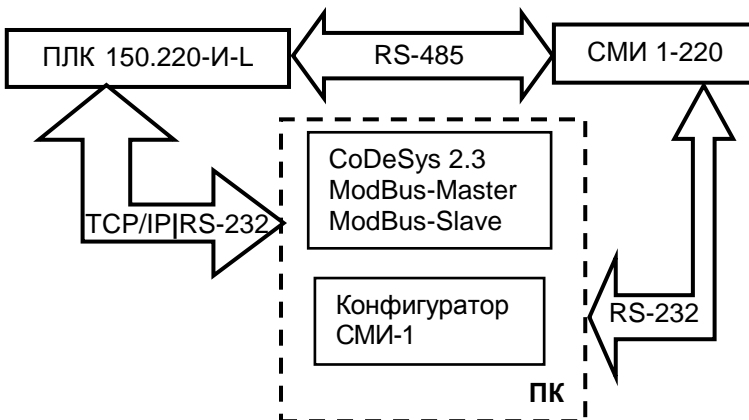
2.2 Апаратне забезпечення лабораторної роботи

Лабораторна робота проводиться на робочих місцях з ПК та стендами з *ПЛК150* та *СМІ-1*. Контролер та панель оператора з'єднані інтерфейсом *RS-485*.

2.3 Програмне забезпечення лабораторної роботи

Для створення та завантаження проєкту до *ПЛК150* використовується середовище *CoDeSys V2.3*. Конфігурування панелі оператора *СМІ-1* здійснюється за допомогою програми *Конфігуратор СМІ-1*.

На рисунку нижче представлена схема комунікаційних зв'язків і апаратного та програмного забезпечення для виконання лабораторної роботи.

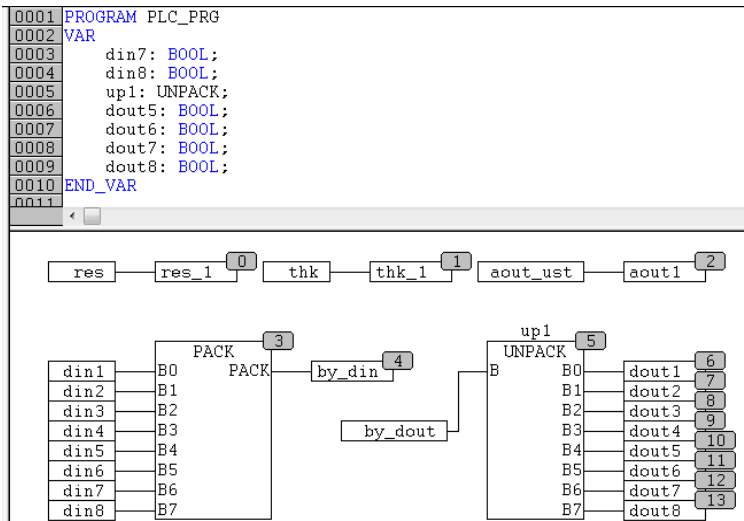


2.4 Послідовність проведення лабораторної роботи

2.4.1. Обмін даними за протоколом *Modbus* у режимі *ПЛК150* – головний та *СМІ-1* – підлеглий з використанням інтерфейсу *RS-485*

2.4.1.1 Програмування та конфігурування ресурсів ПЛК. Тіло

програми та поле об'явленнь змінних буде мати вигляд:



Конфігурування фіксованих модулів зробить таким:

PLC 150 I

- Discrete input 6 bit [FIX]
 - din1 AT %IX0.0: BOOL; (* *)
 - din2 AT %IX0.1: BOOL; (* *)
 - din3 AT %IX0.2: BOOL; (* *)
 - din4 AT %IX0.3: BOOL; (* *)
 - din5 AT %IX0.4: BOOL; (* *)
 - din6 AT %IX0.5: BOOL; (* *)
- Discrete output 4 bit [FIX]
 - dout1 AT %QX1.0: BOOL; (* *)
 - dout2 AT %QX1.1: BOOL; (* *)
 - dout3 AT %QX1.2: BOOL; (* *)
 - dout4 AT %QX1.3: BOOL; (* *)
- Special output [FIX]
- Unified signal sensor [SLOT]
- Unified signal sensor [SLOT]
- Termocouple sensor [SLOT]
 - thk AT %ID5.0: REAL; (* Value
 - AT %IW5.1: WORD; (* Circular
- Analog Input [FIX]
- Unified signal sensor [SLOT]
 - res AT %ID6.0: REAL; (* Value
 - AT %IW6.1: WORD; (* Circular
- Analog Input [FIX]
- Analog output [FIX]
 - aout1 AT %QD7.0: REAL; (* Val
- Analog output [FIX]
- ModBus (Master) [VAR]

Базовые параметры | Параметры модуля

Индекс	Имя	Значение
1	Type of sensor	RO_5000
2	Measure interval, s	0.5
3	Ain low	0
4	Ain high	500
6	First point	0
7	Delta	0
8	Second point	0
9	Delta	0
10	Third point	0
11	Delta	0
12	Visibility	No

Конфігурування модулів, які додаються до конфігурації ПЛК.

Налаштування комунікаційного субмодуля RS-485-1 [SLOT] у модулі ModBus (Master) [VAR] зробіть відповідно:

Индекс	Имя	Значение
1	Communication speed	9600
2	Parity	NO PARITY...
3	Data bits	8 bits
4	Stop length	One stop bit
5	Interface Type	RS485
6	Frame oriented	RTU
7	Framing time ms	0
8	Visibility	No

Налаштування субмодуля Universal Modbus device [VAR] (віртуальна модель підлеглого пристрою) будуть такими:

Индекс	Имя	Значение
1	ModuleIP	10.0.0.23
2	Max timeout	150
3	TCPport	502
4	NetMode	Serial
5	ModuleSlaveAddress	16
6	Work mode	By poll time
7	Polling time ms	100
8	Visibility	No
9	Amount Repeat	0
10	Byte Sequence	Native

Налаштування мережних змінних у підлеглому пристрої (СМІ-1):

- вихідний регістр для запису в СМІ-1 (регістр в панелі має номер «6») значення маски дискретних входів у ПЛК150:

Ин...	Имя	Значение
1	Register Address	6
2	Command	Preset singl register (0x06)
8	Visibility	No

- вхідний регістр для читання значення маски дискретних входів в СМІ-1 (регістр в панелі має номер «40»):

Инде...	Имя	Значение
1	Register address	40
2	Command	Read holding Registers (0x03)
8	Visibility	No

- вхідний реєстр для запису в *СМІ-1* значення змінної типу *WORD* з контролера *ПЛК150* (реєстр у панелі має номер «4»):

Индекс	Имя	Значение
1	Register Address	4
2	Command	Preset singl register (0x06)
8	Visibility	No

- вхідний реєстр для читання значення уставки (ціле, позитивне число) з *СМІ-1* (реєстр у панелі має номер «10»):

Индекс	Имя	Значение
1	Register Address	10
2	Command	Read holding Registers (0x03)
8	Visibility	No

- вхідний реєстр для читання значення уставки (УТС, 4...20 мА) з *СМІ-1* (реєстр у панелі має номер «8»):

Индекс	Имя	Значение
1	Register Address	8
2	Command	Read holding Registers (0x03)
8	Visibility	No

- вихідний реєстр для запису в *СМІ-1* значення змінної типу *REAL* від змінного резистору (0...1000 Ом) (реєстр у панелі має номер «0»):

Индекс	Имя	Значение
1	Register Address	0
2	Command	Preset multiple Registers (0x10)
8	Visibility	No

- вихідний реєстр для запису в *СМІ-1* значення змінної типу *REAL* від термопари (значення температури) на аналоговому вході *ПЛК150* (номер реєстру в панелі «2»):

Индекс	Имя	Значение
1	Register Address	2
2	Command	Preset multiple Registers (0x10)
8	Visibility	No

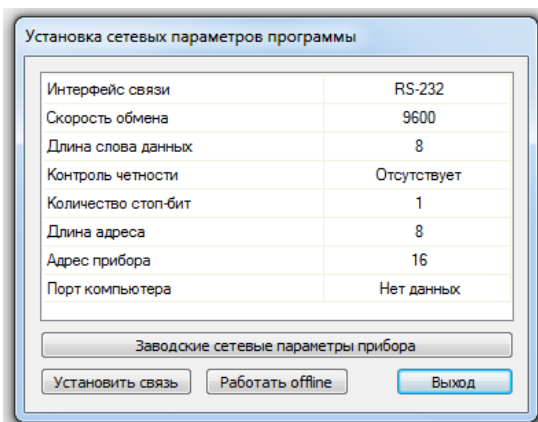
2.4.1.2 Конфігурування панелі оператора *СМИ-1*. Проведіть конфігурування панелі оператора. Для цього запустить програму *Конфигуратор СМІ-1*.

Після запуску програми, буде створено шаблон у вигляді дерева проекту, в якому панель оператора має статус відомого пристрою. Далі відкрийте гілку «Сетевые настройки», а у вікні «Свойства» у вкладенні «Выбор протокола» поставте «маркер» у полі напроти рядка «ОВЕН и Modbus RTU/ASCII». Потім потрібно вказати послідовний порт для мережного обміну з параметрами, подібними до налаштувань у головному пристрої (*ПЛК150*).

У гілці «Экраны редактирования» до переліку екранів додайте за допомогою кнопки «Добавить» параметри для редагування. Параметри редагуйте відповідно до їхнього типу у конфігурації ресурсів мережного модуля ModBus (Master) [VAR] у ПЛК.

У гілці «Экраны отображения» зробіть налаштування параметрів відображення на усіх екранах. Налаштуйте регістр для зберігання стану дискретних входів в панелі оператора, який знаходиться у гілці «Дискретные входы».

Мережні параметри програми-конфігуратора для зв'язку зі *СМИ-1* відповідають екранній формі:



Мережні параметри комунікаційного модуля для зв'язку *СМИ-1* з *ПЛК150* будуть такими:

Дерево проекта

- Проект СМІ1
 - Сетевые настройки
 - Экраны редактирования
 - ✓ Параметр редактирования 1
 - ✓ Параметр редактирования 2
 - Экраны отображения
 - ✓ Параметр отображения 1
 - ✓ Параметр отображения 2
 - Экран отображения 2
 - ✓ Параметр отображения 3
 - ✓ Параметр отображения 4
 - Дискретные входы
 - Безопасность
 - Мастер сети

Свойства

Настройки порта: Выбор протокола

Порт: RS-485 Порт: RS-232

Режим работы: Slave Режим работы: Slave

Скорость обмена	9600	Скорость обмена	9600
Длина слова данных	8	Длина слова данных	8
Контроль четности	Отсутствует	Контроль четности	Отсутствует
Количество стоп-бит	1	Количество стоп-бит	1
Длина адреса	8	Длина адреса	8
Адрес прибора	16	Адрес прибора	16
Задержка ответа	10	Задержка ответа	10

Налаштування мережних змінних у СМІ-1 будуть такими:

- змінна типу *float* для передачі на аналоговий вихід ПЛК150 значення UTC 4...20 мА від СМІ-1 на екрані «Параметр редакування 1»:

Дерево проекта

- Проект СМІ1
 - Сетевые настройки
 - Экраны редактирования
 - ✓ Параметр редактирования 1
 - ✓ Параметр редактирования 2
 - Экраны отображения
 - ✓ Параметр отображения 1
 - ✓ Параметр отображения 2
 - Экран отображения 2
 - ✓ Параметр отображения 3
 - ✓ Параметр отображения 4

Свойства

Тип параметра	float
min Значения	4
max Значения	20
Положение точки	0
Имя параметра	UST1
Текущее значение	6

Порт:

Режим: Slave

Протокол: ModBus-RTU

Источник

Номер регистра	8
Количество регистр...	2

- змінна типу *IntB* для передачі до ПЛК150 значення уставки 0...1000 од. від СМІ-1 на екрані «Параметр редакування 2»:

Дерево проекта

- Проект СМІ1
 - Сетевые настройки
 - Экраны редактирования
 - ✓ Параметр редактирования 1
 - ✓ Параметр редактирования 2
 - Экраны отображения
 - ✓ Параметр отображения 1
 - ✓ Параметр отображения 2
 - Экран отображения 2
 - ✓ Параметр отображения 3
 - ✓ Параметр отображения 4

Свойства

Тип параметра	IntB
min Значения	0
max Значения	1000
Имя параметра	WUST
Текущее значение	111

Порт:

Режим: Slave

Протокол: ModBus-RTU

Источник

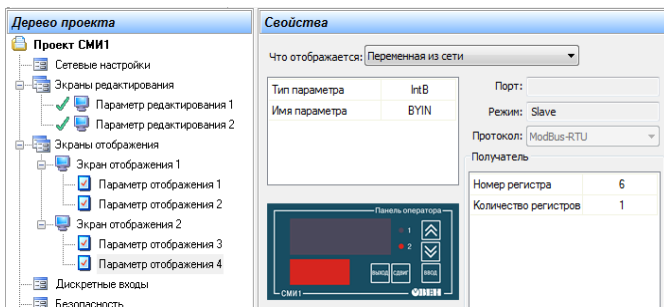
Номер регистра	10
Количество регистр...	1

- відображення значення змінного опору на аналоговому вході *ПЛК150* на першому екрані «Параметр отображения 1» змінної типу *float*:

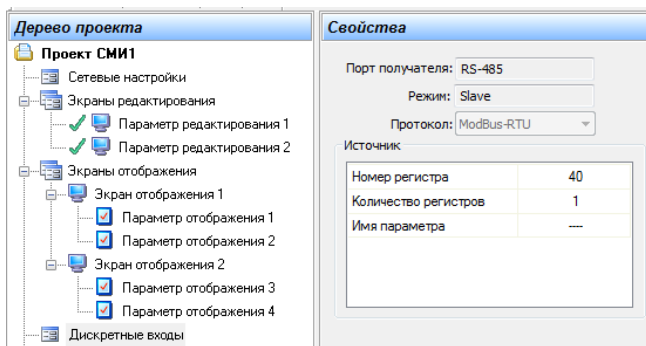
- відображення значення температури (від термопари) на аналоговому вході *ПЛК150* на першому екрані «Параметр отображения 2» змінної типу *float*:

- відображення значення змінної типу *WORD* до *ПЛК150* на другому екрані «Параметр отображения 3» змінної типу *IntB*:

- відображення значення маски дискретних входів до ПЛК150 на другому екрані «Параметр отображения 4» змінної типу *IntB*:



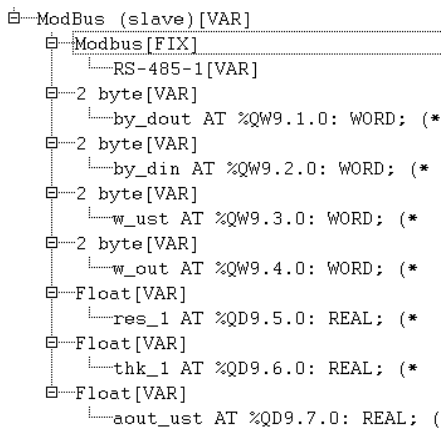
- змінна типу *unsigned* для передачі до ПЛК150 значення маски дискретних входів в СМІ-1 (Дискретные входы):



Насамкінець збережіть та завантажте проект до панелі, попередньо встановивши з нею зв'язок. Перевірте працездатність розробленого макета РСУ.

2.4.2 Обмін даними за протоколом Modbus у режимі СМІ-1 – головний та ПЛК150 – підлеглий із застосуванням інтерфейсу RS-485

2.4.2.1 Програмування та конфігурування ПЛК. Для реалізації мережної взаємодії у даному прикладі програма і змінні відповідають п. 2.4.1.1. Тобто конфігурація фіксованих модулів ПЛК буде такою ж. Відрізнитися буде лише конфігурація модулів ПЛК, що додаються (ModBus (slave) [VAR]). Підлеглому пристрою у конфігурації ПЛК надайте адресу «32». Параметри інтерфейсу залиште як у попередніх прикладах. Список регістрів для зберігання значень змінних буде наступним:

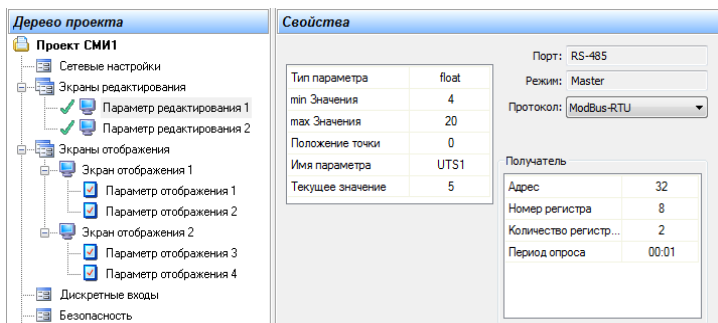


Нагадаємо, що адресування змінних у підлеглому пристрою визначається за правилами вирівнювання (див. ЛР№1). Тому список регістрів складений таким чином, щоб зменшити пропуски. Для цього спочатку створені змінні, які потребують меншого об'єму пам'яті.

2.4.2.2 Конфігурування операторської панелі. Мережні параметри програми-конфігуратора операторської панелі *СМІ-1* для зв'язку з ПК встановить подібними як визначено у п.2.4.1.2. Мережні параметри комунікаційного модуля для зв'язку з *ПЛК150* будуть теж подібними налаштуванням у попередньому прикладі, за винятком того, що тепер панель оператора буде працювати у режимі «Мастера».

Налаштування мережних змінних для обміну даними з *ПЛК150* будуть такими:

- змінна типу *float* для передачі на аналоговий вихід *ПЛК150* значення УТС 4...20 мА на екрані «Параметр редакування 1»:

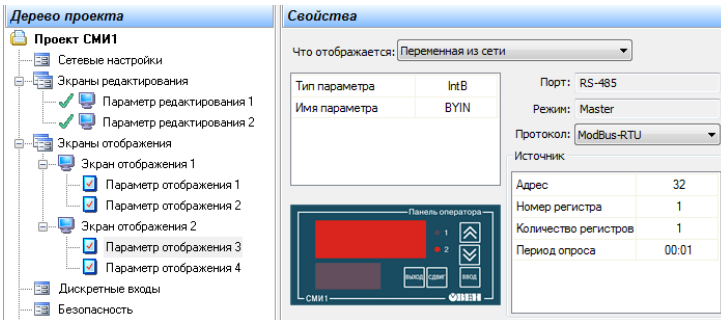


- змінна типу *IntB* для передачі в *ПЛК150* значення уставки 0...1000 од. на екрані «Параметр редакування 2»:

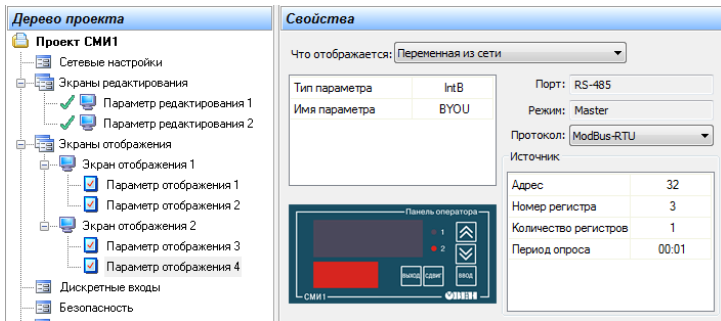
- відображення значення змінного опору на аналоговому вході *ПЛК150* на першому екрані «Параметр отображення 1» змінної типу *float*:

- відображення значення температури на аналоговому вході *ПЛК150* на першому екрані «Параметр отображення 2» змінної типу *float*:

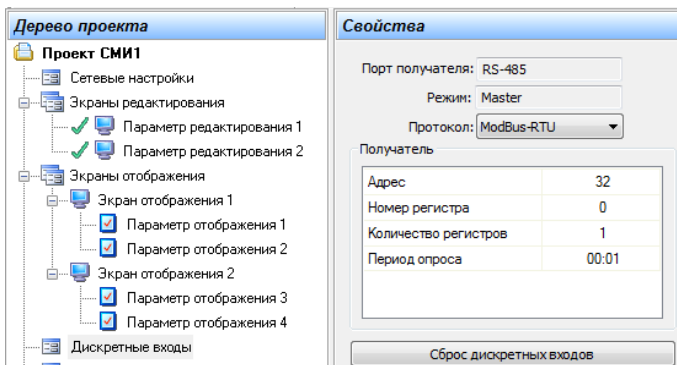
- відображення значення маски дискретних входів в ПЛК150 на другому екрані «Параметр отображения 3» змінної типу *IntB*:



- відображення значення змінної типу *WORD* в ПЛК150 на другому екрані «Параметр отображения 4» змінної типу *IntB*:



- змінна типу *unsigned* для передачі до ПЛК150 значення маски дискретних входів в *СМИ-1*:



2.5 Проєкт PCU для самостійного розроблення

Як приклад пропонується розробити проєкт з програмою користувача з ПД-регулятором температури з дискретним вихідним елементом (нагрівачем або холодильником) та використанням ШІМ-регулювання.

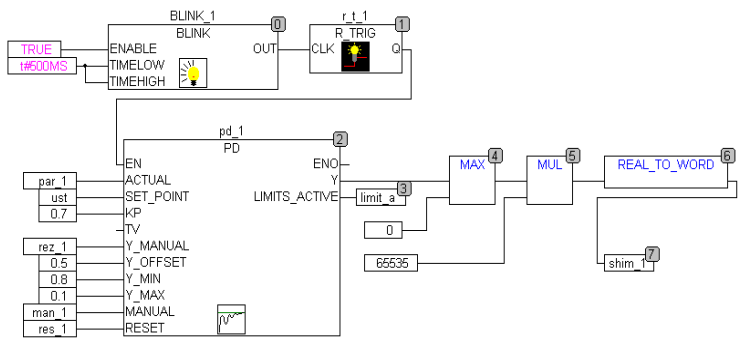
У програмі, змінна `par_1` – це поточне значення температури, змінна `ust` – значення уставки, а змінна `shim_1` керує 4-м дискретним виходом ПЛК (нагрівач), у якого апаратний ШІМ-канал має програмну адресу, яка дорівнює значенню «3» (нумерація каналів починається з нуля). Змінні `man_1`, `res_1` мають тип *bool* та служать для управління режимом роботи ПД-регулятора. Значення уставки вводиться на панелі оператора та передається по мережі до ПЛК. Значення управляючих сигналів контролер одержує із мережі, які формуються за допомогою перемикачів на емуляторі дискретних сигналів в панелі. Поточне значення параметра відправляється мережею до панелі для відображення. Вихід за межі сигналу управління сигналізує змінна `limit_a`. У разі включення ручного режиму управління керуючий вплив передається через змінну `rez_1`, яка пов'язана з аналоговим входом, до якого підключений змінний резистор. Запропонована програма має локальний характер. Тобто у програмі не передбачено обмін даними. Тому її необхідно доопрацювати шляхом дублювання змінних для їх відправлення та отримання з мережі. Також до конфігурації ПЛК потрібно додати відповідний програмний мережний модуль для обміну даними. З панеллю оператора. Крім того, для реалізації ШІМ-управління вихідним елементом (нагрівачем) до модулю дискретних виходів необхідно додати субмодуль «Pulse-wide modulator» з призначеним номером канала («3»).

Отже, створіть у середовищі *CoDeSys V2.3* новий проєкт. Перейдіть до вкладення «Ресурси» та виберіть утиліту «Конфігурація ПЛК». Додайте та проведіть конфігурацію програмного модуля *ModBus-Slave*, адресу залиште за умовчанням, тобто рівною «1». Додайте інтерфейс обміну *RS-485-1*. Параметри інтерфейсу визначте такими: «9600», «8-n-1» и «RTU».

```

PLC_PRG (PRG.CFC)
0001 PROGRAM PLC_PRG
0002 VAR
0003   BLINK_1: BLINK;
0004   r_t_1: R_TRIG;
0005   pd_1: PD;
0006 END_VAR

```



Далі до модуля Modbus (slave) додайте необхідну кількість елементів для зберігання змінних таких типів: 8 bits, Float. Наприклад, це будуть регістри для зберігання значень поточного параметра, уставки та керуючих сигналів («Ручний режим», «Скидання», тощо), якими будуть обмінюватися контролер та панель послідовним інтерфейсом RS-485:

Индекс	Имя	Значение	Тип	Исход.
1	Communication speed	3600	11520	
2	Parity	NO PARITY...	NO PARITY C	
3	Data bits	8 bits	8 bits	
4	Stop length	One stop bit	One stop bit	
5	Interface Type	RS485	RS485	
6	Frame oriented	RTU	ASCII	
7	Framing time ms	0	0	
8	Visibility	No	No	

Збережіть проект на жорсткому диску, скопіюйте його та завантажте код проекту до ПЛК. Запустіть проект в ПЛК на виконання.

Далі проведіть конфігурування панелі оператора відповідно до програми користувача у ПЛК.

У гілці «Экраны отображения» зробіть наступні налаштування:

- налаштуйте «Параметр отображения 1», на якому буде відображатися значення температури усередині нагрівача;
- налаштуйте «Параметр отображения 3», на якому буде відображатися значення уставки.

Далі, у гілці «Дискретные входы» налаштуйте реєстр для зберігання стану дискретних входів в панелі оператора.

Збережіть та завантажте проєкт до панелі, попередньо встановивши з нею зв'язок. Перевірте працездатність розробленого макета РСУ.

2.6 Оформлення результатів проведення заняття і контрольні запитання

Складіть звіт про проведенне заняття, де представте схему з'єднань та умови обміну даними. Також вкажіть, які режими використовувалися для обміну даними.

За результатами проведення заняття студент повинен дати логічне пояснення отриманим результатам. Крім того, студент повинен відповісти на контрольні питання, приклади яких представлені нижче:

- 1) На яких рівнях моделі зв'язку побудований протокол *ModBus*?
- 2) Який формат кадру протоколу *ModBus* на фізичному рівні?
- 3) Який формат кадру протоколу *ModBus* на каналному рівні?
- 4) Який формат кадру протоколу *ModBus* на прикладному рівні?
- 5) Як реалізована достовірність на каналному рівні *ModBus*-протокола?
- 6) Які принципи адресації закладені у протоколі *ModBus*?
- 7) Які режими обміну даними підтримує панель оператора *СМИ-1*?
- 8) Які інтерфейси є на борту панелі оператора *СМИ-1*?
- 9) Які функції може виконувати панель оператора *СМИ-1*?
- 10) Перерахувати типи даних, обмін якими підтримує панель *СМИ-1* по протоколу *ModBus*?

Лабораторна робота 3

ВЗАЄМОДІЯ ПАНЕЛІ ОПЕРАТОРА *СП310 (ИП320)* ТА КОНТРОЛЕРА *ПЛК150* ПО ІНТЕРФЕЙСУ *RS-485* ЗА ПРОТОКОЛОМ *MODBUS*

3.1 Мета проведення лабораторної роботи

На лабораторну роботу відповідно до робочої програми відводиться 4 ак. години аудиторного та 4 ак. години самостійної роботи студента (СРС).

Мета роботи:

- закріпити на практиці отримані теоретичні відомості про специфікації на протокол *ModBus* та на послідовний інтерфейс *RS-485*;
- закріпити на практиці принципи реалізації протоколу *ModBus* послідовним інтерфейсом *RS-485* в *ПЛК150*;
- навчитись конфігурувати панелі оператора *СП310 (ИП320)*.

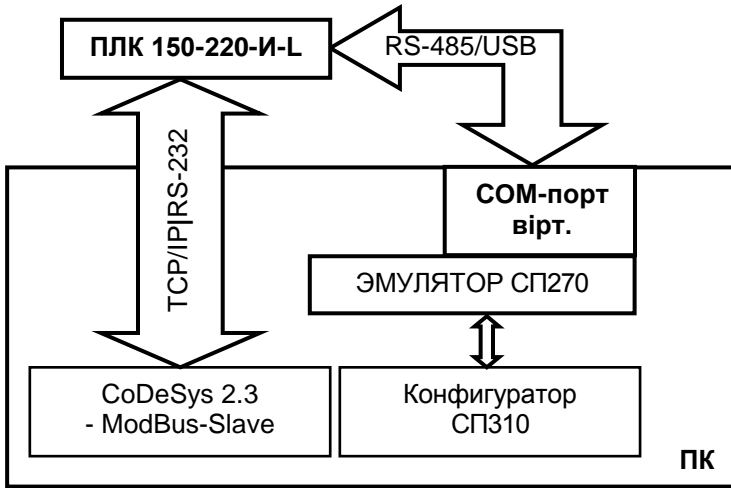
3.2 Апаратне забезпечення лабораторної роботи

Лабораторна робота проводиться на робочих місцях з ПК та стендами з *ПЛК150*. Операторська панель *СП310 ОВЕН* на стендах відсутня, тому в даній роботі застосовується її програмний емулятор, який активується у програмі конфігурації. Зв'язок емулятора панелі та ПЛК буде організований по інтерфейсу *RS-485*. Для реалізації проєкту РСУ з мережною взаємодією між *ПЛК150* та *ИП320* використовується стенд з панеллю та контролером.

3.3 Програмне забезпечення лабораторної роботи

Для створення та завантаження проєкту до ПЛК використовується середовище *CoDeSys V2.3*. Конфігурування панелі оператора *СП310* реалізується за допомогою програми *Конфігуратор СП3хх*, а панелі *ИП320* – програми *Конфігуратор ИП320*.

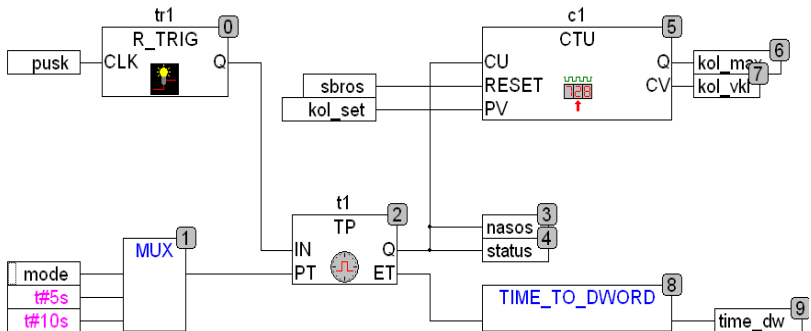
На рисунку нижче представлена схема комунікаційних зв'язків та використаного у лабораторній роботі апаратного та програмного забезпечення.



3.4 Послідовність проведення заняття

3.4.1 Програмування ПЛК для роботи у складі РСУ у режимі підлеглого ModBus-пристрою

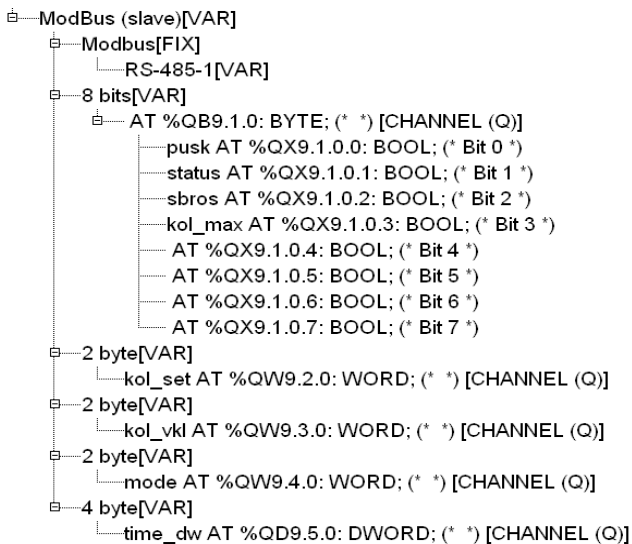
Як приклад пропонується розробити проект з програмою користувача для дистанційного керування насосом за допомогою панелі оператора. На панелі оператора можна обрати режим роботи насоса, визначити кількість дозволених вмикань, скинути лічильник вмикань та ввімкнути насос. Панель також відображає поточну кількість вмикань, стан насоса, поточний час його роботи та момент досягнення встановленого значення кількості вмикань. Нижче показаний проект програми користувача:



У програмі змінна `pusk` типу `bool`, яка надходить із мережі, запускає тригер, який в свою чергу, запускає таймер та інкрементує лічильник. Змінна `nasos` керує дискретним виходом, до якого підключений насос. З мережі надходить значення для змінної `mode`, яка управляє вибором часу роботи насоса. Змінні `mode`, `kol_set`, `kol_max`, `sbros`, `kol_vkl`, `time_dw`, `status` – є мережними та необхідні для обміну даними з операторської панеллю: це вибір режиму роботи насоса, встановлене значення кількості вмикань, сигналізація про досягнення допустимого значення, кнопка скидання кількості вмикань, поточну кількість вмикань, поточний час роботи насоса та його статус, відповідно. Всі змінні, крім `nasos` є мережними.

Створить у середовищі *CoDeSys V2.3* проєкт відповідно до опису та додайте програмний модуль для мережного обміну типу **ModBus (slave)**. Підлеглому пристрою призначте адресу «16». Як інтерфейсу обміну додайте *RS-485-1*. Параметри інтерфейсу визначте такими: «9600», «8-e-1» и «RTU».

До модуля **Modbus (slave)** послідовно додайте елементи для наступних типів даних: один типу **8 bits** – для бітових змінних, три типи **2 byte** – для зберігання даних типу *WORD* и один типу **4 byte** – для зберігання значення поточного часу, якими будуть обмінюватися ПЛК та ОП. Нижче показана структура веденого пристрою в ПЛК.



Збережіть проєкт на жорсткому диску, скопіюйте його та завантажте код проєкту до ПЛК. Запустіть проєкт в ПЛК на виконання.

3.4.2 Конфігурування СП310 для роботи у складі РСУ у режимі головного ModBus-пристрою

Запустіть програму конфігурації панелі оператора *Конфігуратор СП3хх*.

Програма *Конфігуратор СП3хх* – це середовище, яке призначене для конфігурування панелі оператора *СП3хх*. Програма працює під керуванням ОС *MS 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 ПЛК даними, які відображені у екранах користувача.

Панель оператора *СП3хх* відноситься до пристроїв людино-машинного інтерфейсу та призначена для відображення і редагування значень параметрів, які отримані через інтерфейс *RS-485* від ПЛК. Конфігурація панелі є сукупністю значень параметрів, які визначають роботу приладу. Проєкт 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 сенсорних елементів, за допомогою яких оператор здійснює безпосереднє керування функціонуванням об'єкта;
- управління функціонуванням ПЛК та/або інших приладів;
- запис та читання значень регістрів ПЛК та/або інших приладів, до яких підключена панель.

Отже, в програмі конфігурації створіть новий проєкт з режимом роботи панелі оператора як головного пристрою. У програмі передбачено покрокова конфігурація створеного проєкту за допомогою «*Мастера настройки*». На першому екрані діалогу виберіть тип панелі – *СП310* (або *СП307*), на другому екрані виберіть налаштування порту

завантаження – «Порт загрузки не используется», на третьому екрані введіть ім'я проєкту та натисніть на кнопку «Готово». В результаті з'явиться стандартне дерево проєкту з одним порожнім екраном.

Перш, ніж створювати та конфігурувати графічні елементи екрану, необхідно налаштувати інтерфейс та протокол обміну даними з ПЛК. Для цього у першу чергу необхідно зберегти проєкт на жорсткий диск за допомогою стандартних процедур. Далі, у меню «Файл» вибрати команду «Настройка...» та в діалоговому вікні вибрати вкладення «Устройства». У даному вкладенні позначте маркером напроти поля «Ведомый», а у полі «Порт PLC» – Modbus RTU (Панель Мастер) та налаштуйте параметри інтерфейсу, натиснувши на кнопку «...» напроти тексту «Параметры» (за замовчуванням встановлено – «9600», «8-e-1»). У полі напроти тексту «Адрес устройства» вкажіть потрібну адресу. Це буде адреса «16». Після вказаних дій натисніть кнопку «ОК». Вікно закриється, а введені дані будуть збережені у проєкті. Тепер все готово для створення людино-машинного інтерфейсу, тобто екранів з інтерактивними графічними елементами.

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, використовуючи довідкову систему програми, створіть екран з елементами управління та відображення. Нижче показаний варіант оформлення одного екрану у панелі:



На екрані зображені графічні елементи та об'єкти різного функціонального призначення:

- статичний текст, який не пов'язаний з мережними елементами: це інформаційні табло: «*Выбор режима*», «*Установить количество включений*», «*Время работы насоса*» та назва екрану – «*Дистанционное управление насосом*»;

- динамічний текст, який залежить від значення мережної змінної – *mode*. Залежно від значення змінної на панелі відображається значення тривалості роботи насоса: для значення «0» – це 5 с, для значення «1» – 10 с;

- поля для введення значень параметрів роботи насоса – вибір режиму роботи насоса (тривалість роботи насоса) *mode*, встановлене значення кількості вмикань для запису до ПЛК *kol_set*;

- поля для відображення поточних параметрів насоса – поточна кількість вмикань *kol_vkl* та поточного часу роботи насоса *time_dw*, які зчитані з ПЛК;

- кнопки управління насосом - для включення насоса *pusk* та скидання лічильника включень *sbros* для запису в ПЛК;

- індикатори стану параметрів насоса – досягнення встановленої кількості вмикань *kol_max*, яке зчитане з ПЛК;

- імітатори роботи насоса – інтерактивні трубопроводи та насос, які залежать від значення змінної *status*;

- на екрані також присутні табло з відображенням поточного часу та дати.

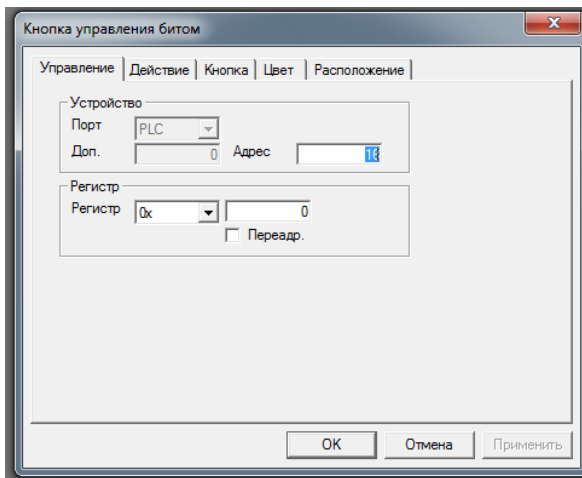
Аби не заглиблюватися в особливості редагування графічних елементів з точки зору дизайну, зупинимося на принципах їх конфігурації з точки зору мережної взаємодії.

Основними параметрами для зв'язку графічного елементу з мережними елементами є адреса пристрою у мережі, номер та тип регістра. Відповідно до структури модуля *Modbus (slave)*, який створений у конфігурації ресурсів ПЛК (дивись вище) адреси та типи регістрів мають наступні значення, які представлені у таблиці.

Для усіх змінних адрес *Modbus*-приладу однаковий – 16. З огляду на принцип вирівнювання змінних, у структурі пам'яті ПЛК у регістрі з адресою «0» не використано байт з адресою «1», а у байті з адресою «0» у явному вигляді не зайняті біти з адресами з «4» до «7».

Ім'я змінної у ПЛК	Тип даних змінної	Тип даних <i>Modbus</i> -протоколу	Номер елемента		
			регістр	байт	біт
pusk	BOOL	8 bits	0	0, 1	0
status	BOOL				1
sbros	BOOL				2
kol_max	BOOL				3
kol_set	WORD	2 byte	1	2, 3	-
kol_vkl	WORD	2 byte	2	4, 5	-
mode	WORD	2 byte	3	6, 7	-
time_dw,	DWORD	4 byte	4, 5	8, 9, 10, 11	-

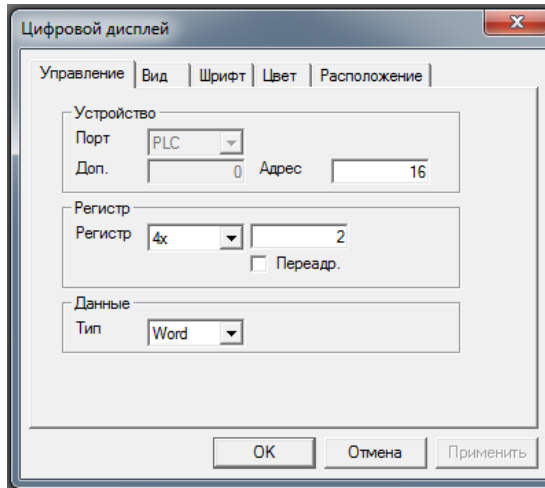
На рисунку показаний приклад конфігурації піктограми Кнопка управління битом, за допомогою якої формується короткий імпульс для вмикання насоса.



На рисунку показані значення адреси *Modbus*-пристрою та тип регістра для зберігання значення мережної змінної. Для кнопки використовується тип «0x», який відповідає у *Modbus*-протоколі типу даних «обмотка», у якій адреса дорівнює «0».

На прикладі налаштування піктограми Цифрової дисплей показано відображення мережної змінної kol_vkl, яка показує кількість

вмикань насоса. Тип регістру «4х», який відповідає у протоколі *Modbus* типу «*holding-регистр*», у якого адреса дорівнює значенню «2».



Після закінчення створення усіх елементів екрану збережіть проект на жорсткому диску. Для перевірки працездатності макета РСУ використовуйте емулятор панелі спільно з ПЛК. Для цього натисніть на відповідну піктограму у панелі інструментів програми-конфігуратора. Виберіть потрібний *SOM*-порт ПК, до якого підключений ПЛК. Якщо на екрані панелі відображається повідомлення про відсутність зв'язку з пристроєм (ПЛК), перевірте номер та налаштування *SOM*-порта в ПК.

У разі вдалого підключення перевірте працездатність РСУ управління насосом.

3.5 Завдання для самостійного опрацювання

Як завдання необхідно розробити проект з мережним обміном між контролером *ПЛК150* та панеллю оператора *ИП320*, у якому теж реалізовано дистанційне керування насосом. Конфігурування панелі оператора *ИП320* здійснюється за допомогою програми *Конфигуратор ИП320*.

Програма *Конфигуратор ИП320* призначена для конфігурування панелі оператора *ИП320*. Програма працює під управлінням ОС *Windows* та дозволяє формувати і зберігати призначені для користувача

екрани, які відображаються на дисплеї панелі у процесі експлуатації та налаштовувати обмін даними між панеллю та ПЛК, які відображені на екранах користувача. Конфігурація панелі являє собою сукупність значень параметрів, які визначають роботу приладу. Проект конфігурації панелі зберігається у спеціальному файлі з розширенням *.dp2.

В програмі конфігурування створіть новий проект, у якому панель оператора буде виконувати функцію головного пристрою. Для цього у вікні «*Выбор ПЛК*» у рядку «*Тип ПЛК*» оберіть Modbus-RTU (ИП320_Master). Далі, в цьому ж вікні за допомогою кнопки «*Настройки*» активуйте вікно «*Настройки параметров соединения*», де вкажіть параметри інтерфейсу – «9600», «8-n-1». Після того, як необхідні значення задані – натисніть кнопку «ОК». Вікно закриється, а введені дані будуть збережені у проекті. Далі натисніть кнопку «ОК» у вікні «*Выбор ПЛК*». Вікно закриється, а введені дані теж будуть збережені у проекті. Тепер все готово для розроблення людино-машинного інтерфейсу.

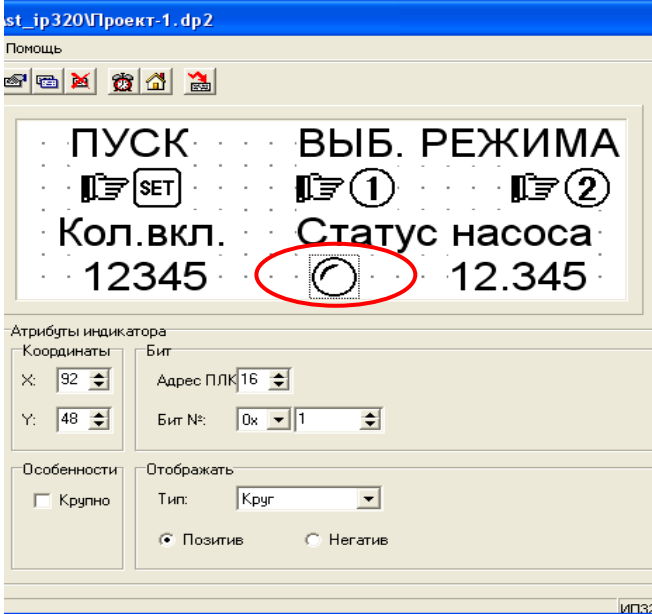
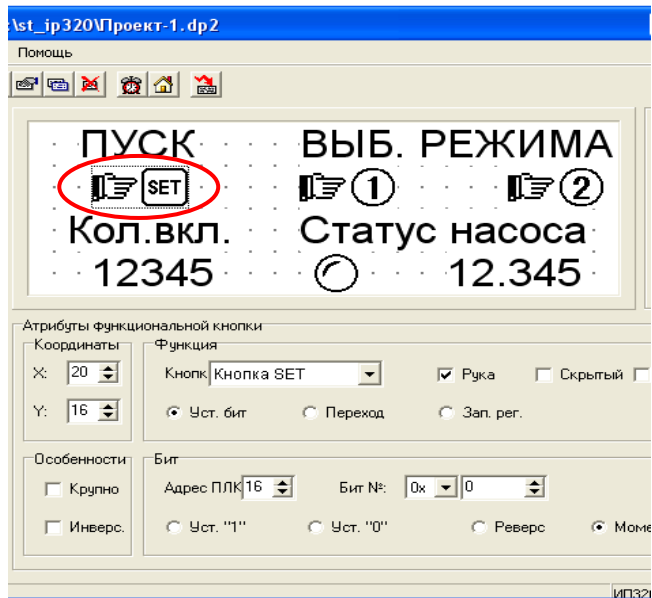
Робота з програмою полягає у створенні екранів, виборі графічних елементів та налаштуванні їхніх властивостей. Для конфігурування панелі оператора *ИП320* та створення екрана з графічними елементами управління і відображення використайте довідкову систему програми.

Нижче показано головний екран з елементами керування насосом:

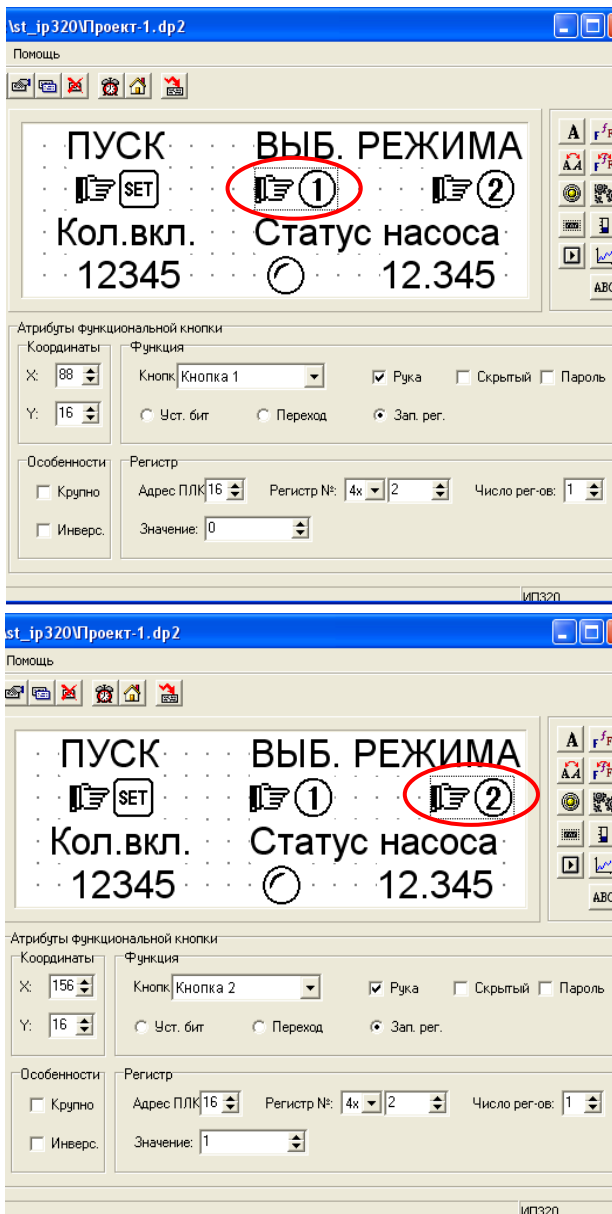


Текстові елементи ПУСК, ВЫБ. РЕЖИМА, Кол.вкл. та Статус насоса є статичним тому не потребують додаткових налаштувань.

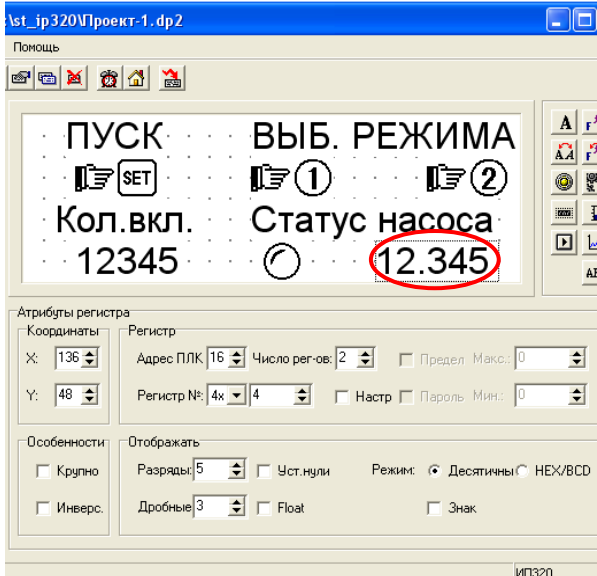
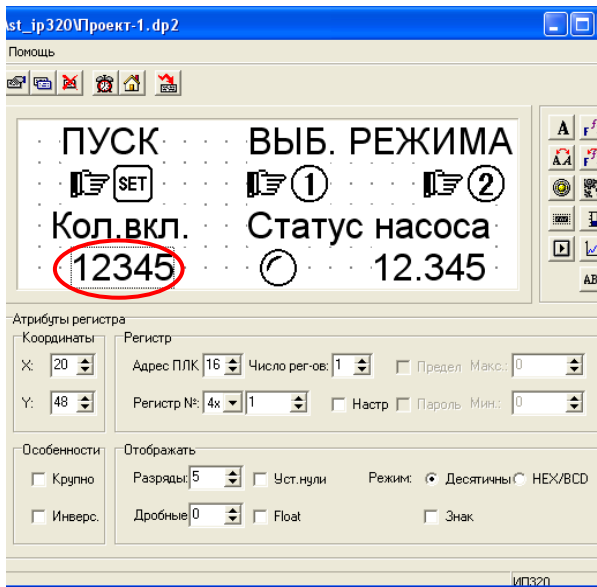
Налаштування елементів управління для вмикання насоса та відображення його статусу:



Налаштування елемента управління «Кнопка» для вибору режимів роботи насоса:



Налаштування елементів відображення кількості вмикань та поточного часу роботи насоса:



Після закінчення розроблення проекту збережіть його на жорсткому диску та завантажте проект до панелі, попередньо встановивши з нею зв'язок. Перевірте працездатність макету РСУ.

3.6 Оформлення результатів проведення заняття та контрольні питання

Складіть звіт про проведене заняття, де надайте схему з'єднань, умови обміну даними. Також вкажіть, які режими використовувалися для обміну даними.

За результатами проведення заняття студент повинен дати логічне пояснення отриманим результатам. Крім того, студент повинен відповісти на контрольні питання, приклади яких представлені нижче:

- 1) Які режими обміну підтримує панель оператора *СП310*?
- 2) Які інтерфейси присутні на борту панелі оператора *СП310*?
- 3) Які функції може виконувати панель оператора *СП310*?
- 4) Перерахувати типи даних, обмін якими підтримує панель *СП310* за протоколом *ModBus*?
- 5) У чому полягає сутність налаштування піктограм сенсорних панелей серії СП?
- 6) Які функції може виконувати панель оператора *ІП320*?
- 7) Які інтерфейси присутні на борту панелі оператора *ІП320*?

Лабораторна робота №4

ОСНОВНІ ПРИНЦИПИ СТВОРЕННЯ РСУ НА ПОСЛІДОВНИХ ІНТЕРФЕЙСАХ ТА ЗА ПРОТОКОЛОМ MODBUS

4.1 Мета проведення лабораторної роботи

На лабораторну роботу відповідно до робочої програми відводиться 4 ак. години аудиторного та 4 ак. години самостійної роботи студента (СРС).

Мета роботи:

- закріпити на практиці навички організації мережної взаємодії контролерів *ПЛК150* з периферійними пристроями по послідовному інтерфейсу *RS-485* за протоколом *ModBus*;
- навчитись конфігурувати локальні регулятори серії *ТРМ*, модулі введення/виведення *МВА*, *МДВВ* та модулі серії *Мх*,

програмовані реле *ПР11х (ПР200)* для роботи у складі РСУ.

4.2 Апаратне забезпечення лабораторної роботи

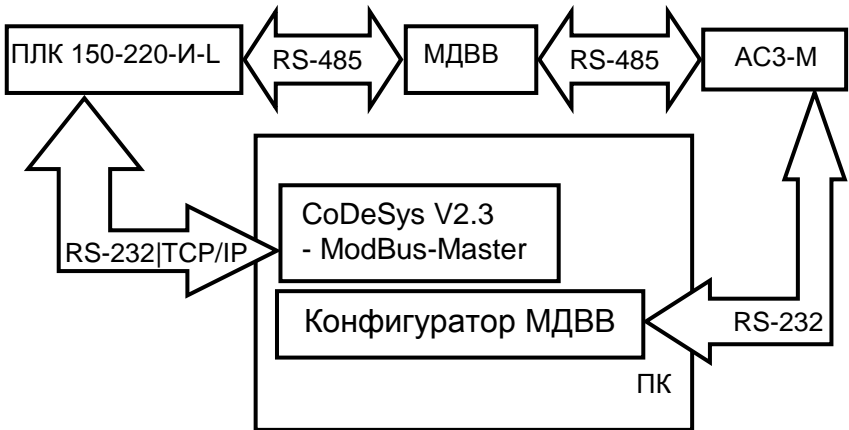
Лабораторна робота виконується на усіх стендах з ПЛК та периферійними пристроями. Якщо на стенді відсутній *ПЛК150*, то для мережної взаємодії з *ТРМ201* та *ПР110* використовується панель оператора або ПК з програмою-емулятором ведучого приладу (*СПЗхх*).

4.3 Програмне забезпечення лабораторної роботи

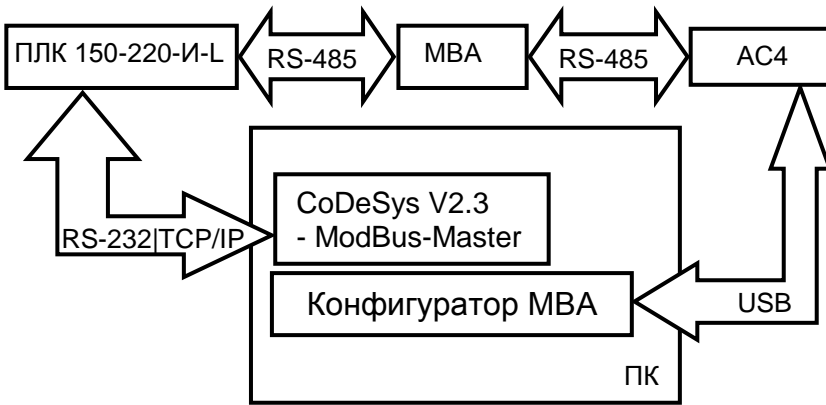
Для створення та завантаження проєкту до ПЛК використовується середовище *CoDeSys V2.3*. Також середовище *CoDeSys V2.3* використовується для візуалізації процесу регулювання та управління. В усіх випадках мережної взаємодії ПЛК виконує функцію головного пристрою, окрім випадку використання ОП. Конфігурування регуляторів та інших приладів проводиться за допомогою програм-конфігураторів. Для програмування реле *ПР110 (ПР200)* використовується програма *OWEN Logic*.

На рисунках представлені схеми комунікаційних зв'язків та використаного апаратного і програмного забезпечення для організації роботи РСУ:

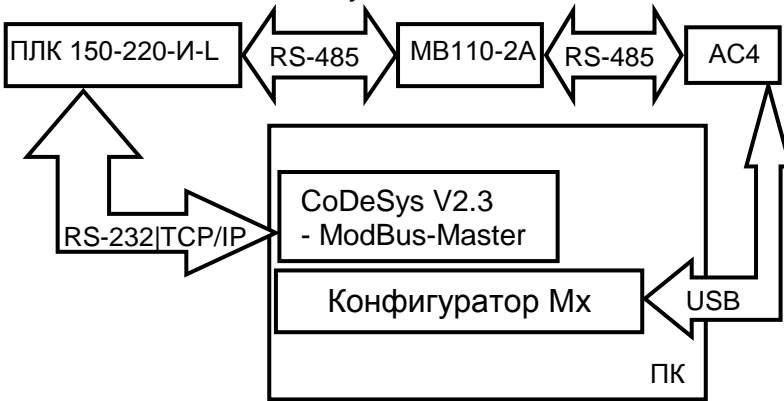
- на стенді з *ПЛК150* та модулем *МДВВ*:



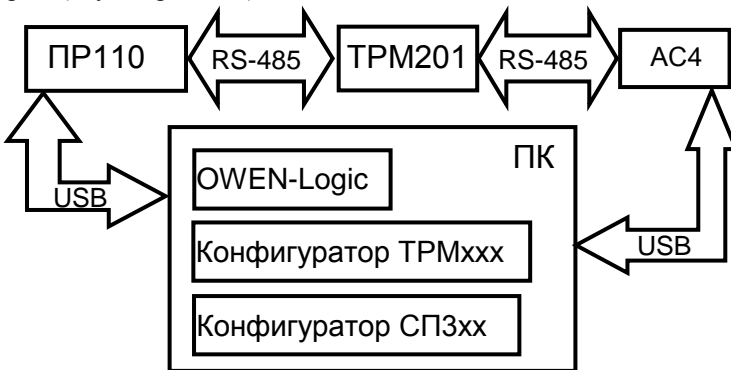
- на стенді з *ПЛК150* та модулем *МВА*:



- на стенді з ПЛК150 та модулем MB110-2A:



- на стенді з ТРМ201 та ПР110, де ПК виконує функцію головного пристрою (емулятор СПЗхх):



4.4 Послідовність проведення заняття

4.4.1 Розроблення РСУ на прикладі взаємодії ПЛК150 та ТРМ201

4.4.1.1 Налаштування приладів серії ТРМ для роботи у складі РСУ. Прилади серії *ТРМ* призначені для вимірювання та автоматичного регулювання температури, а також інших фізичних параметрів, значення яких первинними перетворювачами може бути перетворено у напругу постійного струму або уніфікований електричний сигнал постійного струму. Інформація про будь-яку з вимірюваних фізичних параметрів може відобразитися у цифровому вигляді на вбудованому індикаторі.

Прилади серії *ТРМ* виконують наступні функції:

- вимір температури або іншої фізичної величини;
- регулювання вимірюваної величини по ПД-закону шляхом імпульсного або аналогового управління або по двохпозиційного закону;
- дистанційне керування запуском та зупинкою регулювання;
- робота в мережі, яка організована за інтерфейсом *RS-485*, що дозволяє задавати необхідні режими роботи приладу та здійснювати передачу вимірюваних значень у мережу з максимальною швидкістю обміну до 115200 біт/сек з підтримкою протоколів *Modbus ASCII/RTU* та *OWEN*.

Параметри у приладі ТРМ поділяються на дві групи: конфігураційні та оперативні.

Конфігураційні параметри – це параметри, які задаються користувачем та визначають налаштування приладу. Значення конфігураційних параметрів зберігаються у незалежній пам'яті і не пропадають при виключенні живлення приладу.

Оперативні параметри переносять інформацію про поточний стан приладу або об'єкта регулювання. В оперативних параметрах передаються виміряні або обчислені значення, вихідні потужності регуляторів, номери запущених в даний момент програм, поточні стани вихідних елементів, тощо. Також до оперативних параметрів належать команди для керування приладом.

Усі параметри у групах мають спеціальні імена. Наприклад, оперативний параметр «*PV*» передає виміряне значення на сигнальному

вході, а параметр «SP» – зберігає значення уставки для регулювання. Решта імен параметрів приладу представлені у інструкції з експлуатації, там же вказані кодові значення датчиків. У керівництві з експлуатації представлений перелік реєстрів для зберігання конфігураційних, мережних та оперативних параметрів приладів та їхній опис відповідно до протоколу *Modbus*.

Для інтегрування приладу *TPM* до складу РСУ необхідно здійснити ряд кроків. По-перше, за допомогою програми-конфігурувати налаштувати інтерфейс та протокол обміну даними з ПК або ПЛК, по-друге – провести конфігурування приладу відповідно до схеми стенда. Прилади серії *TPM**** можуть працювати у мережі *RS-485* лише при наявності у ній головного приладу (ПЛК, ПК або ОП).

Конфігурування приладів проводиться за допомогою відповідних програм-конфігураторів, наприклад, програми «*Конфигуратор TPM101, TPM2xx*». Конфігурація пристрою вміщує налаштування параметрів регулятора та параметри мережного обміну. Після запуску програма-конфігуратор встановлює зв'язок з приладом через ПІ.

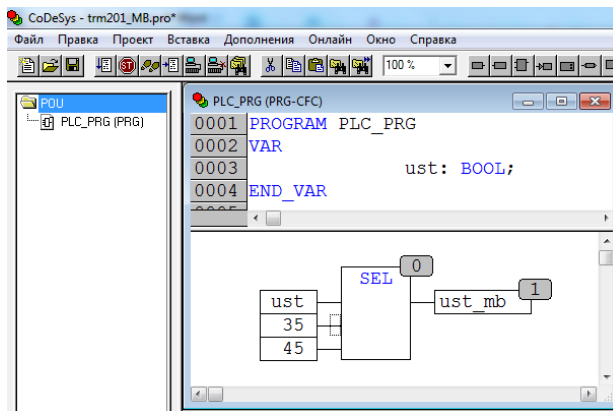
ПРИМІТКА: Якщо робота виконується на ПК, до якого стенд з потрібним приладом не підключений, то необхідно вибрати варіант «*Работать OFFLINE*». Номер *COM*-порта ПК з підключеним стендом призначити виходячи з параметрів меню ПК «*Панель управления*» (*COM*-порт емулюється на фізичному *USB*-порту ПК). У цьому випадку за допомогою програми-симулятора *ModBus*-пристрою створить віртуальну модель регулятора *TPM201* з потрібними реєстрами.

Параметри у програмі-конфігураторі розподілені по групах: Група *Init* – Основні параметри приладу, Група *Adv* – Параметри регулювання і *LBA*, Група *Comm* – Параметри обміну по *RS-485*, а також групи з доступом по паролі. Склад і кількість груп параметрів залежить від типу регулятора. Всі параметри в групах мають спеціальні *HASH*-імена. Наприклад, оперативний параметр «*PV*» передає вимірне значення на вимірювальному вході, а «*SP*» – уставку параметра для регулювання. Решта *HASH*-імена параметрів приладу представлені в інструкції з експлуатації.

Робота з програмою полягає у виборі та редагуванні значень параметрів конфігурації приладу. Збережіть та завантажте до регулятора конфігурацію, попередньо встановивши з ним зв'язок.

Після запуску програми-конфігуратора у вікні налаштувань необхідно налаштувати мережні параметри приладу, щоб ПЛК (ПК або ОП) міг зчитувати поточні параметри по інтерфейсу *RS-485*, наприклад: швидкість «9600», кадр «8-n-2», адрес «16» та номер *COM*-порту для завантаження файлу конфігурації. Потім для зв'язку регулятора з ПЛК у складі РСУ необхідно вибрати необхідний інтерфейс та протокол обміну. Після закінчення конфігурування пристрою збережіть та завантажте до нього конфігурацію. Обов'язково перезавантажте пристрій для переходу до робочого режиму, наприклад, для роботи за протоколом *ModBus-RTU*.

4.4.1.2 Програмування *ПЛК150* для взаємодії з *TPM201*. Для прикладу пропонується розробити проект з програмою користувача для дистанційного керування локальним регулятором серії *TRMxxx*, яка показана нижче:



У проєкті немає комунікаційного модуля та візуалізації. Їх потрібно додати та налаштувати. За допомогою інтерфейсу *RS-485* за протоколом *ModBus-RTU* контролер буде зчитувати значення параметра з регулятора, а також записувати значення уставки та тип логіки роботи регулятора. У програмі передбачений вибір двох значень уставки (35 °C

або 45 °C) за допомогою змінної *ust* типу *bool*, значення якої вводиться за допомогою кнопки на візуалізації. Також за допомогою елемента візуалізації вводиться код типу логіки, який передається до регулятора.

Отже, створіть у середовищі *CoDeSys V2.3* новий проєкт. Перейдіть до вкладення «Ресурси» та оберіть утиліту «Конфігурація ПЛК». Додайте та проведіть конфігурування програмного модуля *ModBus-Master*. У модулі *Modbus (Master) [VAR]* замініть слот *Debug RS-232 [SLOT]* на слот з інтерфейсом *RS-485-1* з наступними параметрами інтерфейсу: «9600», «8-n-2» (ці параметри відповідають налаштуванням регулятора *TPM201* для режиму обміну *ModBus-RTU*) та «RTU».

Далі додайте до модуля *Modbus (Master) [VAR]* субмодуль *Universal Modbus device [VAR]*, який є віртуальною моделлю підлеглого пристрою, тобто, регулятора *TPM201*, який підключений до *ПЛК150* по інтерфейсу *RS-485*. Налаштуйте параметри субмодуля у відповідності з екранною формою, яка представлена нижче. Зверніть увагу на адресу підлеглого пристрою – «16», тип зв'язку – «Serial» та послідовність слідування байт – «Native» (для правильного розкодування чисел з плаваючою комою згідно стандарту *IEEE754*, які передаються двома послідовними регістрами).

Аналоговий вихід [FIX]
Аналоговий вихід [FIX]
ModBus (Master) [VAR]
 AT %QD9.0: DWORD; (* Last e...
 AT %QW9.1: WORD; (* Last e...
 RS-485-1[SLOT]
 Universal Modbus device [VAR]
 AT %QB9.1.0: BYTE; (* C...

Индекс	Имя	Значение	По умолчанию	Мин.	Макс.
1	ModuleIP	10.0.0.223	10.0.0.223		
2	Max timeout	150	150	10	
3	TCPport	502	502		
4	NetMode	Serial	Serial		
5	ModuleSlaveAddress	16		0	255
6	Work mode	By poll time	By poll time		
7	Polling time ms	1000	100	10	10000
8	Visibility	No	No		
9	Amount Repeat	0	0	0	100
10	Byte Sequence	Native	Trace_mode		

Для обміну даними до програмного модуля *Universal Modbus device [VAR]* необхідно додати та налаштувати регістри даних для зберігання значень мережних змінних. У даному випадку це елемент типу *Real input module [VAR]* призначений для отримання та зберігання значення поточного вимірюного параметра, який отримується від

регулятора, та елементи Register output module [VAR], які призначені для запису до регулятора значення уставки та типу логіки його роботи.

```
└─ ModBus (Master) [VAR]
  ├── AT %QD9.0: DWORD; (* Last address *) [CHAN
  ├── AT %QW9.1: WORD; (* Last error *) [CHANN
  └── RS-485-1[SLOT]
    └─ Universal Modbus device[VAR]
      ├── AT %QB9.1.0: BYTE; (* Command (0xff -
      │   └─ Real input module[VAR]
      │       └─ parametr AT %ID9.1.0.0: REAL; (*
      │   └─ Register output module[VAR]
      │       └─ ust_mb AT %QW9.1.1.0: WORD; (* *)
      └─ Register output module[VAR]
          └─ alt_mb AT %QW9.1.2.0: WORD; (* *)
```

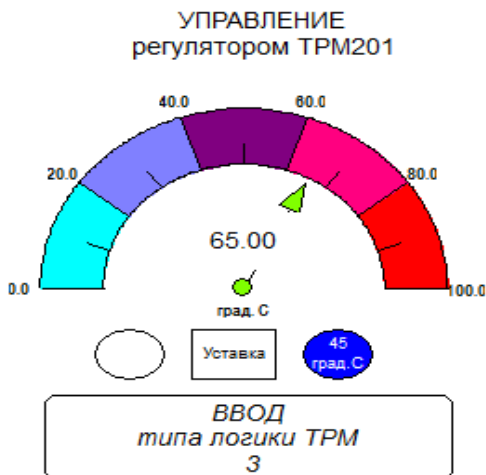
Номери регістрів у приладі можна визначити згідно з таблицею регістрів регулятора *TPM201* з інструкції з експлуатації. Наприклад, для зберігання поточного значення параметра це регістри $0x1009$ и $0x100A$, які відповідають передаванню числа з плаваючою точкою у форматі *IEEE754*. Зверніть увагу, що номер регістра у *Modbus*-модулі вказується у десятковому обчисленні, тобто числу 1009_{16} відповідає число 4105_{10} .

Збережіть проєкт на жорсткому диску, скопіюйте його та завантажте код до ПЛК. Запустіть проєкт на виконання та при необхідності налагодьте готовий проєкт. Перевірте працездатність макета РСУ.

4.4.1.3 Розроблення візуалізації проєкту для відображення мережних змінних. Для створення візуалізації перейдіть до вкладення «Візуалізація» у менеджері проєктів у середовищі *CoDeSys V2.3* та створіть об'єкт для візуалізації з ім'ям *PLC_VISU*. Далі, використовуючи інструменти для створення елементів візуалізації, створіть графічні об'єкти з необхідними налаштуваннями компонентів.

Для відображення значення параметрів (наприклад, поточне значення у вимірювальному каналі приладу *TPM201*) створіть графічний елемент «Прямоугольник». Подвійним кліком ЛКМ

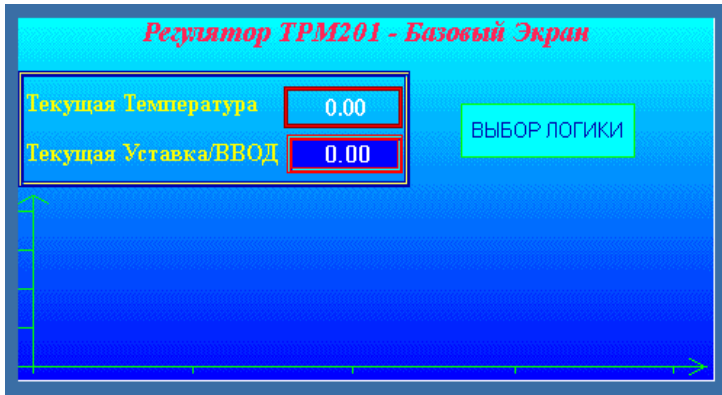
відкрийте бланк налаштування елемента. У категорії «Текст» введіть формат виведення значення змінної – «%3.2f», у категорії «Цвета» налаштуйте кольори фона для нормального та аварійного значення змінної, в категорії «Переменные» в рядок «Вывод текста» введіть символічне ім'я змінної. Аналогічно створіть і налаштуйте графічні елементи для введення та відображення значення уставки і типу логіки.



Отже, створіть в проєкті візуалізацію, наприклад, таку яка зображена на рисунку вище. При необхідності, використайте вбудовану у *CoDeSys V2.3* довідку, розділ «Визуализация».

Знову збережіть проєкт, скопіюйте його та завантажте код до ПЛК. Запустіть проєкт на виконання. Спостерігайте за оперативними параметрами та управляйте регулятором.

4.4.1.4 Програмування панелі оператора *СПЗхх* для взаємодії з *ТРМ201*. Для дистанційного керування регулятором можливе використання панелі оператора, наприклад, панелі *СПЗ07*. Раніше, у п.3.4.2, було розглянуто принципи програмування панелі для її сумісної роботи з контролером. Подібно до вказаного пункту розробіть проєкт для керування регулятором *ТРМ201*. Потрібні номери регістрів зазначені в інструкції з експлуатації. Приклад екрана панелі наведений нижче:



4.4.2 Розроблення PCY на прикладі взаємодії ПЛК, ОП та ПЗО

4.4.2.1 Налаштування пристроїв зв'язку з об'єктом для роботи у складі PCY. Для інтегрування модулів до складу PCY необхідно:

- по-перше, за допомогою програми-конфігуратора налаштувати інтерфейс та протокол обміну даними приладу з ПЛК (ПК або ОП);
- по-друге, провести конфігурування входів та виходів модулів у відповідності зі схемою електричних з'єднань.

Усі пристрої зв'язку з об'єктом призначені для розширення каналів введення/виведення контролерів за допомогою послідовного інтерфейсу *RS-485* в PCY. Усі модулі працюють у мережі *RS-485* лише при наявності у ній головного пристрою за протоколами *MODBUS (ASCII, RTU)*, *DCON* або *OWEN*.

Наприклад, модуль *MBA8* – це 8-канальний універсальний вимірювальний модуль аналогового введення, а модуль *МДВВ* – багатоканальний модуль дискретного введення/виведення. Для мережного обміну з модулями користувач може вибрати будь-який протокол. Основні функції модулів описані в інструкції з експлуатації. Для підготовки модуля до роботи у складі PCY необхідно підключити його через ПІ, наприклад *USB/RS-485* (або аналогічний) до ПК та підключити до приладу живлення.

Конфігурування приладів проводиться за допомогою відповідних програм-конфігураторів. Конфігурування модуля – це налаштування мережних параметрів та мережного інтерфейсу (конфігурація приладу

– це повний набір значень параметрів приладу, який визначає його роботу та функції). Конфігурування модулів *ОВЕН* здійснюється лише за протоколом *OWEN*. Основні можливості та порядок роботи з програмами-конфігураторами можна побачити у довідковій системі програми-конфігуратора або у керівництві з експлуатації.

Після запуску програми конфігурування у вікні налаштування мережних параметрів необхідно задати мережні параметри для приладу, щоб ПЛК міг зчитувати поточні параметри по інтерфейсу *RS-485*, наприклад, з наступними параметрами інтерфейсу: «9600», «8-n-1» та «RTU». Причому, адреса пристрою за умовчанням – «16», а *COM*-порт визначається за допомогою диспетчера пристроїв операційної системи.

ПРИМІТКА: Якщо робота виконується на ПК, до якого стенд з потрібним модулем не підключений, то необхідно вибрати варіант «Работать OFFLINE». Номер *COM*-порта ПК з підключеним стендом призначити виходячи з параметрів меню ПК «Панель управління» (*COM*-порт емулюється на фізичному *USB*-порту ПК). У цьому випадку за допомогою програми-симулятора *ModBus*-пристрою *ModSim32* створить віртуальну модель відповідного модуля з потрібними регістрами.

Далі, згідно з довідковою системою програми-конфігуратора модуля або керівництва з експлуатації, зробіть конфігурування каналів модуля відповідно до принципової схеми стенда. Збережіть конфігурацію та завантажте її до модуля. Обов'язково перезавантажте пристрій для переходу до робочого режиму, наприклад, для роботи за протоколом *ModBus-RTU*.

Робота модулів за протоколом *ModBus* може йти у режимах *ASCII* або *RTU*, в залежності від значення параметра *Prot*. За протоколом *ModBus* можна отримати значення вхідних каналів, час вимірювання (для аналогових каналів) та його статус. Зчитування йде за стандартними для протоколу *ModBus* командами читання групи регістрів (команди «03» або «04»).

Результати вимірювання представляються у двох форматах: 4-х байтових значеннях з плаваючою точкою (без часу) або 2-х байтове ціле. У останньому випадку отримане ціле число необхідно поділити на

число «10» у степені, яка задана у параметрі dP. Використання цілого числа та його зміщення потребує менш ресурсів під час оброблення, ніж використання числа з плаваючою точкою, яке передається двома регістрами та потребує програмного перетворення.

Час вимірювання – це циклічний час роботи АЦП з кроком 0.01 сек, яке закодовано у двох байтах. Це значення точно відповідає часу проведення вимірювання у даному каналі та під час роботи з ним (наприклад, при обчисленні диференціальної складової у процесі ПІД-регулювання) можна не враховувати затримку передачі по мережі RS-485. Відлік циклічного часу починається при включенні приладу та кожні 65536 тактів (що відповідає 655.36 сек) час скидається до нульового значення. Статус вимірювання, тобто код похибки вимірювання знаходиться у відповідному регістрі.

Перелік регістрів для зберігання конфігураційних, мережних та оперативних параметрів приладів та їхнього опису відповідно до протоколу *Modbus* представлений у керівництва з експлуатації.

4.4.2.2 Програмування *ПЛК150* для зв'язку з модулем *MBA*. Для прикладу пропонується розробити проєкт з програмою користувача з двохпозиційним регулятором для управління процесом нагрівання або охолодження.

Нижче представлений лістинг програми користувача:

```
PROGRAM PLC_PRG
VAR
    ust1: REAL:=60;
    ust2: REAL:=50;
END_VAR

IF temperatura>ust1 THEN
    ten:=FALSE;
    vent:=TRUE;
    lamp:=TRUE;
END_IF
IF temperatura<ust2 THEN
    ten:=TRUE;
    vent:=FALSE;
    lamp:=FALSE;
END_IF
```

У програмі дискретні виходи ПЛК зв'язані зі змінними *lamp*, *vent* та *ten*, а їх стан визначається порівнянням з уставками *ust1* та *ust2* поточного значення змінної *temperatura*, яка одержана по мережі *RS-485* від модуля *MBA*.

Отже, створіть в середовищі *CoDeSys V2.3* новий проєкт, перейдіть у вкладення «Ресурси» та виберіть утиліту «Конфігурація ПЛК». Додайте та проведіть конфігурування програмного модуля *Modbus-Master*. У модулі *Modbus (Master) [VAR]* замініть слот *Debug RS-232 [SLOT]* на слот з інтерфейсом *RS-485-1* з наступними параметрами інтерфейса: «9600», «8-n-1» та «RTU».

Далі додайте до програмного модуля *Modbus (Master) [VAR]* субмодуль *Universal Modbus device [VAR]*, який представляє собою віртуальну модель підлеглого пристрою, тобто, модуля *MBA*, який з'єднаний з *ПЛК150* інтерфейсом *RS-485*. Налаштуйте параметри субмодуля. Це адреса підлеглого пристрою – *16*, тип каналу зв'язку – *Serial* та послідовність слідування байтів – *Native*.

Для обміну даними в універсальному *Modbus*-модулі необхідно створити та налаштувати реєстри даних для зберігання значень мережних змінних. У даному випадку це буде реєстр типу *Real input module [VAR]*, який потрібен для зберігання значення температури, яка одержана від модуля *MBA*. Номер реєстра буде дорівнювати значенню «22», що відповідає 4-му вхідному каналу, до якого підключений термометр опору типу *ТСП100*:

Инде...	Имя	Значение
1	Register Address	22
2	Command	Read holding Registers (Ok...
8	Visibility	No

Збережіть проєкт на жорсткому диску, скопіюйте його та завантажте код до ПЛК. Запустіть проєкт на виконання. Спостерігайте за значенням температури та роботою макета РСУ.

4.4.2.3 Розроблення візуалізації проєкту для відображення значень мережних змінних. Для створення візуалізації перейдіть до вкладення «Визуализация» менеджера проєктів. Далі, використовуючи інструменти для створення елементів візуалізації, створіть графічні елементи з необхідними налаштуваннями компонентів.

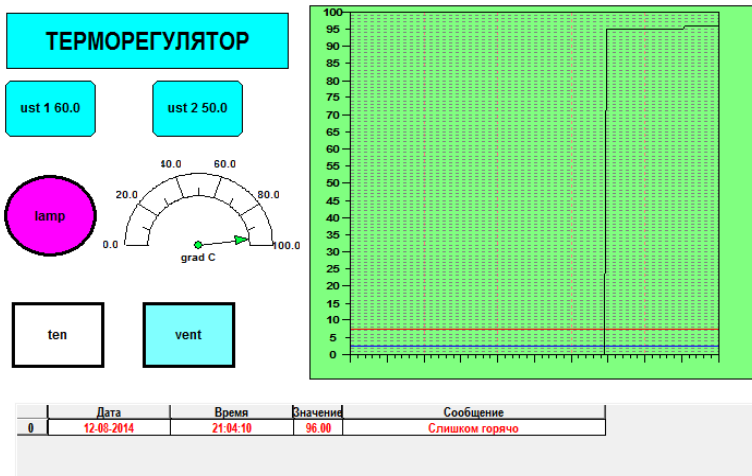
Для відображення значення параметрів у каналах модуля *MBA* створіть графічний елемент «Прямоугольник». Відкрийте бланк налаштувань елемента. У категорії «Текст» введіть формат виведення значення змінної – «%3.1f», у категорії «Цвета» налаштуйте кольори фону для нормального та аварійного значення змінної, у категорії «Переменные» в рядку «Вывод текста», використовуючи «Ассистент ввода», введіть символічне ім'я змінної. Аналогічно створіть та налаштуйте графічні елементи для відображення значення інших каналів модулів.

Для відображення значення поточного параметра додайте та налаштуйте стрілочні індикатори та вікно з трендами. Для стрілочних індикаторів залиште налаштування за замовчуванням. У вікні конфігурації ГЭ «Тренд» у категорії «Тренд» ЛКМ натисніть на кнопку «Выбор переменной» та потім у вікні, що з'явилось, за допомогою асистента (функціональної клавіші F2) додайте потрібну змінну.

Для виведення повідомлень та тривог створіть та налаштуйте бланк журналу тривог. Цей компонент дозволяє стежити за перебуванням вимірюваного параметра у заданому діапазоні значень. Для цього користувач у створених посиланнях визначає верхню та нижню межі діапазону контролю. При виході вимірюваного параметра за вказані межі журнал тривог видає попередження («Аларм»). «Аларми» виводяться в спеціалізованих вікнах, колір яких змінюється в залежності від типу повідомлення.

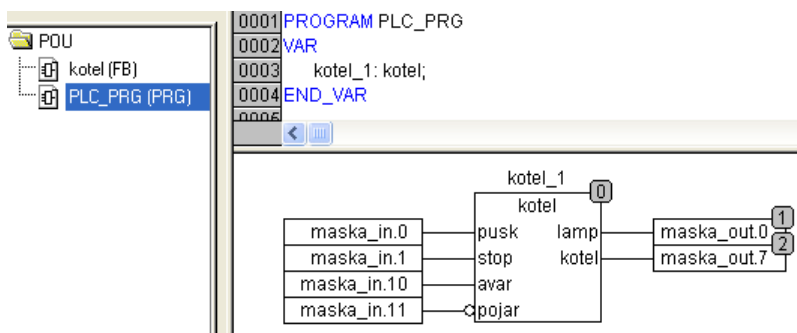
Отже, створіть у проєкті візуалізацію мережного обміну, наприклад, подібну до зображеної нижче на рисунку. При необхідності, використовуйте вбудовану в *CoDeSys* довідку, розділ «Визуализация».

Знову збережіть проєкт, скопіюйте його та завантажте код до ПЛК. Запустіть проєкт на виконання та перевірте працездатність макета РСУ.

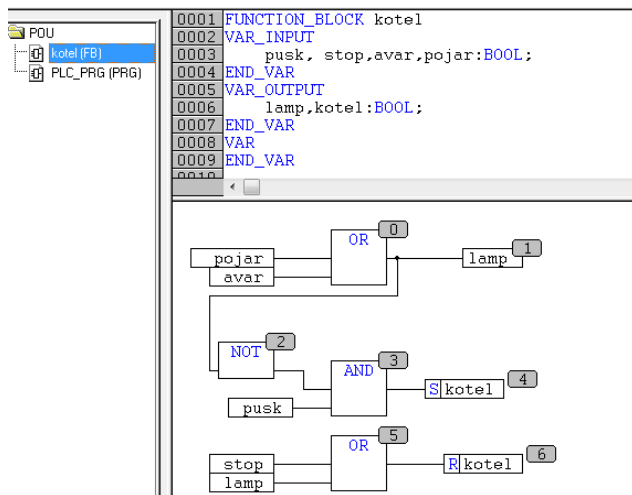


4.4.2.4 Програмування *ПЛК150* для взаємодії з модулем *МДВВ*. Для прикладу пропонується розробити проєкт з програмою користувача для дистанційного керування котлом з аварійним блокуванням.

Нижче представлений варіант програми користувача, який складається з двох *POU* – головної програми *PLC_PRG* та функціонального блока *kotel*.



У програмі за допомогою масок входів та виходів здійснюється обмін по мережі *RS-485* між *ПЛК150* та модулем *МДВВ*, до якого підключені кнопки управління, датчики та вихідні елементи. Усередині ФБ *kotel* реалізований алгоритм управління котлом:



Отже, створіть в середовищі *CoDeSys V2.3* проєкт, перейдіть до вкладення «Ресурси» та виберіть утиліту «Конфігурація ПЛК». Додайте та проведіть конфігурування програмного модуля Modbus-Master. В модулі Modbus (Master) [VAR] замініть слот Debug RS-232 [SLOT] на слот з інтерфейсом *RS-485-1* з наступними параметрами: «9600», «8-n-1» та «RTU».

Далі додайте до модуля Modbus (Master) [VAR] субмодуль Universal Modbus device [VAR], який представляє собою віртуальну модель підлеглого пристрою, тобто, модуля *МДВВ*, який підключений до *ПЛК150* по інтерфейсу *RS-485*. Налаштуйте параметри субмодуля: адресу пристрою – «16», тип зв'язку – «Serial» та послідовність слідування байтів – «Native».

Для обміну даними в універсальному *Modbus*-модулі необхідно створити та налаштувати реєстри даних для зберігання значень мережних змінних: маски входів та маски виходів. У даному випадку елемент типу Register input module [VAR] призначений для зберігання значення маски входів модуля *МДВВ*, а елемент типу Register output module [VAR] – для керування виходами модуля *МДВВ*.

Відправлення групової команди вмикання або вимикання вихідних пристроїв здійснюється у реєстр із номером 50_{10} ($0x32$). У

регістр записується число від 0 до 255 у двійковому вигляді, кожен біт значення відповідає стану номеру дискретного вихідного пристрою у модулі *МДВВ* (номер розряду відповідає номеру каналу). Значення розряду відповідає стану «*ввімкнено*» для вихідного пристрою, якщо воно дорівнює стану «*TRUE*». Для запису значення до регістру використана функція $0x10_{16}$, яка здійснює груповий запис.

Читання стану входів модуля *МДВВ* реалізовано через регістр з номером 51_{10} ($0x33$):

The image shows a screenshot of a software interface. On the left is a ladder logic diagram with the following rungs:

- Rung 1: ModBus (Master) [VAR] with a normally open contact labeled "AT %QD9.0: DWORD; (* Last ac".
- Rung 2: ModBus (Master) [VAR] with a normally open contact labeled "AT %QW9.1: WORD; (* Last er".
- Rung 3: RS-485-1[SLOT] with a normally open contact labeled "AT %QB9.1.0: BYTE; (* Co".
- Rung 4: Universal Modbus device [VAR] with a normally open contact labeled "AT %QB9.1.0: BYTE; (* Co".
- Rung 5: Register input module [VAR] with a normally open contact labeled "maska_in AT %IW9.1.0.0".

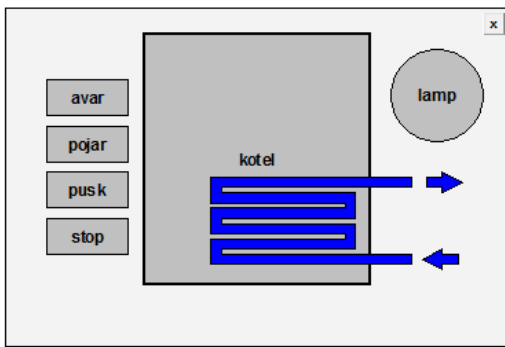
On the right is a window titled "Базовые параметры" (Basic parameters) with a sub-tab "Параметры модуля" (Module parameters). It contains a table:

Индекс	Имя	Значение
1	Register Address	51
2	Command	Read holding Registers (0x03)
8	Visibility	No

Збережіть проєкт, скопіюйте його та завантажте до ПЛК по інтерфейсу *Ethernet*. Запустіть проєкт на виконання та перевірте працездатність макета РСУ.

4.4.2.5 Розроблення візуалізації проєкта управління котлом. Для створення візуалізації перейдіть у вкладення «*Візуалізація*» менеджера проєктів. Далі, використовуючи інструменти для створення елементів візуалізації, створіть графічні елементи з необхідними налаштуваннями компонентів.

Отже, створіть в проєкті візуалізацію для мережного обміну, наприклад, яка зображена на рисунку:



Знову збережіть проєкт, скомпілюйте його та завантажите код до ПЛК. Запустіть проєкт на виконання та керуйте котлом.

4.4.2.6 Програмування панелі оператора *СПЗхх* для взаємодії з *МВА та МДВВ*. Для відображення значення поточного параметра на 4-му каналі модуля *МВА* можна використати панель оператора, наприклад, панелі *СП307*. Раніше, у п.3.4.2, було розглянуто принципи програмування панелі для її сумісної роботи з контролером. Але, панель не може виконувати функцію регулятора безпосередньо. Тобто, вона може бути головним пристроєм у мережі *RS-485* та здійснювати запити до модулів. Проте, вона не має можливості розраховувати керуючий вплив для виконавчих пристроїв за різними законами регулювання. Власне така функція можлива, якщо у конфігурацію панелі додати відповідний скрипт. Але написання скрипту це достатньо складний процес, який потребує значних зусиль та спеціальних навичок у програмуванні.

Подібно до п. 3.4.2 розробіть проєкт для панелі оператора для відображення значення регістрів усіх 8-і каналів модуля *МВА*:

	1	2	3	4	5	6	7	8
Целое Значение	0	0	0	0	0	0	0	0
Смещение Точки	0	0	0	0	0	0	0	0
Значение REAL	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Цикл. Время Измерения	0	0	0	0	0	0	0	0
Код Ошибки	0	0	0	0	0	0	0	0

Також, подібно до п. 3.4.2 розробіть проєкт для панелі оператора для відображення стану входів та керування виходами модуля *МДВВ*:

Состояние входов модуля МДВВ

Управление выходами модуля МДВВ

Вкл

Вкл

Вкл

Вкл

Вкл

Вкл

Вкл

Вкл

Выкл

Выкл

Выкл

Выкл

Выкл

Выкл

Выкл

Выкл

4.4.3 Розроблення PCS на прикладі взаємодії СПЗхх та PP110/114

4.4.3.1 Програмування реле PP110 для роботи у складі PCS.

Програмоване реле PP110 призначене для побудови простих автоматизованих систем управління, а також для заміни релейних систем захисту та контролю.

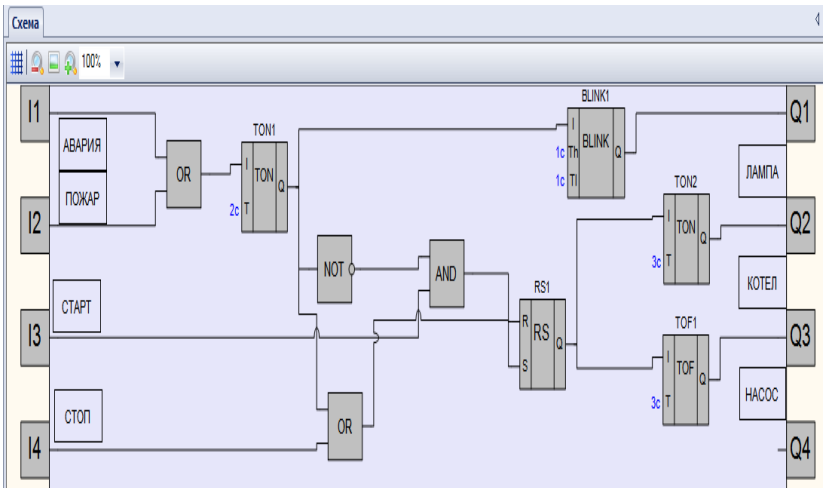
Основні технічні характеристики, опис вхідних сигналів і характеристики вбудованих вихідних елементів реле наведені в керівництві з експлуатації.

Логіка роботи приладу PP110 визначається користувачем в процесі програмування за допомогою середовища OWEN Logic. Створення проєкту в середовищі OWEN Logic зводиться до наступних дій:

- 1) Вибір цільової платформи (вибір з переліку *Target*-файлів реле).
- 2) Написання програми користувача (тобто, створення алгоритму).
- 3) Збереження проєкту.
- 4) Компілювання та налагодження проєкту на емуляторі.
- 5) Налаштування параметрів мережного обміну за протоколом *ModBus*.

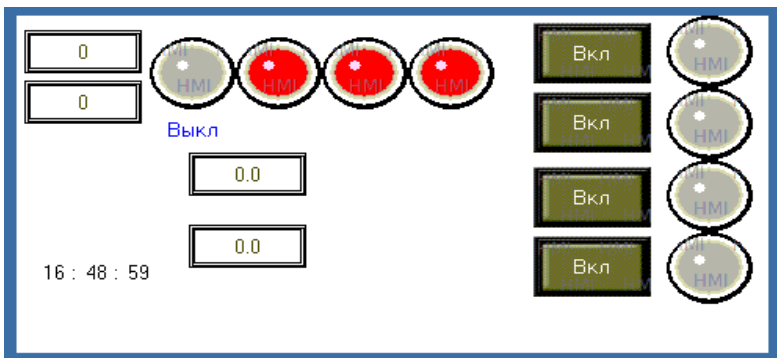
ModBus.

У лабораторній роботі розглядається лише мережна взаємодія за протоколом *ModBus*. Тому залишимо перші чотири пункти порядку створення проєкту у середовищі OWEN Logic. До реле PP110 вже завантажений алгоритм, представлений нижче.



У даному прикладі розглянуте керування абстрактною котельнею. Вона складається з насоса, пальника, кнопок управління, датчика пожежі, аварійної кнопки та сигнальної лампи. При натисканні на кнопку «Старт» повинен ввімкнутись насос, а з деякою затримкою повинен запрацювати пальник. При натисканні кнопки «Стоп» пальник повинен припинити роботу, а з деякою затримкою повинен вимкнутись насос. Якщо спрацьовує датчик пожежі або включили аварійну кнопку, то робота установки припиняється (аналогічно натискання на кнопку «Стоп»). Робота котла не може бути відновлена, поки ці сигнали не зникнуть (пожежа загашена або кнопка вимкнена).

4.4.3.2 Програмування панелі оператора *СПЗхх* для взаємодії з *ПР110/114*. Суть мережного обміну з *СП310* полягає у отриманні станів входів та виходів реле без втручання у його роботу з боку панелі оператора. Для цього у керівництві з експлуатації представлений перелік регістрів для зберігання стану входів та виходів відповідно до протоколу *Modbus* як окремо, так і у вигляді маски. Крім того, в документації представлені параметри налаштування інтерфейсу реле для роботи по протоколу *Modbus* для режимів *ASCII* или *RTU*. Для розроблення проєкту скористайтесь матеріалами п.3.4.2. Основна відмінність модуля реле *ПР110* від модуля *МДВВ* це наявність керуючої програми всередині реле. Така функція неможлива у модулі *МДВВ*. Тому екран панелі буде відображати лише стан входів та виходів:



4.5. Оформлення результатів проведення заняття

Складіть звіт про проведенне заняття, де наведіть схему з'єднань та

умови обміну даними. Також вкажіть, які режими використовувалися для обміну даними між пристроями.

4.6. Контрольні запитання

1) Яке призначення регуляторів, модулів вводу-виводу і логічних реле при їх використанні у складі РСУ?

2) Яким чином налаштовуються периферійні пристрої для організації мережної взаємодії за протоколом *ModBus*?

Лабораторна робота 5

ОСНОВНІ ПРИНЦИПИ ФОРМУВАННЯ ЗАПИТІВ ТА АНАЛІЗ ВІДПОВІДЕЙ У ПРОТОКОЛІ *OWEN*

5.1 Мета проведення лабораторної роботи

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

Мета роботи:

- закріпити на практиці отримані теоретичні відомості про протокол *OWEN* на послідовному інтерфейсі *RS-485*;
- ознайомитися з реалізацією протоколу *OWEN* на послідовному інтерфейсу *RS-485* у *ПЛК150* через конфігурування його ресурсів;
- навчитися інтерпретувати *OWEN*-запити та *OWEN*-відповіді.

5.2 Апаратне забезпечення лабораторної роботи

Лабораторна робота проводиться на робочих місцях, де на стендах є *ПЛК150*. Так як ПК не мають вбудованого інтерфейсу *RS-485*, то для зв'язку з ПЛК застосовуються перетворювачі інтерфейсів *RS-232/RS-485* або *USB/RS-485*. Схему зв'язків наведено у п.1.2 ЛР №1.

5.3 Програмне забезпечення лабораторної роботи

Для зв'язку ПК та ПЛК по послідовному інтерфейсу за допомогою ПИ *USB/RS-485* в ОС *Windows* за допомогою драйвера емулюється віртуальний *COM*-порт, з яким без обмежень може працювати програма *COM Port Toolkit*. Однак ця програма не може

інтерпретувати запити протоколу *OWEN*. Для аналізу кадрів відповіді за протоколом *OWEN* від підлеглого пристрою можна використовувати програму *OWEN Test Console*, яка емує роботу головного пристрою.

5.4. Короткі теоретичні відомості:

Протокол *OWEN* розроблений для реалізації обміну інформацією приладів *OBEH* між собою та ПК у мережі *RS-485* та *RS-232* (рідше). Протокол *OWEN* має зручну організацію для конфігурації приладів.

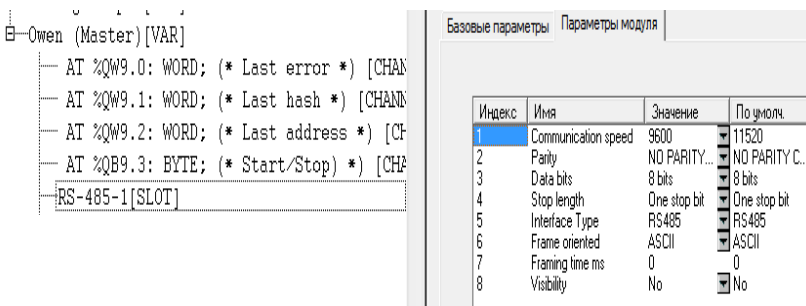
OWEN – це протокол обміну даними та розроблений для ПЛК однойменний програмний модуль, який забезпечує роботу ПЛК у відповідності до цього протоколу. Для доступу до даних ПЛК із застосуванням протоколу *OWEN* по послідовним інтерфейсам використовує технологію «головний-підлеглий», при якій лише один пристрій (головний) може ініціювати передачу (зробити запит). Інші пристрої (підлегли) передають запитувані головним пристроєм дані або виробляють запитувані дії.

Для обміну даними за протоколом *OWEN* у ПЛК150 існує два варіанти реалізації: за допомогою бібліотеки *OwenNet.lib* або за допомогою конфігурування ресурсів ПЛК. В останньому випадку до стандартної конфігурації додаються програмні модулі *Owen (Master)* або *OWEN (slave/spy)*. За допомогою цих модулів реалізується обмін даними з ПЛК у статусі головного або підлеглого (режим *spy* подібний до режиму *slave*, але у цьому випадку пристрій лише слухає і не відповідає на запити). При цьому у проєкті, який створений в середовищі *CoDeSys* для ПЛК, програма користувача може бути порожньою. До складу зазначених модулів включені модулі послідовних інтерфейсів, які налаштовує користувач. Крім того, користувач додає потрібні змінні для обміну даними. Лабораторна робота складається з двох частин: спочатку ПЛК виконує функцію головного пристрою та формує запити підлеглим. Запити приймає програма *COM Port Toolkit*. Друга частина роботи полягає у тому, що ПЛК виконує функцію підлеглого пристрою. Запити робить програма *OWEN Test Console*.

5.5. Послідовність проведення заняття

5.5.1. Програмування ПЛК для роботи у режимі головного пристрою

Створіть у середовищі *CoDeSys V2.3* новий проект, визначте необхідний таргет-файл та мову реалізації головного *POU PLC_PRG*. Як цільову платформу для ПЛК оберіть *OWEN150-IL*, а мову програмування оберіть *CFC*. Контролер буде лише опитувати модуль, тому вікно *POU PLC_PRG* залиште порожнім. Далі перейдіть до вкладення «Ресурси» та виберіть утиліту «Конфігурація ПЛК». До неї додайте програмний модуль *Owen (Master) [VAR]*, а в ньому змініть слот з інтерфейсом *RS-232* на *RS-485* з такими параметрами інтерфейсу: швидкість – «9600», формат кадру – «8-n-1» та режим передачі – «ASCII», як це показано на рисунку:



Индекс	Имя	Значение	По умолч.
1	Communication speed	9600	11520
2	Parity	NO PARITY...	NO PARITY C...
3	Data bits	8 bits	8 bits
4	Stop length	One stop bit	One stop bit
5	Interface Type	RS485	RS485
6	Frame oriented	ASCII	ASCII
7	Framing time ms	0	0
8	Visibility	No	No

Додайте мережні змінні. Наприклад, змінну для читання з приладу поточного значення параметра (*PV*) та змінну для запису до приладу значення уставки (*SP*). Налаштуйте параметри мережних змінних, вкажіть адресу – «16», *HASH-ім'я*: «*PV*» та «*SP*», а також тип змінних – «*Float*». Інші параметри залиште без змін.

Для зручності введення значень уставок та відображення значення створіть візуалізацію проекту. Скомпілюйте проект, встановіть з'єднання з ПЛК за допомогою програмного комунікаційного модуля *Codesys Gateway* та завантажте код проекту до ПЛК. Запустіть ПЛК на виконання програми користувача та не переривайте зв'язок середовища *CoDeSys* з ПЛК для спостереження за змінними у проекті та введення значень уставки.

```

└─ Owen (Master) [VAR]
  └─ AT %QW9.0: WORD; (* Last error *) [CHANNEL
  └─ AT %QW9.1: WORD; (* Last hash *) [CHANNEL (
  └─ AT %QW9.2: WORD; (* Last address *) [CHANNE
  └─ AT %QB9.3: BYTE; (* Start/Stop *) [CHANNEL
  └─ RS-485-1[SLOT]
  └─ Float variable (Listen) [VAR]
    └─ AT %QD9.1.0: REAL; (* *) [CHANNEL (Q)]

```

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолчанию
1	Address length	8 bit	8 bit
2	Address	16	0
3	Hash name	PV	
4	Index	0	0
5	Use a index?	No	No
6	Float type	Float	Float
7	Precision	2	2
8	Polling time ms	100	100
9	Work Mode	Polling Time	Polling Time
10	Repiat counter	0	0
15	Visibility	No	No

```

└─ Owen (Master) [VAR]
  └─ AT %QW9.0: WORD; (* Last error *) [CHANNEL
  └─ AT %QW9.1: WORD; (* Last hash *) [CHANNEL (
  └─ AT %QW9.2: WORD; (* Last address *) [CHANNE
  └─ AT %QB9.3: BYTE; (* Start/Stop *) [CHANNEL
  └─ RS-485-1[SLOT]
  └─ Float variable (Listen) [VAR]
    └─ AT %QD9.1.0: REAL; (* *) [CHANNEL (Q)]
  └─ Float variable (Write) [VAR]
    └─ AT %QD9.2.0: REAL; (* *) [CHANNEL (Q)]

```

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолчанию
1	Address length	8 bit	8 bit
2	Address	16	0
3	Hash name	SP	
4	Index	0	0
5	Use a index?	No	No
6	Float type	Float	Float
7	Precision	2	2
8	Polling time ms	100	100
9	Work Mode	Polling Time	Polling Time
10	Repiat counter	0	0
15	Visibility	No	No

Для приймання запитів запустіть на ПК програму моніторингу COM-порта *COM Port Toolkit*. Оберіть потрібний COM-порт та налаштуйте його параметри згідно з налаштуваннями комунікаційного модуля *RS-485* у проекті для ПЛК, а саме: швидкість – «9600», формат кадру – «8-n-1» та режим передачі – «ASCII». Спостерігайте за прийнятими кадрами у вікні програми.

Отже, виходячи з проекту до ПЛК, мережний модуль формує два запити: на читання та на запис. Для розшифрування запитів використаємо офіційний опис протоколу *OWEN*, який можна завантажити з сайту www.owen.ua.

Програма моніторингу COM-порта одержить два кадри. Це, кадр №1 – 23 48 47 48 47 52 4F 54 56 52 53 49 51 0D – всього 14 байт, та кадр №2 – 23 48 47 47 4B 50 48 47 4E 47 47 47 47 47 47 47 47 4A 4F 55 56 0D – всього 22 байти.

Обидва кадри містять два службових символи-роздільники кадрів на початку та кінці посилки. Це символ початку кадру – 23_{hex} (#) та символ кінця кадру – $0D_{\text{hex}}$ (<CR> – символ повернення каретки). Відкинемо їх, так як вони не входять до контрольної суми. Тепер відповідно до таблиці в опису протоколу *OWEN* перетворимо отримані символи за схемою «ASCII-символ у тетраду», потім «склеїмо» тетради у байти та перетворимо байти у кадр прикладного рівня, тобто отримаємо адресу пристрою у мережі та *HASH*-ім'я мережної змінної.

Результат розшифрування кадру №1 буде таким:

48	47	48	47	52	4F	54	56	52	53	49	51
0001	0000	0001	0000	1011	1000	1101	1111	1011	1100	0010	1010
10_{hex}		00010000		B	8	D	F	B	C	2	A
16_{10}		Виставлений <i>біт запиту</i> , кількість байт поля даних (- 2)		B8		DF		BC		2A	
				B8DF – <i>HASH</i> - ім'я змінної з ім'ям $PV = \langle 50 \rangle \langle 62 \rangle$				CRC – контрольна сума			

Результат розшифрування кадру №2 буде таким:

48	47	47	4B	50	48	47	4E	47	47	47	47
0001	0000	0000	0100	1001	0001	0000	0111	0000	0000	0000	0000
10_{hex}		00000000		9	1	0	7	0	0	0	0
16_{10}		<i>біт запиту</i> скинутий кількість байт поля даних (- 2+4)=6		91		07		00		00	
				9107 – <i>HASH</i> - ім'я змінної з ім'ям $SP = \langle 56 \rangle \langle 50 \rangle$				Поле з даними – значення змінної у форматі <i>IEEE 754</i>			

Продовження таблиці:

47	47	47	47	4A	4F	55	56
0000	0000	0000	0000	0011	1000	1110	1111
00		00		3	8	E	F
Поле з даними – значення змінної у форматі <i>IEEE 754</i>				38		EF	
				CRC – контрольна сума			

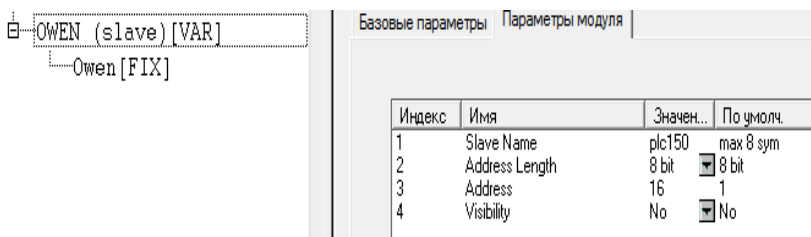
Приклад запису ненульового значення уставки (45,25) до приладу показаний нижче, де кутувими дужками виділено закодоване значення уставки:

23 48 47 47 4A 50 48 47 4E < 4B 49 4A 4C 47 47 > 54 51 4E 4E 0D .

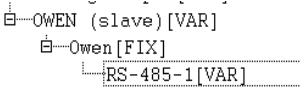
Відзначимо, що в даному випадку змінна має тип *Pic-Float*, тобто складається з трьох байт та має скорочену мантису (таке представлення має незначний вплив на 5-ий знак після коми). Така точність при передачі даних цілком достатня.

5.5.2 Програмування ПЛК для роботи у режимі підлеглого пристрою

Повторіть дії, які викладені у пункті 5.5.1, однак замість головного пристрою додайте до конфігурації модуль підлеглого пристрою. Це буде програмний модуль OWEN (slave) [VAR]. У його параметрах обов'язково вкажіть його адресу «16» та ім'я, наприклад, «plc150».



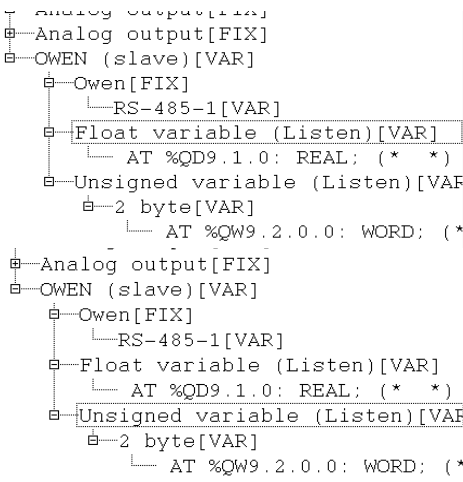
Далі додайте до мережного модуля комунікаційний інтерфейс RS-485-1 [VAR]. Параметри інтерфейсу встановіть у відповідності з пунктом 5.5.1, тобто «9600», «8-n-1», режим обміну – «ASCII».



Индекс	Имя	Значение	По умолч.
1	Communication speed	9600	11520
2	Parity	NO PARITY...	NO PARITY C...
3	Data bits	8 bits	8 bits
4	Stop length	One stop bit	One stop bit
5	Interface Type	RS485	RS485
6	Frame oriented	ASCII	ASCII
7	Framing time ms	0	0
8	Visibility	No	No

Далі додайте до модуля дві мережні змінні – типу *Float variable (Listen) [VAR]* та *Unsigned variable (Listen) [VAR]* з субмодулем 2 byte [VAR]. У даному випадку ПЛК буде імітувати роботу підлеглого пристрою, наприклад, модуля *MBA* та передавати по запиту значення та тип датчика на першому вимірювальному каналі. Для цього у налаштуваннях мережних змінних потрібно вказати їхні *HASH*-імена: «*EAD*» та «*in-t*», а також адресу першого каналу – «*16*», який співпадає з базовою адресою пристрою. Інші параметри залишить без змін.

В результаті буде отримана наступна конфігурація підлеглого пристрою з двома мережними змінними:

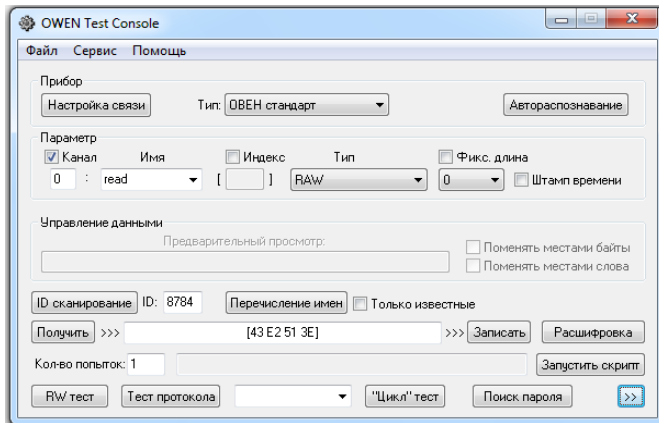


Индекс	Имя	Значение
1	Address length	8 bit
2	Address	16
3	Hash name	read
4	Index	0
5	Use a index?	No
6	Float type	Float
7	Precision	2
8	Polling time ms	100
9	Work Mode	Polling Time
10	Repiat counter	0
15	Visibility	No

Индекс	Имя	Значение
1	Address length	8 bit
2	Address	16
3	Hash name	in-t
4	Index	0
5	Use a index?	No
6	Precision	100
7	Work Mode	Polling Time
8	Repiat counter	0
15	Visibility	No

Для читання значень в підлеглому пристрої запустить програму *OWEN Test Console*. Після запуску програми необхідно провести налаштування послідовного порту, вибрати тип протоколу, вказати ім'я змінної.

Отже, у зоні робочого простору програми *OWEN Test Console* з ім'ям «Прибор» натисніть на кнопку «Настройка связи» та у діалоговому вікні «Установка параметров обмена» вкажіть тип III. У даному випадку тип III – автоматичний. Виберіть потрібний COM-порт та зробіть необхідні налаштування у зоні «Формат данных»: «9600», «8-n-1». У зоні «Сетевые параметры» введіть адресу підлеглого пристрою – «16». Інші параметри залишить за умовчанням. Для запису налаштувань натисніть на кнопку «ОК». Далі оберіть тип прибору – «ОВЕН стандарт», так як використовується протокол *OWEN*. У зоні «Параметр» оберіть тип представлення переданих даних – RAW (так звані «сирі» дані, тобто, необроблені), номер каналу – 0 та у полі «Имя» введіть ім'я змінної – «гEAd».



Якщо натиснути ЛКМ на кнопку «Получить», то можна отримати значення на першому вимірювальному каналі. У даному випадку це буде закодоване значення параметру (необроблене) – [43 E2 51 3E].

Якщо необхідно отримати значення розшифроване, то у зоні «Параметр» змініть тип RAW на тип FLOAT IEEE (PIC), який відповідає стандарту *IEEE 754* з відкинутим молодшим байтом мантиси. Істинне значення параметра буде дорівнює числу 452.6347 (перевірити самостійно). Можна вручну записати значення параметра за допомогою кнопки «Записать». У таблиці представлено десяткове число 452.6347 у 32-х бітному форматі *IEEE754*.

1 бит	8 бит	23 бит	<i>IEEE 754</i>
0	100 0011 1	110_0010_0101 0001_0011_1110	43E2_513E _{hex}
0(dec)	135(dec)	6 443 326 (dec)	452.6347 ₁₀
знак числа	зміщена експонента	залишок від мантиси	число у форматі <i>IEEE754</i>

Для прикладу запишіть нове значення до змінної у конфігурації ПЛК, а потім прочитайте його у «сирому» вигляді (код) та у розшифрованому вигляді (число). Зверніть увагу, що число приходить з великою кількістю знаків після коми. Це особливість формату стандарту *IEEE 754*.

Існує ще один варіант подання числа типу *FLOAT*. Знакове число з односторонньою десятковою крапкою у протоколі *OWEN* представляється формулою:

$$N = (-1)^S \cdot 10^{(Exponent)} \cdot Mantissa,$$

де *S* – визначає знак, займає один біт;

Exponent – десятковий порядок числа, який є числом без знака, та у якому зазначено зміщення, тобто, положення крапки (0 – для цілого числа, 1 – для числа з десятими частками, тощо), займає три наступних біта після біта знака;

Mantissa – ненормалізована мантиса, інші біти (довжина мантиси визначається розміром поля даних). Сумарна довжина повинна складати ціле число байтів.

Мантиса може виражатися як в двійковому, так і в двійково-десятковому вигляді. Наприклад, число (-10,38) може бути закодовано так:

- в двійковій-десятковому поданні мантиси – 0xA01038;

A	0	1	0	3	8
1__010	0000	0001	0000	0011	1000
S=1 EXP=2	Доповнює до цілої кількості байт (N=3)	1	0	3	8

- в двійковому поданні (на один байт коротше, N=2) – 0xA40E.

A	4	0	E
1__010	0100	0000	1110
S=1 EXP=2	1038		

5.6 Оформлення результатів проведення заняття

У контексті лекцій складіть звіт про проведене заняття, де уявіть схему з'єднань, умови обміну даними. Також вкажіть, які режими використовувалися для обміну даними.

5.7 Контрольні запитання

- 1) Які інтерфейси використовують для реалізації протоколу *OWEN*?
- 2) На яких рівнях моделі зв'язку побудований протокол *OWEN*?
- 3) Який формат кадру у протоколу *OWEN*?
- 4) Як забезпечується достовірність даних на каналному рівні протоколу *OWEN*?
- 5) Які принципи адресації закладені у протоколі *OWEN*?
- 6) Якого типу змінні описані у протоколі *OWEN*?
- 7) Як поділяють кадри при передаванні у протоколі *OWEN*?
- 8) У чому сутність кодування «ASCII-символ у тетраду»?
- 9) Що значить «склеювання» байтів у протоколі *OWEN*?
- 10) Яким чином підлеглий пристрій розпізнає запит у протоколі *OWEN*?

Лабораторна робота 6

ВЗАЄМОДІЯ КОНТРОЛЕРА ПЛК150 ТА ПАНЕЛІ ОПЕРАТОРА СМІ-1 ПО ІНТЕРФЕЙСУ RS-485 ЗА ПРОТОКОЛОМ OWEN

6.1. Мета проведення лабораторної роботи

На дану лабораторну роботу відповідно до робочої програми відводиться 4 ак. години аудиторного та 4 ак. години самостійної роботи студента (СРС).

Мета заняття:

- закріпити на практиці отримані теоретичні відомості про специфікації протоколу *OWEN* на послідовному інтерфейсі *RS-485*;
- ознайомитись з реалізацією протоколу *OWEN* на послідовному інтерфейсі *RS-485* у *ПЛК150* (головний та підлеглий пристрій) за допомогою конфігурування його ресурсів;
- навчитись конфігурувати панель оператора *СМИ-1* для мережної взаємодії з *ПЛК150*.

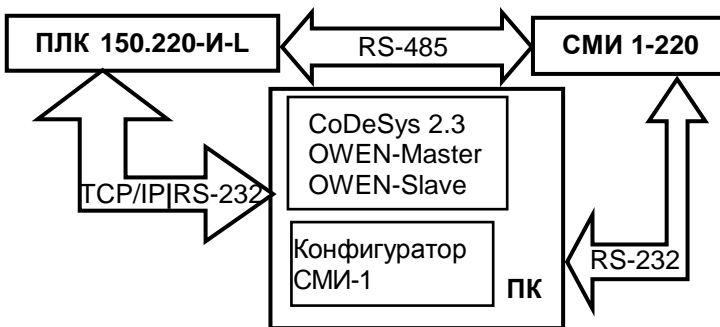
6.2. Апаратне забезпечення лабораторної роботи

Практичне заняття проводиться на робочих місцях з ПК, де на стендах встановлені *ПЛК150* та *СМИ-1*. Контролер та панель з'єднані інтерфейсом *RS-485*.

6.3. Програмне забезпечення лабораторної роботи

Для створення та завантаження проєкту до *ПЛК150* використовується середовище *CoDeSys V2.3*. Конфігурування панелі оператора *СМИ-1* реалізується за допомогою програми «Конфігуратор *СМИ-1*».

На рисунку представлена схема комунікаційних зв'язків та використаного апаратного та програмного забезпечення:



6.4 Послідовність проведення заняття

6.4.1 Обмін даними за протоколом *OWEN* у режимі *ПЛК150* – головний та *СМИ-1* – підлеглий з використанням інтерфейсу *RS-485*

6.4.1.1 Програмування та конфігурування контролера. Для обміну даними використайте програму, яка наведена у п. 2.4.1.1 на стор. 17 зі змісту лабораторної роботи 2. Конфігурація ресурсів з фіксованими модулями теж співпадає з конфігурацією, яка теж зображена на стор.17.

Відмінність полягає у програмному модулі OWEN (Master) [VAR], якій додається:

Індекс	Імя	Значення
1	Communication speed	9600
2	Parity	NO PARITY...
3	Data bits	8 bits
4	Stop length	One stop bit
5	Interface Type	RS485
6	Frame oriented	ASCII
7	Framing time ms	0
8	Visibility	No

Налаштування мережних змінних у модулі OWEN (Master) [VAR]:

- змінна типа *float* для запису до СМІ-1 значення змінного опору, який підключений до 4-го аналогового входу ПЛК150:

Індекс	Імя	Значення
1	Address length	8 bit
2	Address	16
3	Hash name	res
4	Index	0
5	Use a index?	No
6	Float type	Float
7	Precision	2
8	Polling time ms	100
9	Work Mode	Polling Time
10	Repeat counter	0
15	Visibility	No

- змінна типа *float* для запису в СМІ-1 значення температури від термопари, яка підключена до 3-го аналогового входу ПЛК150:

Індекс	Імя	Значення
1	Address length	8 bit
2	Address	16
3	Hash name	thk
4	Index	0
5	Use a index?	No
6	Float type	Float
7	Precision	2
8	Polling time ms	100
9	Work Mode	Polling Time
10	Repeat counter	0
15	Visibility	No

- змінна типа *unsigned* розміром в один байт для запису до СМІ-1

значення маски дискретних входів на ПЛК150:

```
---Owen (Master) [VAR]
  AT %QW9.0: WORD; (* Last err
  AT %QW9.1: WORD; (* Last has
  AT %QW9.2: WORD; (* Last add
  AT %QB9.3: BYTE; (* Start/St
  RS-485-1[SLOT]
  Float variable (Write) [VAR]
    res_1 AT %QD9.1.0: REAL;
  Float variable (Write) [VAR]
    thk_1 AT %QD9.2.0: REAL;
  Unsigned variable (Write) [VAR]
    8 bits [VAR]
      by_din AT %QB9.3.0.0:
```

Индекс	Имя	Значение
1	Address length	8 bit
2	Address	16
3	Hash name	byin
4	Index	0
5	Use a index?	No
6	Polling time ms	100
7	Work Mode	Polling Time
8	Repiat counter	0
15	Visibility	No

- змінна типа *unsigned* розміром один байт для отримання в ПЛК150 значення маски дискретних входів у СМІ-1 (параметр r.Cn):

```
--- AT %QW9.2: WORD; (* Last add
--- AT %QB9.3: BYTE; (* Start/St
--- RS-485-1[SLOT]
--- Float variable (Write) [VAR]
  res_1 AT %QD9.1.0: REAL;
--- Float variable (Write) [VAR]
  thk_1 AT %QD9.2.0: REAL;
--- Unsigned variable (Write) [VAR]
  8 bits [VAR]
    by_din AT %QB9.3.0.0:
--- Unsigned variable (Listen) [VAR]
  8 bits [VAR]
    by_dout AT %QB9.4.0.0:
```

Индекс	Имя	Значение
1	Address length	8 bit
2	Address	16
3	Hash name	r.cn
4	Index	0
5	Use a index?	No
6	Polling time ms	100
7	Work Mode	Polling Time
8	Repiat counter	0
15	Visibility	No

- змінна типа *unsigned* розміром два байти для отримання в ПЛК150 значення уставки (ціле, позитивне число) від СМІ-1:

```
--- AT %QW9.2: WORD; (* Last address
--- AT %QB9.3: BYTE; (* Start/Stop)
--- RS-485-1[SLOT]
--- Float variable (Write) [VAR]
  res_1 AT %QD9.1.0: REAL; (* *
--- Float variable (Write) [VAR]
  thk_1 AT %QD9.2.0: REAL; (* *
--- Unsigned variable (Write) [VAR]
  8 bits [VAR]
    by_din AT %QB9.3.0.0: BYTE
--- Unsigned variable (Listen) [VAR]
  8 bits [VAR]
    by_dout AT %QB9.4.0.0: BYTE
--- Unsigned variable (Listen) [VAR]
  2 byte [VAR]
    w_ust AT %QW9.5.0.0: WORD;
```

Индекс	Имя	Значение
1	Address length	8 bit
2	Address	16
3	Hash name	wust
4	Index	0
5	Use a index?	No
6	Polling time ms	100
7	Work Mode	Polling Time
8	Repiat counter	0
15	Visibility	No

- змінна типа *float* для отримання в *ПЛК150* значення уставки (УТС 4...20 мА) від *СМІ-1*:

The image shows a configuration window with variable declarations on the left and a table of system parameters on the right. A red dashed circle highlights the 'Float variable' declaration and the corresponding row in the table.

```

    Unsigned variable (Listen) [VAR]
      └─ 8 bits [VAR]
         └─ by_dout AT %QB9.4.0.0:
    Unsigned variable (Listen) [VAR]
      └─ 2 byte [VAR]
         └─ w_ust AT %QW9.5.0.0: WO:
    Float variable (Listen) [VAR]
      └─ aout_ust AT %AQ9.6.0: REAL;
  
```

Индекс	Имя	Значение
2	Address length	8 bit
3	Address	16
4	Hash name	aust
5	Index	0
6	Use a index?	No
7	Float type	Float
8	Precision	2
9	Polling time ms	100
10	Work Mode	Polling Time
11	Repiat counter	0
15	Visibility	No

6.4.1.2 Конфігурування панелі оператора *СМІ-1*. Мережні параметри програми-конфігуратора для зв'язку ПК та *СМІ-1* будуть подібними до значень, які подані у лабораторній роботі №2 (див. п.2.4.1.2, стор. 20-21). Мережні параметри комунікаційного модуля для зв'язку *СМІ-1* та *ПЛК150* теж співпадають з налаштуваннями, що подані на стор. 21.

Налаштування мережних змінних у панелі оператора *СМІ-1*:

- змінна типа *IntB* для передавання до *ПЛК150* значення уставки 0...1000 од. від *СМІ-1* на екрані Параметр редакування 1:

The image shows the 'Свойства' (Properties) window for 'Параметр редакування 1'. The 'Тип параметра' (Parameter type) is set to 'IntB'. The 'min Значения' (min value) is 0, and the 'max Значения' (max value) is 1000. The 'Имя параметра' (Parameter name) is 'WUST', and the 'Текущее значение' (Current value) is 55. The 'Порт' (Port) is empty, 'Режим' (Mode) is 'Slave', and 'Протокол' (Protocol) is 'ОБЕИ'.

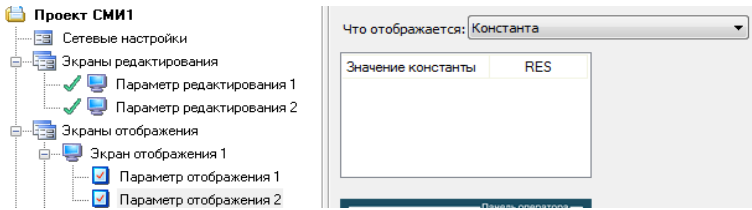
Тип параметра	IntB
min Значения	0
max Значения	1000
Имя параметра	WUST
Текущее значение	55

- змінна типа *float* для передавання на аналоговий вихід *ПЛК150* значення уставки (УТС 4...20 мА) від *СМІ-1* на екрані Параметр редакування 2:

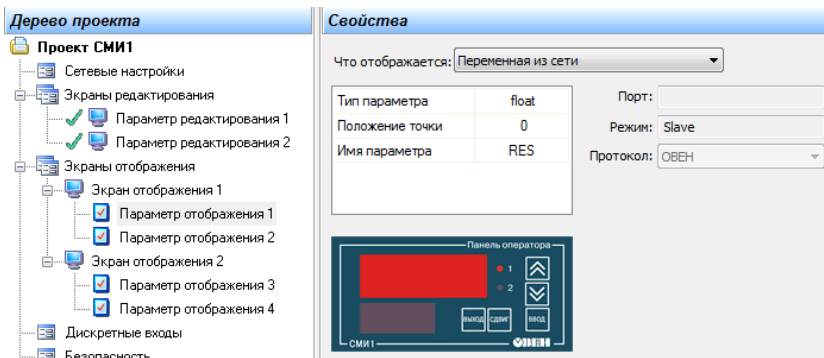
The image shows the 'Свойства' (Properties) window for 'Параметр редакування 2'. The 'Тип параметра' (Parameter type) is set to 'float'. The 'min Значения' (min value) is 4, and the 'max Значения' (max value) is 20. The 'Положение точки' (Point position) is 0. The 'Имя параметра' (Parameter name) is 'AUST', and the 'Текущее значение' (Current value) is 12. The 'Порт' (Port) is empty, 'Режим' (Mode) is 'Slave', and 'Протокол' (Protocol) is 'ОБЕИ'.

Тип параметра	float
min Значения	4
max Значения	20
Положение точки	0
Имя параметра	AUST
Текущее значение	12

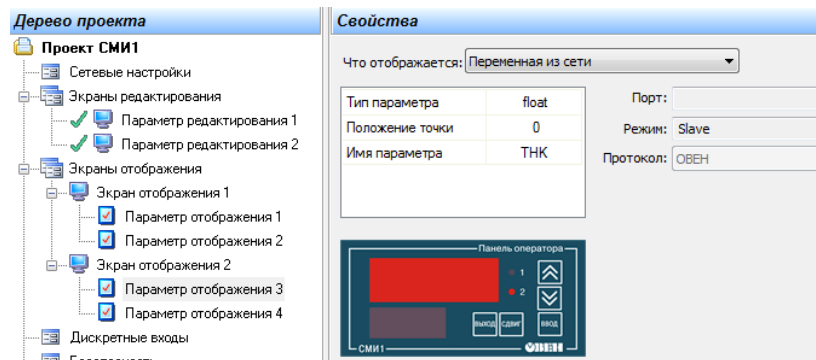
- відображення на першому екрані (Параметр отображения 2) константі RES



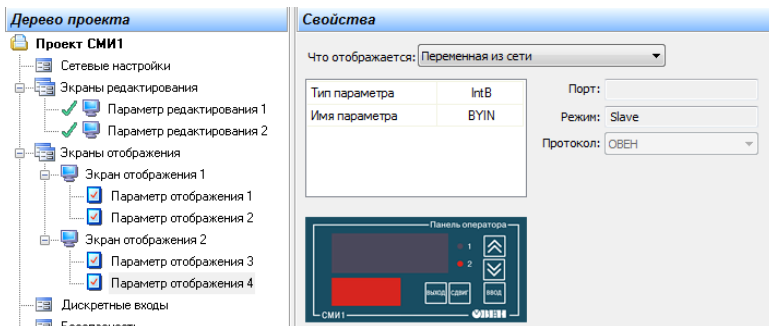
- відображення значення змінного опору RES на першому екрані Параметр отображения 1 змінної типа float:



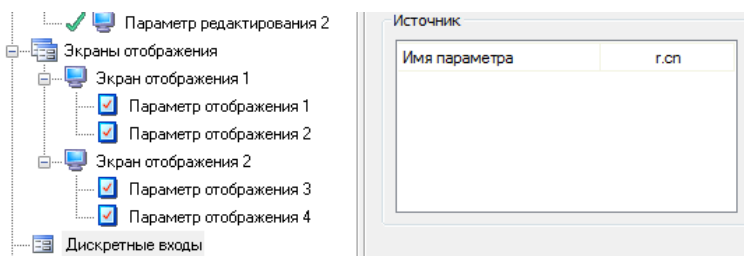
- відображення на другому екрані Параметр отображения 3 значення температуры змінної типа float, отриманої від ПЛК150:



- відображення на другому екрані Параметр отображения 4 значення маски дискретных входов ПЛК150 змінної типа unsigned:



- змінна типа *unsigned* для передачі до ПЛК150 значення маски дискретних входів на панелі СМІ-1:



6.4.2 Обмін даними за протоколом OWEN у режимі СМІ-1–головний та ПЛК150 – підлеглий з використанням інтерфейсу RS-485

6.4.2.1 Програмування та конфігурування контролера ПЛК150. Тіло програми та оголошення змінних:

```

0001 PROGRAM PLC_PRG
0002 VAR
0003     up1: UNPACK;
0004     dout5: BOOL;
0005     dout6: BOOL;
0006     dout7: BOOL;
0007     dout8: BOOL;
0008 END_VAR
0009

```

Конфігурація фіксованих модулів контролера:

```
└─ PLC 150 I
  └─ Discrete input 6 bit [FIX]
  └─ Discrete output 4 bit [FIX]
      ├─ dout1 AT %QX1.0: BOOL; (* *) [CHANNEL (Q)]
      ├─ dout2 AT %QX1.1: BOOL; (* *) [CHANNEL (Q)]
      ├─ dout3 AT %QX1.2: BOOL; (* *) [CHANNEL (Q)]
      └─ dout4 AT %QX1.3: BOOL; (* *) [CHANNEL (Q)]
  └─ Special output [FIX]
  └─ Unifed signal sensor [SLOT]
  └─ Unifed signal sensor [SLOT]
  └─ Unifed signal sensor [SLOT]
  └─ Unifed signal sensor [SLOT]
  └─ Analog output [FIX]
      └─ aout1 AT %QD7.0: REAL; (* Value *) [CHANNEL (Q)]
  └─ Analog output [FIX]
```

Конфігурування модулів, які додаються.

Склад модуля OWEN (slave) [VAR] з мережними змінними та параметри налаштування веденого пристрою:

```
└─ Unifed signal sensor [SLOT]
└─ Unifed signal sensor [SLOT]
└─ Analog output [FIX]
└─ Analog output [FIX]
└─ OWEN (slave) [VAR]
    └─ Owen [FIX]
        └─ Float variable (Listen) [VAR]
        └─ Unsigned variable (Listen) [VAR]
        └─ Unsigned variable (Listen) [VAR]
```

Индекс	Имя	Значение
1	Slave Name	plc150
2	Address Length	8 bit
3	Address	32
4	Visibility	No

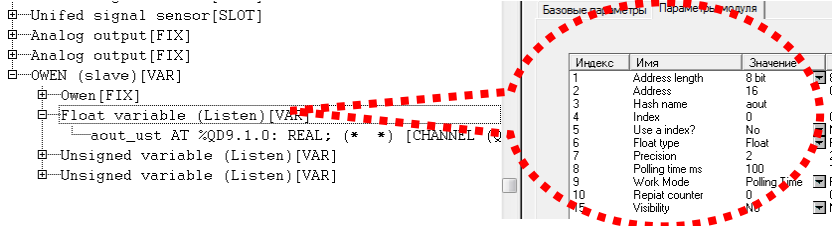
Параметри налаштування субмодуля RS-485-1 [VAR] в комунікаційному модулі Owen [FIX]:

```
└─ Unifed signal sensor [SLOT]
└─ Analog output [FIX]
└─ Analog output [FIX]
└─ OWEN (slave) [VAR]
    └─ Owen [FIX]
        └─ RS-485-1 [VAR]
            └─ Float variable (Listen) [VAR]
            └─ Unsigned variable (Listen) [VAR]
            └─ Unsigned variable (Listen) [VAR]
```

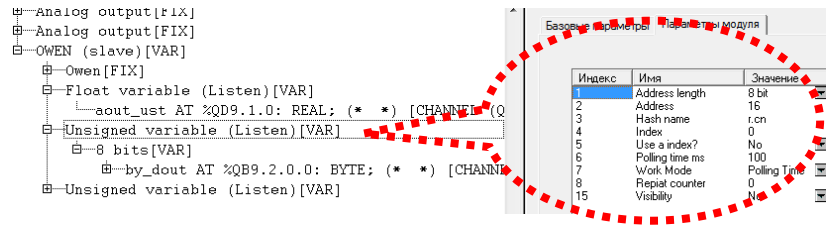
Индекс	Имя	Значение
1	Communication speed	9600
2	Parity	NO PARITY...
3	Data bits	8 bits
4	Stop length	One stop bit
5	Interface Type	RS485
6	Frame oriented	ASCII
7	Framing time ms	100
8	Vis bit	No

Параметри налаштування мережних змінних для обміну з СМІ-1

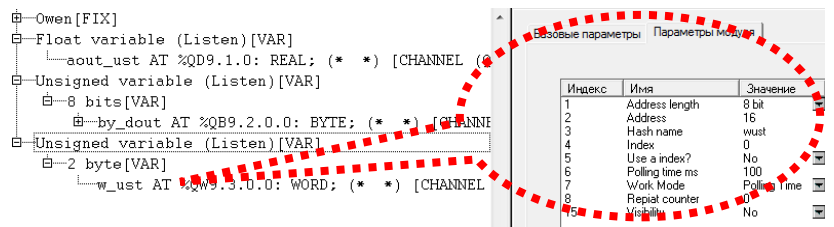
- змінна типу *float* для отримання в ПЛК150 значення УТС 4...20 мА від панелі оператора СМІ-1:



- змінна типа *unsigned* для передавання до ПЛК150 значення маски дискретних входів на операторській панелі СМІ-1:



- змінна типа *unsigned* для передачі в ПЛК150 значення уставки (ціле, позитивне число) з операторської панелі СМІ-1:



6.4.2.2 Конфігурування операторської панелі СМІ-1. Мережні параметри програми-конфігуратора для зв'язку ПК та СМІ-1 встановить такими ж як в п.6.4.1.2. Мережні параметри комунікаційного модуля для зв'язку ПЛК150 та СМІ-1 будуть такими ж, за винятком того, що панель буде працювати у режимі головного пристрою.

Налаштування мережних змінних в панелі СМІ-1:

- змінна типа *float* для передачі на аналоговий вихід ПЛК150 значення уставки (UTC 4...20 мА) від СМІ-1 на екрані Параметр редактування 2:

Дерево проекта

- Проект СМІ1
 - Сетевые настройки
 - Экраны редактирования
 - ✓ Параметр редактирования 1
 - ✓ Параметр редактирования 2
 - Экраны отображения
 - ✓ Параметр отображения 1
 - ✓ Параметр отображения 2
 - ✓ Экран отображения 2
 - ✓ Параметр отображения 3
 - ✓ Параметр отображения 4
 - Дискретные входы
 - Безопасность

Свойства

Тип параметра	float
min Значения	4
max Значения	20
Положение точки	0
Имя параметра	AOUT
Текущее значение	12

Порт: RS-485
Режим: Master
Протокол: OVBH

Получатель	
Адрес	32
Имя параметра	AOUT
Линейный индекс	Нет
Период опроса	00:01

- змінна типа *IntB* для передачі в *ПЛК150* значення уставки 0...1000 ед. від *СМІ-1* Параметр редактирования 2:

Дерево проекта

- Проект СМІ1
 - Сетевые настройки
 - Экраны редактирования
 - ✓ Параметр редактирования 1
 - ✓ Параметр редактирования 2
 - Экраны отображения
 - ✓ Параметр отображения 1
 - ✓ Параметр отображения 2
 - ✓ Экран отображения 2
 - ✓ Параметр отображения 3
 - ✓ Параметр отображения 4
 - Дискретные входы
 - Безопасность

Свойства

Тип параметра	IntB
min Значения	0
max Значения	1000
Имя параметра	WUST
Текущее значение	111

Порт: RS-485
Режим: Master
Протокол: OVBH

Получатель	
Адрес	32
Имя параметра	WUST
Линейный индекс	Нет
Период опроса	00:01

- відображення на першому екрані Параметр отображения 1 та Параметр отображения 2 уставок, тобто значень параметрів редагування WUST и AOUT відповідно:

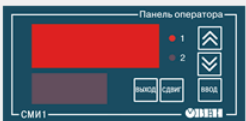
Дерево проекта

- Проект СМІ1
 - Сетевые настройки
 - Экраны редактирования
 - ✓ Параметр редактирования 1
 - ✓ Параметр редактирования 2
 - Экраны отображения
 - ✓ Экран отображения 1
 - ✓ Параметр отображения 1
 - ✓ Параметр отображения 2
 - ✓ Экран отображения 2
 - ✓ Параметр отображения 3
 - ✓ Параметр отображения 4
 - Дискретные входы
 - Безопасность

Свойства

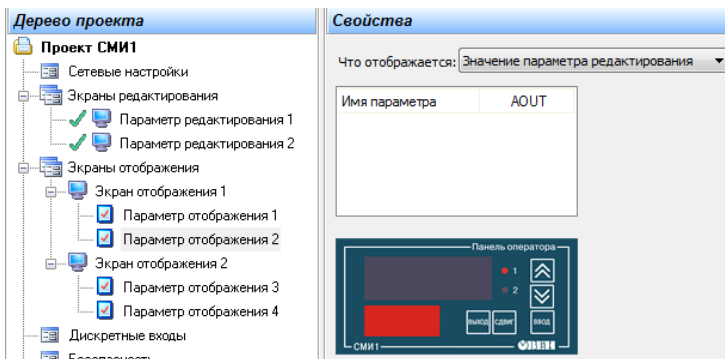
Что отображается: Значение параметра редактирования

Имя параметра	WUST
---------------	------

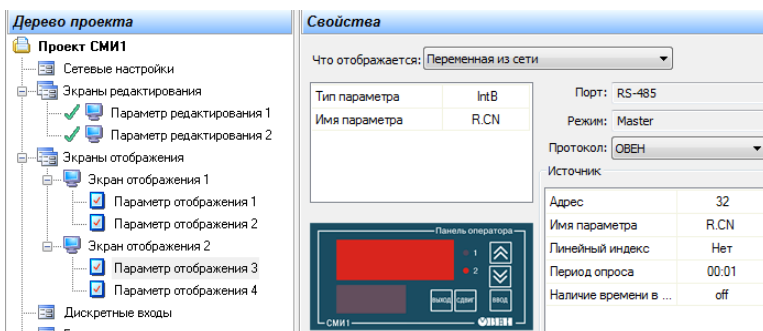


Панель оператора

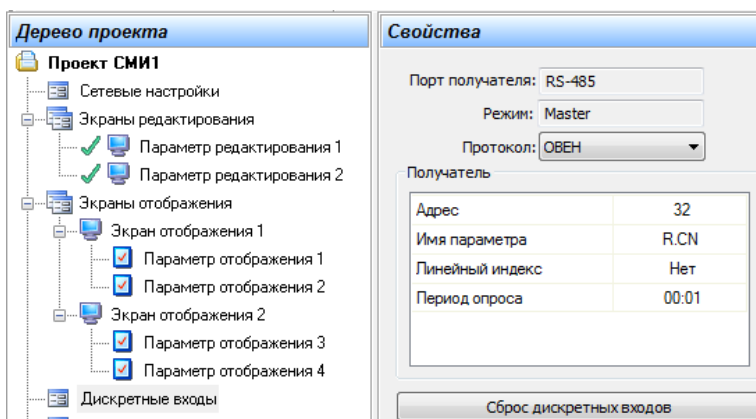
СМІ1



- відображення на другому екрані Параметр отображения 3 значення маски дискретних входів на СМІ-1 для передачі до ПЛК150:

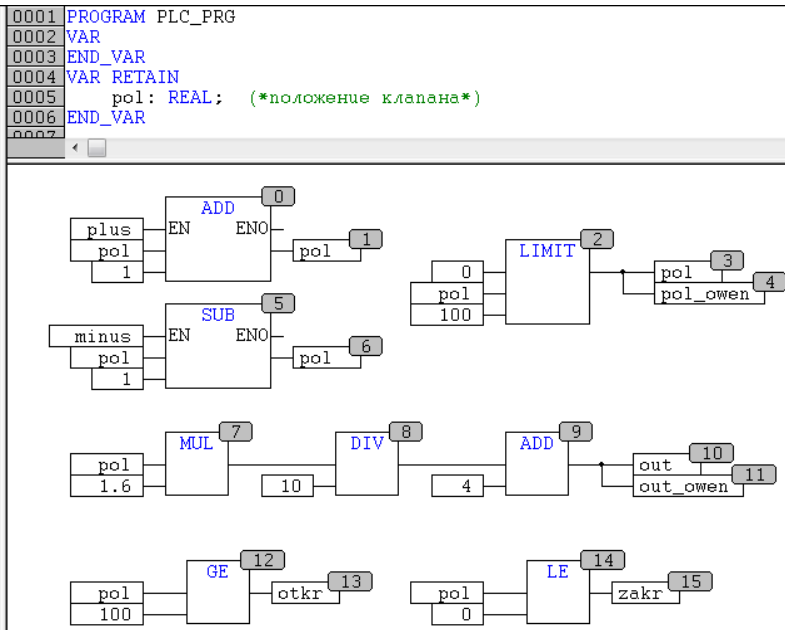


- змінна типа *unsigned* для передачі до ПЛК150 значення маски дискретних входів на панелі СМІ-1:



6.5 Проект РСУ для самостійного розроблення

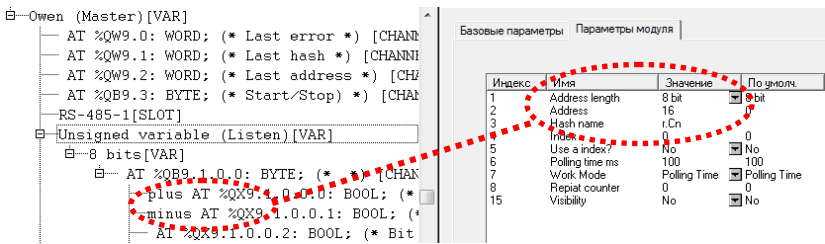
Як самостійне завдання пропонується розробити проект з програмою користувача для ручного управління клапаном. У програмі значення змінних `plus` та `minus`, які отримані від модуля `OWEN (Master)`, який, в свою чергу, зв'язаний з операторської панеллю `СМІ-1` за протоколом `OWEN` за допомогою інтерфейсу `RS-485`. Значення локальної змінної `pol` залежить від статусу змінних `plus` и `minus`, які керують операцією інкремента і декремента, та представляє значення положення клапана у відсотках. Тому значення змінної `pol` зверху та знизу обмежено діапазоном від 0 до 100 %. Значення струмового сигналу, тобто змінна `out`, обчислюється виходячи з діапазону його зміни (4...20 мА) і подається на аналоговий вихід ПЛК. Для передачі значень змінних `pol` і `out` в мережу в програмі передбачено їх дублювання – `pol_owen` та `out_owen` відповідно. Крім того, у програмі передбачена сигналізація про досягнення клапаном крайніх станів – це глобальні змінні `otkr` та `zacr`, які зв'язані з дискретними виходами контролера.



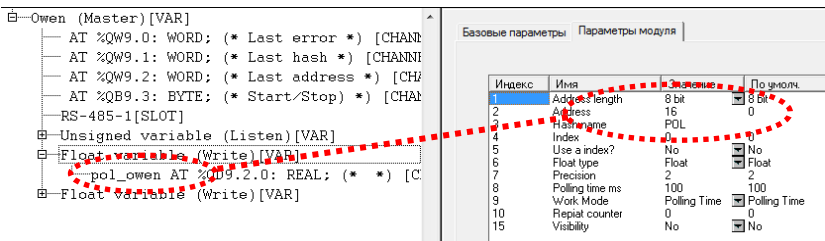
Отже, створіть у середовищі *CoDeSys V2.3* проєкт, перейдіть до вкладення «Ресурси» та оберіть утиліту «Конфігурація ПЛК». Додайте та проведіть конфігурування мережного програмного модуля Owen (Master). У модулі Owen (Master) [VAR] змініть слот Debug RS-232 [SLOT] на слот з інтерфейсом RS-485-1 з наступними параметрами інтерфейсу: «9600», «8-n-1» та «ASCII».

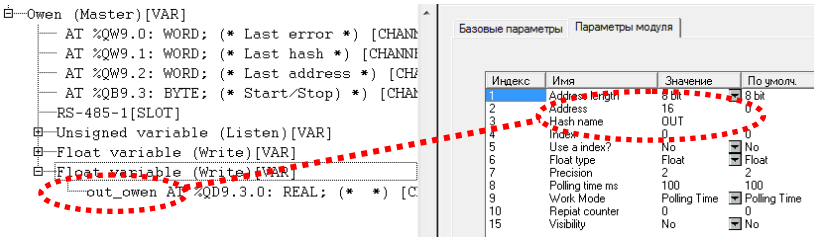
Додайте до модуля Owen (Master) мережні змінні: типу Unsigned variable (Listen) [VAR] з компонентом 8 bites [VAR] для читання статусу дискретних входів панелі оператора – plus та minus; типа Float variable (Write) [VAR] – для запису до панелі оператора значень положення клапана та значення струмового сигналу на аналоговий вихід pol_owen та out_owen відповідно.

Для читання статусу дискретних входів в *СМII-1* використовується змінна з *HASH*-іменем r.Cn, причому адреса підлеглого пристрою, тобто панелі оператора дорівнює «16»:



Для запису до панелі оператора значень положення клапана та струмового сигналу використані змінні з *HASH*-ім'ям: POL та OUT.





Збережіть проєкт на жорсткому диску, скопіюйте його та завантажте код проєкту до ПЛК. Запустіть проєкт в ПЛК на виконання.

Далі необхідно провести конфігурування панелі оператора *СМІ-1* за допомогою програми-конфігуратора. Після запуску програми, буде створено шаблон у вигляді дерева проєкту, в якому панель оператора має статус підлеглого пристрою. У вікні «Свойства» у вкладенні «Выбор протокола» поставте позначку в полі напроти рядка *ОВЕН*. Потім потрібно вказати порт для мережного обміну з параметрами, відповідними до налаштувань модуля інтерфейсу головного пристрою у ПЛК.

У гілках дерева конфігурації «Экраны отображения» зробіть наступні налаштування:

- налаштуйте поле «*Параметр отображения 1*», на якому буде відображатися значення положення клапана у відсотках 0...100%;
- налаштуйте поле «*Параметр отображения 2*» для відображення елемента «*Константа*» з ім'ям POL;
- налаштуйте поле «*Параметр отображения 3*», на якому буде відображатися значення вихідного струмового сигналу 4...20 мА;
- налаштуйте поле «*Параметр отображения 4*» для відображення елемента «*Константа*» з ім'ям OUT.

Гілку дерева конфігурації «*Дискретные входы*» залишіть без змін.

Збережіть та завантажте проєкт до панелі, попередньо встановивши з нею зв'язок. Перевірте працездатність макета РСУ.

6.6 Оформлення результатів проведення заняття

Складіть звіт про проведене заняття, де відобразіть схему з'єднань та умови обміну даними. Також вкажіть, які режими використані для обміну даними.

6.7 Контрольні запитання

- 1) На яких рівнях моделі зв'язку побудований протокол *OWEN*?
- 2) Який формат кадру у протоколі *OWEN*?
- 3) Як реалізована достовірність даних на каналному рівні протоколу *OWEN*?
- 4) Які принципи адресації закладені у протоколі *OWEN*?

Лабораторна робота 7

ОСНОВНІ ПРИНЦИПИ СТВОРЕННЯ РСУ НА ПОСЛІДОВНИХ ІНТЕРФЕЙСАХ ТА ЗА ПРОТОКОЛОМ *OWEN*

7.1. Мета проведення лабораторної роботи

На лабораторну роботу відповідно до робочої програми відводиться 4 ак. години аудиторного та 4 ак. години самостійної роботи студента (СРС).

Мета заняття:

- закріпити на практиці навички організації мережної взаємодії контролерів *ПЛК150* з периферійними пристроями по послідовному інтерфейсу *RS-485* із застосуванням протоколу *OWEN*;
- навчитися конфігурувати локальні регулятори серії *TPM* (*TPM101*, *TPM201*), модулі введення/виведення (*MBA*, *MB110-2A*, *МДВВ*) для роботи у складі РСУ.

7.2. Апаратне забезпечення лабораторної роботи

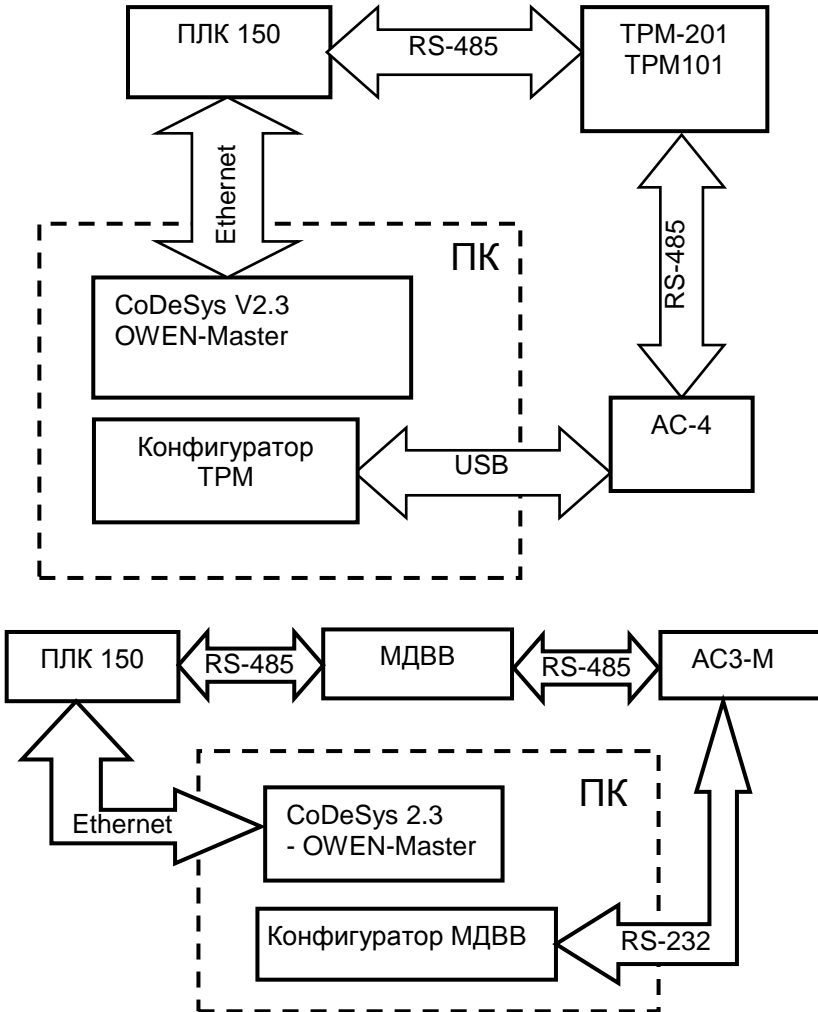
Лабораторна робота виконується на стендах, де встановлені ПЛК та периферійне устаткування (ПЗО): регулятори (*TPM101*, *TPM201*) та модулі введення/виведення (*МДВВ*, *MBA*, *MB110-2A*). Якщо на стенді відсутній *ПЛК150*, то для мережної взаємодії з ПЗО використовується ПК з програмою емулявання головного пристрою *OWEN Test Console*.

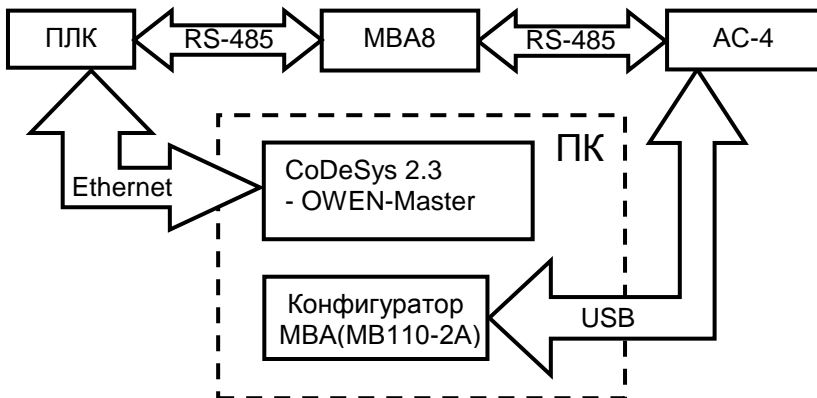
7.3 Програмне забезпечення лабораторної роботи

Для створення та завантаження проєкту до ПЛК використовується середовище *CoDeSys V2.3*. Це середовище також використовується для

візуалізації процесів регулювання та керування ПЗО. У всіх випадках мережної взаємодії ПЛК150 виконує функцію головного пристрою. Конфігурування регуляторів та інших приладів проводиться за допомогою програм-конфігураторів.

На рисунках представлені схеми комунікаційних зв'язків та використаного апаратного і програмного забезпечення для організації взаємодії ПЛК150 та ТРМ201, ТРМ101, МДВВ, МВА та МВ110-2А.





7.4 Послідовність проведення заняття

7.4.1 Розроблення PCУ на прикладі взаємодії ПЛК150 та TPM101 (TPM201)

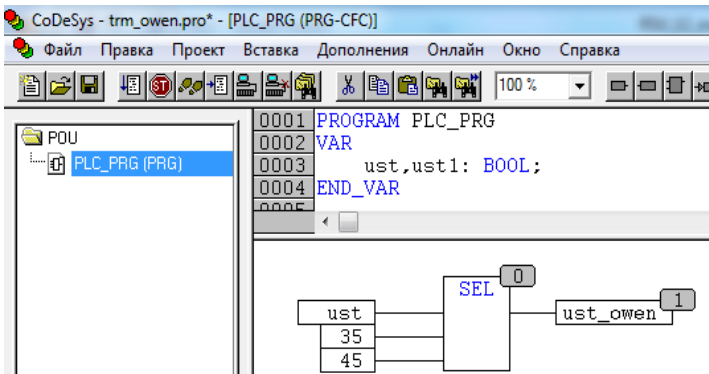
7.4.4.1 Конфігурування регуляторів серії *TPM* для роботи у складі PCУ. Основні відомості про призначення, порядок конфігурування регуляторів серії *TPM* викладені у п. 4.4.1.1 з ЛР №4. Відмінність полягає лише у виборі протоколу для мережної взаємодії. У лабораторній роботі застосовується протокол *OWEN*. Решта мережних параметрів залишаються незмінними. Для інтегрування приладу *TPM* до складу PCУ необхідно здійснити ряд кроків. По-перше, за допомогою програми-конфігуратора налаштувати інтерфейс та протокол обміну даними з ПК або ПЛК, по-друге – провести конфігурування функцій приладу.

Параметри у програмі-конфігураторі регулятора розподілені по групах: група *lnit* – основні параметри приладу, група *Adv* – параметри регулювання та *LBA*, група *Comm* – параметри обміну по мережі *RS-485*, а також групи з доступом по пароллю. Склад та кількість груп параметрів залежить від типу регулятора. Всі параметри в групах мають спеціальні *HASH*-імена. Ці *HASH*-імена параметрів приладу представлені в інструкції з експлуатації.

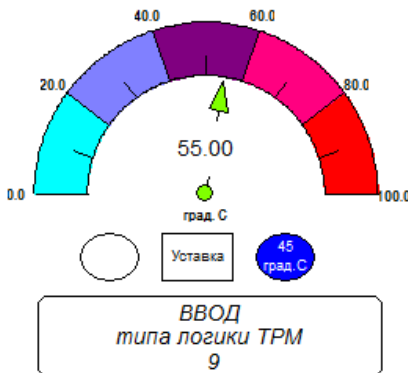
Робота з програмою полягає у введенні та редагуванні значень параметрів конфігурації приладу. Збережіть та завантажте конфігурацію до регулятора, попередньо встановивши з ним зв'язок.

7.4.1.2 Програмування ПЛК150 для керування регулятором ТРМ.

Як проєкт пропонується проєкт з програмою користувача для керування локальним регулятором *TRM101 (TRM201)*. За допомогою інтерфейсу *RS-485* за протоколом *OWEN* контролер зчитує значення на вимірювальному каналі регулятора, а також записує значення уставки та тип логіки регулятора. У програмі передбачений вибір двох значень уставки (35 °C або 45 °C) за допомогою змінної *ust* бульового типу, значення якої вводиться за допомогою кнопки у візуалізації. Змінна *ust* у прямому та інвертованому значенні, управляє елементами візуалізації – індикаторами. Також, за допомогою елемента візуалізації вводиться код типу логіки, який передається до регулятора. Нижче показаний фрагмент з програмою та візуалізація проєкту:



УПРАВЛЕНИЕ регулятором ТРМ101



Отже, створіть у середовищі *CoDeSys V2.3* проєкт, перейдіть до вкладення «Ресурси» та оберіть утиліту «Конфігурація ПЛК». Додайте та проведіть конфігурування програмного модуля *Owen-Master*. В модулі *Owen (Master)* [VAR] змініть інтерфейс *Debug RS-232 [SLOT]* на інтерфейс *RS-485-1* з наступними параметрами інтерфейсу: «9600», «8-n-1» та «ASCII».

Далі, послідовно додайте до модуля мережні змінні:

- типа *Float variable (Listen)* [VAR] з ім'ям *parametr* – для читання значення параметра на вимірювальному вході регулятора;
- типа *Float variable (Write)* [VAR] з ім'ям *ust_owen* – для запису значення уставки до регулятора;
- типа *Unsigned variable (Write)* [VAR] з компонентом *8 bites* [VAR] та ім'ям *alt_owen* – для запису кода типа логіки до регулятора.

Введіть *HASH*-ім'я та адресу мережних змінних. Для читання параметру на вимірювальному вході регулятора у всіх *TPM* використовується *HASH*-ім'я «PV». Для запису значення уставки та типа логіки регулятора використовується *HASH*-імена «SP» та «Alt», відповідно. Адреса підлеглого пристрою у мережі, тобто регулятора, дорівнює значенню «16». Нижче показані налаштування мережних змінних.

Left pane (tree view):

- Owen (Master) [VAR]
 - AT %QW9.0: WORD; (* Last error *)
 - AT %QW9.1: WORD; (* Last hash *)
 - AT %QW9.2: WORD; (* Last address
 - AT %QB9.3: BYTE; (* Start/Stop) *
 - RS-485-1[SLOT]
 - Float variable (Listen) [VAR]
 - parametr AT %QD9.1.0: REAL; (*
 - Float variable (Write) [VAR]
 - Unsigned variable (Write) [VAR]

Right pane (Parameters of the module):

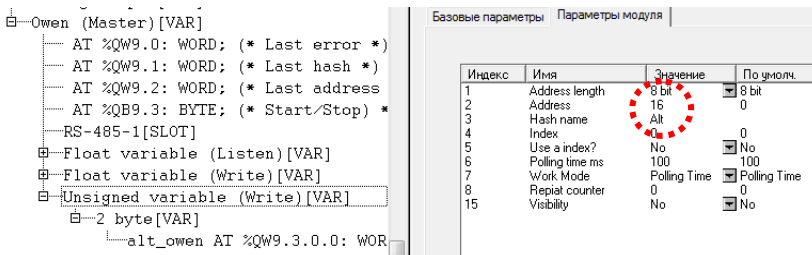
Индекс	Имя	Значение	По умолчанию
1	Address length	8 bit	8 bit
2	Address	16	0
3	Hash name	PV	0
4	Index		0
5	Use a index?	No	No
6	Float type	Float	Float
7	Precision	2	2
8	Polling time ms	100	100
9	Work Mode	Polling Time	Polling Time
10	Repeat counter	0	0
15	Visibility	No	No

Left pane (tree view):

- Owen (Master) [VAR]
 - AT %QW9.0: WORD; (* Last error *)
 - AT %QW9.1: WORD; (* Last hash *)
 - AT %QW9.2: WORD; (* Last address
 - AT %QB9.3: BYTE; (* Start/Stop) *
 - RS-485-1[SLOT]
 - Float variable (Listen) [VAR]
 - Float variable (Write) [VAR]
 - ust_owen AT %QD9.2.0: REAL; (*
 - Unsigned variable (Write) [VAR]

Right pane (Parameters of the module):

Индекс	Имя	Значение	По умолчанию
1	Address length	8 bit	8 bit
2	Address	16	0
3	Hash name	SP	0
4	Index		0
5	Use a index?	No	No
6	Float type	Float PIC	Float
7	Precision	2	2
8	Polling time ms	100	100
9	Work Mode	Polling Time	Polling Time
10	Repeat counter	0	0
15	Visibility	No	No



Збережіть проект на жорсткому диску, скопіюйте його та завантажте код до ПЛК. Запустіть проект в ПЛК на виконання. Перевірте працездатність макета РСУ. Проект буде працездатним з будь-яким регулятором, наприклад, з *TPM101* та *TPM201*.

За допомогою програми моніторинга *COM*-порта *COM Port Toolkit* прочитайте запити ПЛК. Проведіть аналіз запитів (див. ЛР №6):

- 1) 23 48 47 47 4A 50 48 47 4E 4B 49 4A 4B 47 47 4E 54 4C 55 0D;
- 2) 23 48 47 48 47 52 4F 54 56 52 53 49 51 0D;
- 3) 23 48 47 47 48 55 49 4D 55 47 4C 4D 4A 4A 4B 0D.

7.4.2 Створення РСУ на прикладі взаємодії ПЛК150 та ПЗО

7.4.2.1 Конфігурування ПЗО до роботи у складі РСУ. Основні відомості про призначення, порядок конфігурування модулів *MBA*, *MB110-2A* та *МДВВ* викладені раніше, у п. 4.4.2.1 ЛР №4. Відмінність полягає лише в протоколі для мережної взаємодії. Це протокол *OWEN*. Решта мережних параметрів залишаються незмінними.

Для інтегрування модулів до складу РСУ необхідно:

- по-перше, за допомогою програми-конфігуратора налаштувати інтерфейс та протокол обміну даними з ПЛК (ПК або ОП);
- по-друге, провести конфігурування входів та виходів у відповідності зі схемою стенда.

Після запуску програми у вікні налаштування мережних параметрів необхідно задати параметри мережі для зв'язку ПК та модуля, наприклад: «9600», «8-n-1», «16» та номер *COM*-порта.

ПРИМІТКА: Якщо робота виконується на ПК, до якого підключений стенд без потрібного модуля, то необхідно вибрати режим «Роботать OFFLINE».

7.4.2.2 Програмування ПЛК150 для зв'язку з модулем MBA (MB110-2A). Пропонується розробити проект з програмою користувача двохпозиційного регулятора температурою. Передбачається, що в ПЛК аналогових входів немає, або вони всі зайняті для інших цілей.

Нижче представлений варіант програми користувача:

```

0001 PROGRAM PLC_PRG
0002 VAR
0003     ust1: REAL:=60;
0004     ust2: REAL:=50;
0005 END_VAR

0001 IF temperatura>ust1 THEN
0002     ten:=FALSE;
0003     vent:=TRUE;
0004     lamp:=TRUE;
0005 END_IF
0006 IF temperatura<ust2 THEN
0007     ten:=TRUE;
0008     vent:=FALSE;
0009     lamp:=FALSE;
0010 END_IF

```

У програмі дискретні вихідні елементи ПЛК зв'язані зі змінними «lamp», «vent» та «ten», а їхній статус визначається порівнянням поточного значення змінної «temperatura», яка одержана по мережі RS-485 від модуля MBA (MB110-2A), з уставками «ust1» та «ust2».

Створіть у середовищі CoDeSys V2.3 проект з мережним модулем OWEN-Master та комунікаційним інтерфейсом «RS-485-1» з наступними параметрами: «9600», «8-n-1» та «ASCII».

Далі додайте до програмного модуля Owen (Master) [VAR] мережну змінну Float variable+time (Listen) [VAR] та налаштуйте її параметри. Вкажіть HASH-ім'я змінної для читання значення вимірювального каналу модуля – «READ». Для опитування усіх каналів використовується єдиний оперативний параметр, тому для їх ідентифікації використовується базова адреса приладу та номер каналу. наприклад, 4-ий канал MBA з базовою адресою «16» буде визначатися, як сума базової адреси та номера каналу, починаючи з нульового. Таким чином, адреса змінної буде дорівнювати значенню «19».

```

Analog output[FIX]
Owen (Master)[VAR]
  AT %QW9.0: WORD; (* Last error *) [CHANNE
  AT %QW9.1: WORD; (* Last hash *) [CHANNEL
  AT %QW9.2: WORD; (* Last address *) [CHAN
  AT %QB9.3: BYTE; (* Start/Stop *) [CHANNE
RS-485-1[SLOT]
  Float variable + time (Listen)[VAR]
    AT %QW9.1.0: WORD; (* Circular time *)
    temperatura AT %QD9.1.1: REAL; (* *) [

```

Индекс	Имя	Значение
1	Address length	8 bit
2	Address	19
3	Hash name	READ
4	Index	0
5	Use a index?	No
6	Float type	Float
7	Precision	2
8	Polling time ms	1000
9	Work Mode	Polling Time
10	Repeat counter	0
15	Visibility	No

Збережіть проєкт на жорсткому диску, скопіюйте його та завантажте код до ПЛК. Запустіть проєкт на виконання. Перевірте працездатність макета РСУ. Для реалізації людино-машинного інтерфейсу в РСУ створіть у середовищі *CoDeSys V2.3* візуалізацію проєкта. Приклад візуалізації використайте подібний до зображення на стор. 56.

Завантажте код проєкту до ПЛК. Запустіть проєкт на виконання та перевірте працездатність макета РСУ за допомогою людино-машинного інтерфейсу.

7.4.2.3 Програмування ПЛК150 для зв'язку з модулем МДВВ. Пропонується розробити проєкт з програмою користувача. Цей проєкт призначений для дистанційного керування котлом та насосом з аварійним блокуванням та відключенням. Він складається з двох *POU* – головної програми *PLC_PRG* та функціонального блоку *KOTEL*. У середині ФБ *KOTEL* реалізований алгоритм управління котлом. Значення дискретних датчиків і керуючі сигнали для вихідних елементів *МДВВ* передаються по мережі *RS-485*. Для запуску та зупинки котла в візуалізації проєкту використовують віртуальні кнопки. Відмінність полягає у протоколі обміну. У даному випадку це протокол *OWEN*.

The screenshot shows the CoDeSys IDE interface. The top window displays the ladder logic for the *PLC_PRG* program:

```

0001 PROGRAM PLC_PRG
0002 VAR
0003   kot_1: kotel;
0004   pusk_in: BOOL;
0005   stop_in: BOOL;
0006 END_VAR

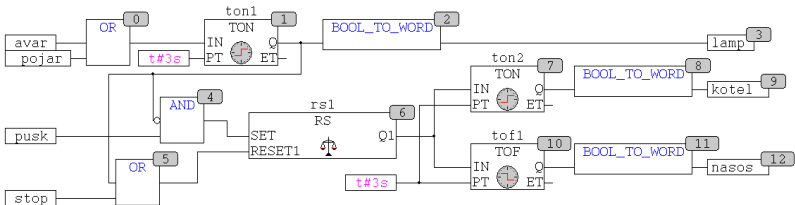
```

The bottom window shows the function block diagram for the *KOTEL* block. The block is named *kot_1* and has a parameter *kotel* set to 0. It has four input terminals: *maska_in.10*, *maska_in.11*, *pusk_in*, and *stop_in*. It has three output terminals: *out_1* (labeled *lamp*), *out_8* (labeled *pojar kotel*), and *out_7* (labeled *pusk nasos*).

```

0001 FUNCTION_BLOCK kotel
0002 VAR_INPUT
0003   avar:pojar,pusk,stop:BOOL;
0004 END_VAR
0005 VAR_OUTPUT
0006   lamp,kotel,nasos:WORD;
0007 END_VAR
0008 VAR
0009   ton1,ton2: TON;
0010   rs1: RS;
0011   tof1: TOF;
0012 END_VAR

```



Котельня складається з насоса, нагрівача, кнопок керування, датчика пожежі, аварійної кнопки та сигнальної лампи. Після натискання на кнопку «Старт» повинен включитися насос, а з деякою затримкою має запрацювати нагрівач. При натисканні на кнопку «Стоп» нагрівач повинен припинити роботу, а з деякою затримкою повинен вимкнутися насос. Якщо спрацює датчик пожежі або включили аварійну кнопку, то робота котла припиняється (подібно до натискання на кнопку «Стоп»), і не може бути відновлена, поки ці сигнали не зникнуть (пожежа загашена, кнопка вимкнена). Повторний запуск котла можливий лише після усунення причин та натискання на кнопку «Старт».

Отже, створіть в середовищі *CoDeSys V2.3* проєкт з мережним модулем *Owen (Master) [VAR]* та комунікаційним модулем *RS-485-1 [SLOT]*. Параметри інтерфейсу налаштуйте такими: «9600», «8-n-1» та «ASCII».

Додайте до конфігурації програмного мережного модуля *Owen (Master) [VAR]* мережні змінні для читання стану датчиків та для управління вихідними елементами:

- типа *Unsigned variable (Listen) [VAR]* з компонентом 2 byte [VAR] та ім'ям *maska_in* – для читання статусу входів МДБВ;
- типа *Unsigned variable (Write) [VAR]* з компонентом 2 byte [VAR] та ім'ям *out_1* – для вмикання 1-го виходу МДБВ, який сигналізує про наявність аварії;

– типа **Unsigned variable (Write) [VAR]** з компонентом 2 byte [VAR] та ім'ям `out_8` – для вмикання нагрівача, який підключений до 8-го виходу *МДВВ*;

– типа **Unsigned variable (Write) [VAR]** з компонентом 2 byte [VAR] та ім'ям `out_7` – для вмикання насосу, який підключений до 7-го виходу *МДВВ*.

Статус входів *МДВВ* визначається за допомогою оперативного параметру з *HASH*-ім'ям «`r.Cn`». Відзначимо, що за протоколом *OWEN* у модулі *МДВВ* групове управління виходами неможливо. Для виходів передбачено лише передавання значення шпаруватості ШІМ-сигналу на окремий канал за допомогою оперативного параметра з *HASH*-ім'ям «`r.OE`». Особливістю цього параметра полягає у тому, що він передає значення типу *WORD* для ШІМ-управління виходами. У проєкті керуючі сигнали будуть мати лише два значення: «0» або «1». Фактично це рівнозначно вмиканню або вимиканню виходу у процесі ШІМ-регулювання. Оперативні параметри не мають індексу, тому вони індексуються через адресу приладу. Таким чином, мережні змінні для управління виходами матимуть зміщені адреси відносно базової. Так, 1-ий вихід буде мати адресу, яка співпадає з базовою адресою приладу, тобто «16». Для управління насосом та нагрівачем адреси будуть дорівнювати значенням «22» і «23» (виходи №7 та №8).

Нижче показані налаштування мережних змінних для прослуховування статусу входів *МДВВ* та запису значення ШІМ у 1-ий канал.

The image shows a configuration window with two panes. The left pane displays a tree view of device settings:

- Analog output[**FIX**]
- Owen (Master)[**VAR**]
 - AT %QW9.0: WORD; (* Last er
 - AT %QW9.1: WORD; (* Last ha
 - AT %QW9.2: WORD; (* Last ac
 - AT %QB9.3: BYTE; (* Start/S
- RS-485-1[SLOT]
- Unsigned variable (Listen)[**V**]
 - 2 byte[**VAR**]
 - maska_in AT %QW9.1.0.0

The right pane is titled "Базовые параметры" and "Параметры модуля". It contains a table with the following data:

Индекс	Имя	Значение
1	Address length	8 bit
2	Address	16
3	Hash name	r.Cn
4	Index	0
5	Use a index?	No
6	Polling time ms	1000
7	Work Mode	Polling Time
8	Repeat counter	0
15	Visibility	No


```

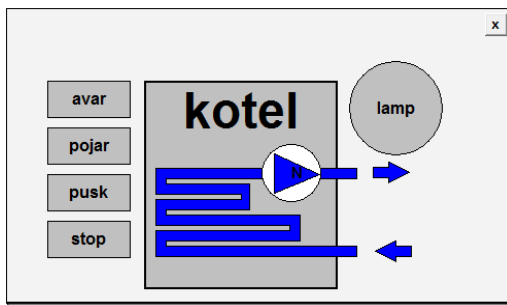
└─ Analog output[FIX]
└─ Owen (Master)[VAR]
    ├── AT %QW9.0: WORD; (* Last er
    ├── AT %QW9.1: WORD; (* Last ha
    ├── AT %QW9.2: WORD; (* Last ac
    ├── AT %QB9.3: BYTE; (* Start/S
    ├── RS-485-1[SLOT]
    ├── Unsigned variable (Listen)[V
        └─ 2 byte[VAR]
            └─ maska_in AT %QW9.1.0.0
    ├── Unsigned variable (Write)[VA
        └─ 2 byte[VAR]
            └─ out_1 AT %QW9.2.0.0: w
    ├── Unsigned variable (Write)[VA
        └─ 2 byte[VAR]
            └─ out_8 AT %QW9.3.0.0: w
    ├── Unsigned variable (Write)[VA
        └─ 2 byte[VAR]
            └─ out_7 AT %QW9.4.0.0: w

```

Базовые параметры | Параметры модуля

Индекс	Имя	Значение
1	Address length	8 bit
2	Address	16
3	Hash name	r,OE
4	Index	0
5	Use a index?	No
6	Polling time ms	1000
7	Work Mode	Polling Time
8	Repeat counter	0
15	Visibility	No

Збережіть проект на жорсткому диску, скомпілюйте його та завантажуйте код до ПЛК. Запустіть проект на виконання та перевірте працездатність макета РСУ. Для реалізації людино-машинного інтерфейсу в РСУ створіть візуалізацію проекту. Візуалізацію керування котлом зробіть подібною до зображеної:



Збережіть проект знову, скомпілюйте його та завантажуйте код до ПЛК. Тут є можливість вмикання та вимикання котла, є спостереження за станом датчиків аварії та пожежі, а також, за станом вихідних пристроїв – лампи аварії, котла та насоса. Всередині котла реалізовано інтерактивне відображення стану котла та насоса. Запустіть проект на виконання та перевірте працездатність макета РСУ за допомогою людино-машинного інтерфейсу.

7.5. Оформлення результатів проведення заняття

Складіть звіт про проведене заняття, де представте схему з'єднань, умови обміну даними. Також вкажіть, які налаштування та режими використані для обміну даними.

7.6. Контрольні запитання

- 1) На яких рівнях моделі зв'язку побудований протокол *OWEN*?
- 2) Який формат кадру у протоколі *OWEN*?
- 3) Як реалізована перевірка достовірності кадрів на каналному рівні у протоколі *OWEN*?
- 4) Які принципи адресації закладені у протоколі *OWEN*?
- 5) Як розділяють кадри у протоколі *OWEN*?

Лабораторна робота 8

ОСНОВНІ ПРИНЦИПИ ФОРМУВАННЯ ЗАПИТІВ ТА АНАЛІЗУ ВІДПОВІДЕЙ У ПРОТОКОЛІ *DCON*

8.1 Мета проведення лабораторної роботи

На лабораторну роботу відповідно до робочої програми відводиться 4 ак. години аудиторного та 4 ак. години самостійної роботи студента (СРС).

Мета заняття:

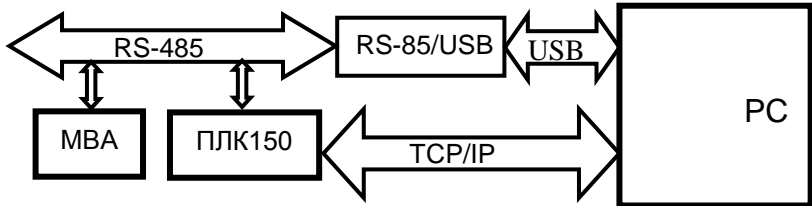
- закріпити на практиці отримані теоретичні відомості про стандарт на протокол *DCON* та на послідовні інтерфейси *RS-232* та *RS-485*;
- ознайомитися з реалізацією протоколу *DCON* на послідовному інтерфейсу *RS-485* в *ПЛК150* через конфігурування його ресурсів;
- навчитися інтерпретувати запити та відповіді у протоколі *DCON*.

8.2 Апаратне забезпечення лабораторної роботи

Лабораторна робота проводиться на робочих місцях з ПК та підключеними до них ПЛК з модулями введення/виведення, які встановлені на стендах. Для зв'язку з модулями введення/виведення

застосовуються ПІ *RS-232/RS-485* або *USB/RS-485*. Також можливе проведення заняття на інших ПК в режимі прослуховування запитів від ПЛК з допомогою програми *COM Port Toolkit*.

Для зв'язку ПК з ПЛК *ОВЕН150* та модулями *МВА*, *МВ110-2А* та *МДВВ* використовується ПІ *USB/RS-485*. Схема з'єднань ПЛК та модуля *МВА* показана нижче:



Схеми з'єднань з іншими модулями подібні до наведеної.

8.3 Програмне забезпечення лабораторної роботи

Для зв'язку ПК з модулями по послідовному інтерфейсу за допомогою ПІ *RS-485/USB* в ОС *Windows* за допомогою програмного драйвера емулюється віртуальний *COM*-порт, з яким без обмежень може працювати програма *COM Port Toolkit*, яка була використана у ЛР №1. Однак вона не може інтерпретувати запити протоколу *DCON* та відповідати на них. Для аналізу кадрів протоколу *DCON* від підлеглого пристрою можна використовувати програму *OWEN Test Console*, яка емулює роботу ведучого пристрою *DCON* та була використана у ЛР №5. Для цього досить налаштувати роботу модуля на роботу по протоколу *DCON* та опитувати модуль зазначеною програмою. Конфігурація модулів для роботи за протоколом *DCON* проводиться за допомогою спеціальних програм «*Конфігуратор МВА8*», «*Конфігуратор МДВВ*» та «*Конфігуратор МВ110*», які взаємодіють з модулями за допомогою ПІ.

8.4 Короткі теоретичні відомості

Протокол *DCON* хоча не відноситься до стандартних, але досить популярний завдяки застосуванню модулів введення/виведення, які розроблені компаніями *ICP DAS* (Тайвань) та *Advantech* (Тайвань). Цей

протокол реалізує символний обмін даними на послідовних інтерфейсах. Крім того, протокол *DCON* підтриманий у деяких приладах *ОВЕН*, наприклад, в ПЛК 100-ї серії та модулях введення/виведення: *MBA*, *МДВВ*, *МВУ* та *Мх110*. Популярність протоколу *DCON* зумовлена відсутністю необхідності у спеціалізованих мікросхемах для реалізації стека протоколу.

Протокол використовує лише фізичний та прикладної рівні моделі *OSI*. На фізичному рівні використовується пряме бінарне кодування *ASCII*-символів. Вимоги до середовища передачі визначаються стандартом на інтерфейси *RS-485* та *RS-232*. Також можлива робота з використанням звичайного або *GSM*-модема. У протоколі *DCON* застосовується метод доступу до каналу зв'язку «головний-підлеглий». У мережі може бути лише один головний пристрій і до 255 підлеглих. У протоколі *DCON* для організації опитування пристрою створюється рядок запиту на який підлеглий пристрій може повернути два варіанти відповіді: правильну відповідь, якщо команда розпізнана і є дані та неправильну відповідь, якщо команда не розпізнана або даних немає.

Для підвищення достовірності інформації на фізичному рівні використовується метод обчислення контрольної суми. Так як каналний рівень у протоколі не передбачено, то похибки передавання можуть бути виявлені лише на прикладному рівні, тобто безпосередньо у програмі користувача.

Кадр прикладного рівня протоколу *DCON* має такий вигляд:

Роздільник	Адреса	Команда	Блок даних	Контрольна сума
1 байт	1 байт	1...5 байт	1...256 байт	1 байт

Вся інформація, яка міститься у кадрі, передається в *ASCII*-кодах символів. Кожен кадр починається з роздільника. Як роздільники можуть бути використані символи: \$, #, %, @, *; у відповідях підлеглому пристрою використовують символи ~, !, ?, >. За деякими командами можуть слідувати дані, але іноді вони можуть бути відсутніми. Контрольна сума складається з двох символів теж може

бути відсутною. Кожен кадр повинен закінчуватися символом повернення каретки <CR> (*ASCII*-код $0D_{hex}$), який не входить до контрольної суми. Рядки формату відповіді можуть не задаватися, якщо пристрій не відповідає на запити. Рядок формату відповіді містить певні символи та спецкоманди.

Символ – будь-який символ, крім службових, до яких відносяться символи \$, [та]. За необхідністю виведення службовий символ як звичайний, він вводиться в рядок два рази поспіль.

Спецкоманда має наступний формат: [{модифікатор} дія].

Модифікатор – це рядок символів, які обробляються дією. Він є десятковим цілим числом. Може бути у всіх дій, крім обчислення контрольних сум. Наявність модифікатора необов'язкова, значення за замовчуванням дорівнює одиниці.

Дія – відображається в рядку спецкоманди одним з наступних символів – D, H, F, S, *, +, %. Регістр значення не має. Символи відповідають наступним видам дій:

D – представляє передану змінну у *ASCII*-коді у десятковому форматі (без знака) або перетворює *ASCII*-рядок з десяткового формату (без знака) в прийняту змінну. Кількість символів задається модифікатором;

H – представляє передану змінну у *ASCII*-символи у *HEX*-форматі або перетворює *ASCII*-рядок з *HEX*-формату в прийняту змінну. Кількість символів задається модифікатором;

F – представляє передану змінну у *ASCII*-символи у десятковому форматі зі знаком, роздільником цілої та дробової частини числа (точкою). Рядок має фіксоване число символів, яке задане модифікатором. Для прийнятих змінних генерує зворотне перетворення з *ASCII*-рядка у число;

S – здійснює пряме копіювання з переданої строкової змінної у рядок запиту числа символів, заданого модифікатором або зворотне копіювання з рядка відповіді у прийняту змінну строкового типу;

* – задає у рядку відповіді набір символів, які треба пропустити. Кількість символів може бути задана модифікатором;

+ – додає до рядку запиту контрольну суму або отримує її у рядку відповіді. Контрольна сума обчислюється шляхом додавання з переповненням по модулю «256». Дана дія не має модифікатора;

% – додає до рядка запиту контрольну суму або отримує її у рядку відповіді. Контрольна сума обчислюється за допомогою використання 8-ми бітного циклічного поліному (CRC). Дана дія не має модифікатора.

Контрольна сума (CHK) складається з двох символів HEX-формату (до суми не входить код символу <CR>) та дозволяє виявити похибки у командах, які були надіслані головним пристроєм, а також у відповідях підлеглого. Контрольна сума передається безпосередньо перед символом <CR> та повинна дорівнювати сумі кодових значень всіх ASCII-символів команди. Якщо сума більше FF_{hex}, то контрольна сума визначається лише молодшим байтом.

Наприклад, якщо потрібно переслати підлеглому пристрою команду \$052<CR>, то контрольна сума ASCII-кодів символів команди (символ <CR> не враховується) дорівнює такому значенню:

$$“\$”+“0”+“5”+“2”=24_{\text{h}}+30_{\text{h}}+35_{\text{h}}+32_{\text{h}}=1\text{BB}_{\text{h}}.$$

Таким чином, перед символом <CR> до команди треба додати код BB_h. В результаті команда \$052 буде мати такий вигляд: \$052BB<CR>.

Якщо відповідь модуля на цю команду без контрольної суми буде, наприклад, !05200600<CR>, то сума ASCII-кодів символів цієї команди дорівнює такому значенню:

$$\begin{aligned} “!”+“0”+“5”+“2”+“0”+“0”+“6”+“0”+“0”= \\ =21_{\text{h}}+30_{\text{h}}+35_{\text{h}}+32_{\text{h}}+30_{\text{h}}+30_{\text{h}}+36_{\text{h}}+30_{\text{h}}+30_{\text{h}}=1\text{AE}_{\text{h}}. \end{aligned}$$

У цьому випадку відповідь модуля з контрольною сумою буде наступним: !05200600C0AE<CR>, де передостанній байт «C0» вказує, що встановлений режим обміну з контрольною сумою.

При написанні програми прикладного рівня використовується набір стандартних команд, приклади яких наведені у табл. 8.1.

Таблиця 8.1 – Приклади команд протоколу *DCON*

Команда	Відповідь	Опис
%AANNTTCCFF	!AA	Встановлює адресу, діапазон вхідних напруг, швидкість обміну, формат даних, контрольну суму
#AA	>(Data)	Повертає всі вхідні значення аналогових каналів для заданого модуля
#AAN	>(Data)	Повертає дійсне значення аналогового каналу номер N для заданого модуля
\$AA0	!AA	Виконує калібрування аналогового модуля для компенсації похибки коефіцієнта передачі
\$AA1	!AA	Виконує калібрування аналогового модуля для компенсації похибки зміщення нуля
\$AA2	!AATTCCFF	Повертає параметри конфігурації модуля за вказаною адресою

Команди протоколу *DCON* поділяються на чотири типи:

- команди для модулів аналогового вводу;
- команди для модулів аналогового виводу;
- команди для модулів дискретного введення та виведення;
- команди для модулів лічильників та таймерів.

Розглянемо приклади формування запитів та аналізу відповідей за протоколом *DCON*.

1) Розглянемо приклад використання команди #AA для отримання даних від модуля аналогового введення (наприклад, 8-канального).

Синтаксис команди запиту буде таким:

AA [CHK] <CR> ,

де AA – адреса модуля (від 00_h до FF_h), [CHK] – контрольна сума, <CR> – повернення каретки.

Відповідь модуля на команду буде мати такий формат:

> (Data) [CHK] <CR> ,

якщо команда прийнята, правильно розшифрована та виконана. Тут символ > – це символ-роздільник, (Data) – виміряні дані, [CHK] – контрольна сума, <CR> – символ повернення каретки. Якщо мали місце синтаксичні помилки або є похибка зв'язку, то відповіді від модуля не буде.

Наприклад, для опитування модуля з адресою «26₁₀» розглянута команда отримує такий вигляд: # 1A [CHK] <CR>.

Відповідь модуля на команду може бути такою:

```
>+1.2345+0.3456+0.0001+2.5000+1.2345+  
+0.3456+0.0001+2.5000[CHK]<CR>.
```

У відповіді представлені вісім значень напруг на 8-ми входах модуля введення аналогових сигналів (0...10 В).

2) Приклад встановлення значення на вихідному каналі модуля аналогового виведення *IPC-7021* з періодичністю в одну секунду.

Синтаксис команди запиту буде таким:

```
# AA (дані) [CHK] [CR],
```

де # – роздільник; AA – адреса приладу; (дані) – значення для встановлення виходу (5 чисел + знак + точка), разом сім символів, [CHK] – контрольна сума, [CR] – повернення каретки.

Якщо команда прийнята та виконана, то відповідь модуля на команду буде мати такий формат:

```
! [CHK] [CR],
```

де ! – роздільник у разі успішного виконання команди; [CHK] – контрольна сума, [CR] – символ повернення каретки.

Нижче наведено приклад конфігурації ресурсів ПЛК у середовищі *CoDeSys V2.3* для програмного мережного модуля DCON (Master) [VAR] з підключеним до нього модулем Universal DCON device [VAR]. Тут проводиться запис значення на вихід модуля *IPC-7021* з адресою 24₁₀ (18_{hex}). Дані, які посилаються до модуля *IPC 7021*, задаються у вихідній змінній output_1 типу Float output (VAR). Вікно конфігурації модуля, яке налаштоване для періодичного записування вихідних значень до модуля *IPC-7021*, показано нижче:


```

├── Analog output[FIX]
├── Analog output[FIX]
├── DCON (Master)[VAR]
│   ├── AT %QW9.0: WORD; (* Last er
│   ├── RS-485-1[SLOT]
│   └── Universal DCON device[VAR]
│       ├── AT %QW9.1.0: WORD; (* St
│       └── Float output[VAR]
│           └── output_1 AT %ID9.1.0.0

```

Индекс	Имя	Значение	По умолч.
1	Request format	#18[7][+]	
2	Good response format	?	!
3	Bad response format	?	?
4	Max response timeout	150	150
5	Work mode	Both	<input checked="" type="checkbox"/> By poll time
6	Polling time ms	1000	100
7	Visibility	No	<input checked="" type="checkbox"/> No

Формат рядка запиту буде наступним:

18 [7f] [+],

де # – символ роздільника команди; 18 – адреса приладу у *HEX*-форматі (*важливо, що для букв використовується верхній регістр*); [7f] – спецкоманда, яка вказує на те, що сім символів запити повинні бути сформовані у вигляді числа у форматі «[знак] число число число число число», причому дані повинні бути взяті з вихідної змінної, яка повинна мати формат *float*; [+] – спецкоманда підрахунку та додавання в кінець запити контрольної суми «за модулем 256».

Увага! Символ повернення каретки у кадр додається автоматично.

Формат позитивної відповіді буде таким: !. Формат негативної відповіді – ?. У цих випадках дані символи не містять ніякої іншої інформації.

8.5 Послідовність проведення лабораторної роботи

8.5.1 Конфігурування модулів введення/виведення ОВЕН для зв'язку з ПЛК (ПК) за протоколом DCON

Модулі вводу/виводу *ОВЕН*, в яких підтримано протокол *DCON*, перед підключенням до РСУ повинні бути переведені на відповідний протокол роботи у мережі *RS-485*. Модулі працюють в мережі лише при наявності в ній «головного пристрою» (ПК або ПЛК). Конфігурування модуля здійснюється за протоколом *OWEN* за допомогою програм-кофігураторів. Для інтегрування модулів до складу РСУ необхідно:

- по-перше, за допомогою програми-конфігурувати налаштувати інтерфейс та протокол обміну даними з ПК або ПЛК;

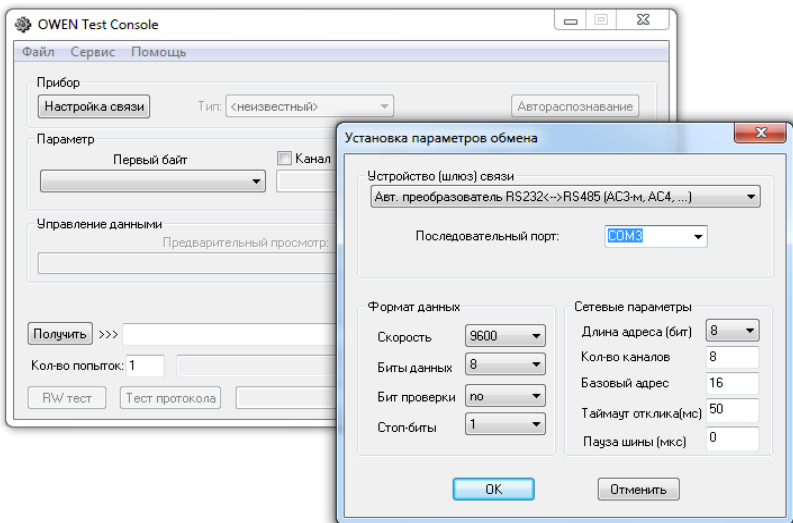
– по-друге, провести конфігурування входів та виходів відповідно до типу датчиків, які підключаються, та виконавчих механізмів.

Після запуску програми-конфігуратора у вікні налаштування мережних параметрів програми необхідно задати параметри для зв'язку з модулем по інтерфейсу *RS-485*, а саме: швидкість – «9600», кадр – «8-n-1», адреса – «16» та номер *COM*-порту.

Для зв'язку приладу з ПЛК (ПК) у складі РСУ необхідно вибрати протокол обміну *DCON*, а також налаштувати вимірювальні входи. Після закінчення конфігурування модуля збережіть та завантажте конфігурацію до модуля, попередньо встановивши з ним зв'язок. Перезавантажте модуль для переходу на роботу за протоколом *DCON*.

8.5.2 Тестування роботи модулів введення/виведення за протоколом *DCON*

Для тестування роботи модулів за протоколом *DCON* з боку ПК застосовується програма *OWEN Test Console*. У цьому випадку, для зв'язку використовується ПІ. Запустіть програму *OWEN Test Console*, яка виконує роль головного пристрою у мережі. Для читання значень у підлеглому пристрої виберіть протокол, тип ПІ, налаштуйте параметри інтерфейсу та вкажіть адресу модуля. На рисунку показані параметри обміну для тестування модулів за протоколом *DCON*:



Після налаштування параметрів обміну виберіть тип роздільника – перший байт (виберіть символ-роздільник «#») та канал, який опитується (наприклад, 4-ий канал модуля *MBA*) та ЛКМ натисніть на кнопку «Получить». Увага, нумерація каналів у модулі починається з нульового. У поле відповіді спостерігайте повідомлення від модуля.

8.5.3 Програмування ПЛК150 для роботи за протоколом DCON

Функціонування *ПЛК150* за протоколом *DCON* у середовищі *CoDeSys V2.3* реалізовано лише у режимі головного пристрою та реалізується додаванням до конфігурації ПЛК програмного модуля *DCON (Master)*, який за обраним комунікаційним інтерфейсом обмінюється даними з іншими пристроями. При цьому, у проєкті, який створений у середовищі *CoDeSys V2.3* для ПЛК, програма користувача може бути порожньою. До складу модуля доданий лише послідовний інтерфейс, які користувач налаштовує самостійно. Крім того, користувач додає потрібні змінні для обміну даними. У процесі роботи ПЛК відправляє запити підлеглим пристроям по інтерфейсу *RS-485* на читання параметрів каналів та встановлення значень на виходи.

Під час роботи *ПЛК150* за протоколом *DCON* є три варіанта роботи модуля: без розрахунку контрольних сум, з розрахунком контрольних сум шляхом складання значень всіх символів (за модулем «256») та з розрахунком контрольних сум 8-ми бітових символів з циклічним поліномом (*CRC*). Варіант роботи користувач обирає відповідно до того, який варіант розрахунку контрольної суми використовується у модулі, що опитується.

Використовуються такі алгоритми перетворення даних у модулі:

при формуванні запиту – усі символи поза спецкомандою копіюються у рядок запиту без зміни, спецкоманди замінюються на значення переданих (вихідних) змінних. Значення змінних кодуються у форматі, яка задана дією, число символів відповідає модифікатору;

при розшифруванні відповіді – усі символи поза спецкомандою порівнюються з відповідними позиціями відповіді та, при знаходженні відмінності, виробляється повідомлення про похибку. Дані у позиціях відповіді, які відповідають спецкоманді, перетворюються та

зберігаються у відповідних прийнятих (вхідних) змінних.

Якщо запит жорстко фіксований, тобто у рядку не містяться змінювані дані, то рядок формується без будь-яких команд, і в такому вигляді відсилається, при цьому може бути додана контрольна сума.

Аналогічно обробляється відповідь. Якщо приходять рядок, що не містить будь-яких даних (у кінці може бути контрольна сума), це означає, що модуль працює, реагує, тощо. Тобто сам факт отримання відповіді від модуля вже є інформацією.

Після додавання модуля **DCON (Master) [VAR]** та підключення до нього універсального пристрою **Universal DCON device [VAR]** у конфігурації ресурсів ПЛК з'являється незаповнений шаблон, параметри та змінні якого не прив'язані до конкретного зовнішнього пристрою. Після заповнення полів параметрів значеннями, підключення та визначення вхідних та вихідних змінних буде сформовано конкретний віртуальний пристрій для обміну даними. У вхідних змінних (**input variables**) значення записуються, коли приходять відповідь від опитуваного приладу і в них значення фіксуються; вихідні змінні (**output variables**) використовуються при формуванні запиту.

У модуль можна додати змінні різного розміру (8, 16 або 32 біта та 16 байтовий рядок) і типу (ціле без знака, з плаваючою точкою або рядок). Тип та порядок розташування вхідних та вихідних змінних у модулі повинні відповідати рядкам команд у полях запиту та відповіді. Створіть у середовищі *CoDeSys V2.3* проєкт, визначте необхідний таргет-файл та мову реалізації *POU PLC_PRG*. Цільовою платформою виберіть ПЛК *ОВЕН150-I.L*, а мову програмування виберіть *CFC*. Контролер буде лише опитувати модуль, тому вікно *POU PLC_PRG* залиште порожнім. Далі перейдіть до вкладення «Ресурси» та оберіть утиліту «Конфігурація ПЛК». Додайте до конфігурації ресурсів ПЛК програмний модуль **DCON (Master) [VAR]**, а у ньому змініть слот з інтерфейсом *RS-232* на *RS-485*. Далі, налаштуйте параметри інтерфейсу обміну: швидкість – «9600», формат кадру – «8-n-1» та режим обміну – «ASCII».

8.5.4 Моніторинг на ПК запитів головного пристрою та аналіз відповідей підлеглих пристроїв

Моніторинг запитів контролера та відповідей модулів можна здійснювати за допомогою програм сканування COM-портів, яка зазначені раніше.

Для моніторингу запитів на стендах, де немає пристроїв, які підтримують протокол *DCON*, запустіть на ПК програму сканування COM-порта. Оберіть потрібний COM-порт (віртуальний порт, який створений драйвером для ПІ *USB/RS-485*) та налаштуйте його параметри у відповідності з налаштуванням комунікаційного модуля *RS-485* у проєкті для ПЛК, а саме: швидкість – «9600», формат кадру – «8-n-1» та режим обміну – «ASCII».

Далі спостерігайте за прийнятими кадрами на екрані програми для моніторингу COM-порта. Прийняті запити розшифруйте та зіставте з налаштуванням запитів у проєкті для ПЛК. Зверніть увагу на значення контрольної суми та самостійно перевірте її правильність.

Наприклад, був прийнятий такий запит:

HEX: 40 31 30 34 30 30 35 0D ASCII: @ 10 40 05 .

Кадр містить два службових символа-роздільника кадрів. Це початок кадру – «40» («@») та кінець кадру – «0D» (<CR>, у ASCII-посилці позначений точкою). У полі даних вказана адреса приладу – байти «31» і «30», що в ASCII-кодi відповідає значенню «10₁₆». Далі вказано значення маски – байти «34» и «30», що відповідає значенню «40₁₆». У числі «40₁₆» виставлений лише біт в п'ятому розряді, що відповідає вмиканню дискретного виходу №6 в модулі *МДВВ*. Байти «30» та «35» є контрольною сумою, що відповідає значенню «05₁₆». Для перевірки відповідності значення суми прийнятого кадру підсумуємо прийняті байти, крім символу <CR>.

Отже: $40_{\text{hex}}+31_{\text{hex}}+30_{\text{hex}}+34_{\text{hex}}+30_{\text{hex}}=105_{\text{hex}}$. Так як контрольна сума становить два символи (один байт), то залишаємо молодший байт – 05_{hex} , що відповідає у запиті двом байтам «30_{hex}» і «35_{hex}». Остаточо можна зробити висновок: ПЛК формує запит на включення виходу №6 у модулі *МДВВ* з адресою №16.

8.5.5 Приклади запитів у протоколі DCON

1) Розглянемо приклад опитування четвертого каналу модуля *MBA* з адресою 16₁₀ (10_{hex}).

Формат запиту в параметрах універсального модуля буде таким:
10 3 [+],

де # – роздільник; 10 – адреса модуля у *HEX*-форматі; 3 – номер каналу (в даному випадку це четвертий канал модуля *MBA*); [+] – автоматичний розрахунок *CHK*.

Формат позитивної відповіді модуля буде наступним:

> [7f] [+],

де > – роздільник, 7 – модифікатор; f – представляє дію, тобто тип переданої змінної в *ASCII*-символах в десятковому форматі зі знаком, роздільником цілої та дробової частини числа. Рядок має фіксоване число символів, заданий модифікатором; [+] – автоматичний розрахунок *CHK*.

Вікно налаштування універсального модуля *DCON* для опитування 4-го аналогового каналу модуля *MBA* представлено нижче. Для зберігання значення на 4-ом вході *MBA* до віртуального універсального пристрою додана вихідна змінна типу *float*.

```
├─ Unified signal sensor[SLOT]
├─ Analog output[FIX]
├─ Analog output[FIX]
├─ DCON (Master)[VAR]
│   └─ AT %QW9.0: WORD; (* Last error
│       └─ RS-485-1[SLOT]
│           └─ Universal DCON device[VAR]
│               └─ AT %QW9.1.0: WORD; (* Status
│                   └─ Float output[VAR]
│                       └─ par_1 AT %ID9.1.0.0: REAL;
```

Индекс	Имя	Значение	По умолчанию
1	Request format	#103[+]	
2	Good response format	>[7f][+]	!
3	Bad response format	?	?
4	Max response timeout	150	150
5	Work mode	By poll time	By poll time
6	Polling time ms	100	100
7	Visibility	No	<input checked="" type="checkbox"/> No

2) Розглянемо приклад опитування усіх каналів модуля *MBA* з адресою 16₁₀.

Формат запиту у параметрах універсального модуля буде таким:
10 [+],

де # – роздільник; 10 – адреса модуля у *HEX*-форматі; [+] – автоматичний розрахунок *CHK*.

Формат позитивної відповіді модуля має вигляд:

> [7f] [7f] [7f] [7f] [7f] [7f] [7f] [7f] [+],

де > – роздільник, 7 – модифікатор; f – представляє дію, тобто тип переданої змінної в *ASCII*-символах в десятковому форматі зі знаком, роздільником цілої та дробової частини числа. Рядок має фіксоване число символів, заданий модифікатором, причому кількість спецкоманд відповідає кількості входів модуля; [+] – автоматичний розрахунок *CHK*.

Вікно налаштування універсального модуля *DCON* для опитування усіх аналогових каналів модуля *MBA* представлено нижче. Для зберігання значень на входах *MBA* до віртуального універсального пристрою додані вісім вихідних змінних типу *float*.

Индекс	Имя	Значение	По умолчанию
1	Request format	#10[+]	
2	Good response...	>[7f][7f][7f][7f][7f][7f]	!
3	Bad response...	?	?
4	Max response...	150	150
5	Work mode	By poll time	<input checked="" type="checkbox"/> By poll ...
6	Polling time ms	1000	100
7	Visibility	No	<input checked="" type="checkbox"/> No

3) Розглянемо приклад групового опитування усіх вхідних дискретних каналів (маски входів) модуля *МДВВ* з адресою 16₁₀.

Формат запиту у параметрах універсального модуля буде таким:

@ 10 [+],

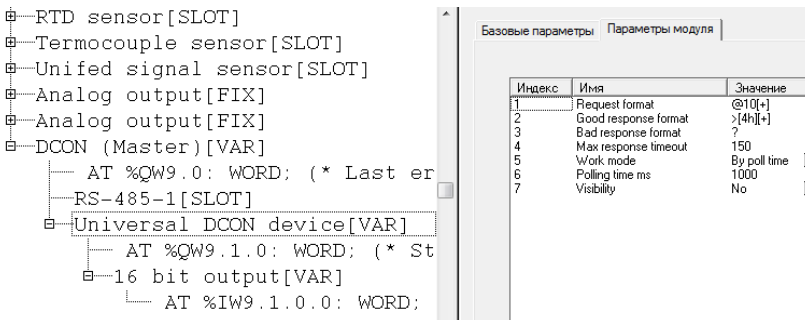
де @ – роздільник; 10 – адреса модуля у *HEX*-форматі; [+] – автоматичний розрахунок *CHK*.

Формат позитивної відповіді модуля має вигляд:

> [4h] [+],

де > – роздільник, 4 – модифікатор; h – представляє дію, тобто переданої змінної в *ASCII*-символи у *HEX*-форматі або перетворює *ASCII*-рядок з *HEX*-формату у прийняту змінну, причому рядок має фіксовану кількість символів, задану модифікатором; [+] – автоматичний розрахунок *CHK*.

Вікно налаштувань універсального модуля *DCON* для групового опитування усіх вхідних дискретних каналів (маски входів) модуля *МДВВ* представлено нижче. Для зберігання значення маски входів модуля *МДВВ* в універсальному пристрої оголошена вихідна змінна беззнакового типу розміром 16 біт.



Индекс	Имя	Значение
1	Request format	@10[+]
2	Good response format	>[4h][+]
3	Bad response format	?
4	Max response timeout	150
5	Work mode	By poll time
6	Polling time ms	1000
7	Visibility	No

4) Розглянемо приклад запису значення групової маски виходів модуля *МДВВ* з адресою 16₁₀.

Формат запиту на запис маски виходів у параметрах універсального модуля буде таким:

@ 10 [2h] [+],

де @ – роздільник; 10 – адреса модуля у *HEX*-форматі; [2h] – спецкоманда, яка складається з модифікатора – 2, яка визначає кількість символів та дії – h, яка представляє передану змінну у *ASCII*-

символах в *HEX*-форматі або перетворює *ASCII*-рядок з *HEX*-формату у змінну, що приймається; **[+]** – автоматичний розрахунок *CHK*.

Формат позитивної відповіді модуля після виконання команди запису значення маски виходів має вигляд:

> [+],

де **>** – роздільник; **[+]** – автоматичний розрахунок *CHK*.

Вікно налаштування універсального модуля *DCON* для запису значення маски дискретних виходів модуля *МДВВ* представлено нижче. Для зберігання значення маски виходів модуля *МДВВ* в універсальному пристрої додана вхідна змінна без знаку типу розміром 16 біт:

The screenshot shows a configuration tree on the left and a parameter table on the right. The tree includes:

- United signal sensor[SLOT]
- Analog output[FIX]
- Analog output[FIX]
- DCON (Master)[VAR]
 - AT %QW9.0: WORD; (* Last error
 - RS-485-1[SLOT]
 - Universal DCON device[VAR]
 - AT %QW9.1.0: WORD; (* Statu
 - 16 bit input[VAR]
 - AT %QW9.1.0.0: WORD; (*

The parameter table on the right is titled "Параметри модуля" and contains the following data:

Индекс	Имя	Значение
1	Request format	@10[2h][+]
2	Good response format	>[+]
3	Bad response format	?
4	Max response timeout	150
5	Work mode	By poll time
6	Polling time ms	1000
7	Visibility	No

5) Розглянемо приклад читання значення лічильника імпульсів на дискретному вході модуля *МДВВ* з адресою 16₁₀.

Формат запиту в параметрах універсального модуля буде таким:

@ 10 1 [+],

де **@** – роздільник; **10** – адреса модуля у *HEX*-форматі; **1** – номер каналу лічильника (у даному випадку це 2-ий вхід модуля *МДВВ*), **[+]** – автоматичний розрахунок *CHK*.

Формат позитивної відповіді модуля має вигляд:

! [5d] [+],

де **!** – роздільник, **5** – модифікатор; **d** – це дія, тобто тип змінної, яка передається у *ASCII*-коді у десятковому форматі (без знаку) або перетворює *ASCII*-рядок з десяткового формату без знаку у змінну, яка приймається, причому рядок має фіксовану кількість символів, яка задана модифікатором; **[+]** – автоматичний розрахунок *CHK*.

Вікно налаштування універсального модуля *DCON* для опитування значення лічильника на 2-му дискретному вході модуля *МДВВ* представлено нижче. Для зберігання значення лічильника до універсального пристрої додана вихідна змінна без знаку розміром 32 біта:

The screenshot shows the configuration interface for the DCON module. On the left, a tree view displays the project structure:

- Analog output[FIX]
- DCON (Master)[VAR]
 - AT %QW9.0: WORD; (* Last err
 - RS-485-1[SLCT]
 - Universal DCON device[VAR]
 - AT %QW9.1.0: WORD; (* Sta
 - 32 bit output[VAR]
 - AT %ID9.1.0.0: DWORD;

On the right, a window titled "Параметры модуля" (Module Parameters) displays a table of parameters:

Индекс	Имя	Значение
1	Request format	#101[+]
2	Good response format	[[5d][+]
3	Bad response format	?
4	Max response timeout	150
5	Work mode	By poll time
6	Polling time ms	1000
7	Visibility	No

Отже, відкомпілюйте проєкт для опитування потрібного модуля введення/виведення (*MBA, MB110-2A* або *МДВВ*), встановіть з'єднання з ПЛК та завантажте код до нього. Запустіть ПЛК на виконання проєкту і, не відключаючи середовище *CoDeSys V2.3* від ПЛК, спостерігайте за змінними проєкту та вводите значення для запису їх до модуля, наприклад для управління маскою виходів *МДВВ*.

8.6. Оформлення результатів проведення заняття

Складіть звіт про проведене заняття, де відобразить схему з'єднань та умови обміну даними. Також вкажіть, які режими використані для обміну даними.

8.7. Контрольні запитання

- 1) Які інтерфейси використовують для реалізації протоколу *DCON*?
- 2) На яких рівнях моделі зв'язку працює протокол *DCON*?
- 3) Який алгоритм роботи мережі за протоколом *DCON*?
- 4) Який формат кадру у протоколі *DCON*?
- 5) Яким чином реалізована достовірність переданих даних у протоколі *DCON*?
- 6) Які принципи адресації закладені у протоколі *DCON*?
- 7) Якого типу змінні описані у протоколі *DCON*?
- 8) Як поділяють кадри при передачі повідомлень у протоколі *DCON*?

ЗМІСТ

Вступ	3
<i>Лабораторна робота 1. Основні принципи формування запитів та аналіз відповідей у протоколі ModBus</i>	5
<i>Лабораторна робота 2. Взаємодія контролера ПЛК150 та панелі оператора СМІ-1 по інтерфейсу RS-485 за протоколом ModBus .</i>	15
<i>Лабораторна робота 3. Взаємодія панелі оператора СП310 (ИП320) та контролера ПЛК150 по інтерфейсу RS-485 за протоколом ModBus</i>	30
<i>Лабораторна робота 4. Основні принципи створення РСУ на послідовних інтерфейсах та за протоколом ModBus</i>	42
<i>Лабораторна робота 5. Основні принципи формування запитів та аналіз відповідей у протоколі OWEN</i>	61
<i>Лабораторна робота 6. Взаємодія контролера ПЛК150 та панелі оператора СМІ-1 по інтерфейсу RS-485 за протоколом OWEN . .</i>	71
<i>Лабораторна робота 7. Основні принципи створення РСУ на послідовних інтерфейсах та за протоколом OWEN</i>	85
<i>Лабораторна робота 8. Основні принципи формування запитів та аналізу відповідей у протоколі DCON</i>	96
Список джерел інформації	113
Зміст	114

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ

для проведення лабораторних робіт з навчальної дисципліни

«Розподілені системи управління»

для студентів спеціальності

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

денної та заочної (дистанційної) форм навчання

Укладачі: ЛИСАЧЕНКО Ігор Григорович

ПОДУСТОВ Михайло Олексійович

БАБІЧЕНКО Анатолій Костянтинівич

ДЗЕВОЧКО Альона Ігорівна

Відповідальний за випуск проф. Михайло Подустов

Роботу до видання рекомендував доцент Олексій Дудник

В авторській редакції

План 2020 р., поз. 333.

Підп. до друку __. __. 20. Гарнітура Times New Roman.

Видавничий центр НТУ «ХПІ».

Свідоцтво про державну реєстрацію ДК № 5478 від 21.08.2017 р.

61002, Харків, вул. Кирпичова, 2.

Самостійне електронне видання.