

## ЛАБОРАТОРНА РОБОТА № 1 (модуль 2).

### Асинхронні послідовні інтерфейси. Інтерфейс RS-232C.

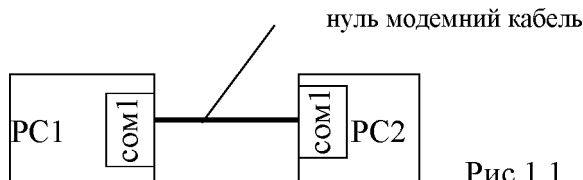
**Тривалість:** 2 акад. години (1 пара).

**Мета:** ознайомлення з принципами роботи асинхронного послідовного інтерфейсу RS-232C.

#### Лабораторна установка.

Лабораторна робота виконується робочих місцях 2-5, 7-10.

**Апаратне забезпечення.** Схема установки показана на рис.1.1.



**Програмне забезпечення.** Використовується прикладна програма для сканування і запису COM-порта комп'ютера Compt.exe.

**Загальна постановка задачі.** З'єднання між двома ПК по RS-232 проводиться з використанням нуль-модемного кабелю. У лабораторній роботі можна використати як повний так і мінімальний нуль-модемний кабель.

**УВАГА!** COM-порти ПК не ізольовані, тому для безпечного з'єднання необхідно підключати нуль-модемний кабель при вимкненому ПК, з дозволу викладача.

#### Послідовність виконання роботи.

- 1) Зібрати лабораторну установку. Запустити на виконання програму Compt.exe на PC1 та PC2. Ознайомитись з її роботою (див. додаток 1).
- 2) Налаштувати PC1 та PC2 на роботу з наступними параметрами :

	робочі місця			
	2,3	4,5	7,8	9,10
Бітова швидкість (біт/с)	9600	9600	9600	9600
кількість бітів даних	8	8	8	8
паритет	парний	непарний	парний	відсутній
кількість стопових біт	1	2	1	2

- 3) З PC1 на PC2 передати наступні символи (в 16-ковому форматі):

	робочі місця			
	2,3	4,5	7,8	9,10
	08, FF	02, FE	14, E1	AA, BB

- 4) Нарисувати часову діаграму стану сигналу на лінії як на рис.1.2 в масштабі: 1 клітинка = 1/19200 с. Нарисувати діаграму з поясненням інтерпретації сигналу приймачем PC2 при бітовій швидкості 9600 біт/с.

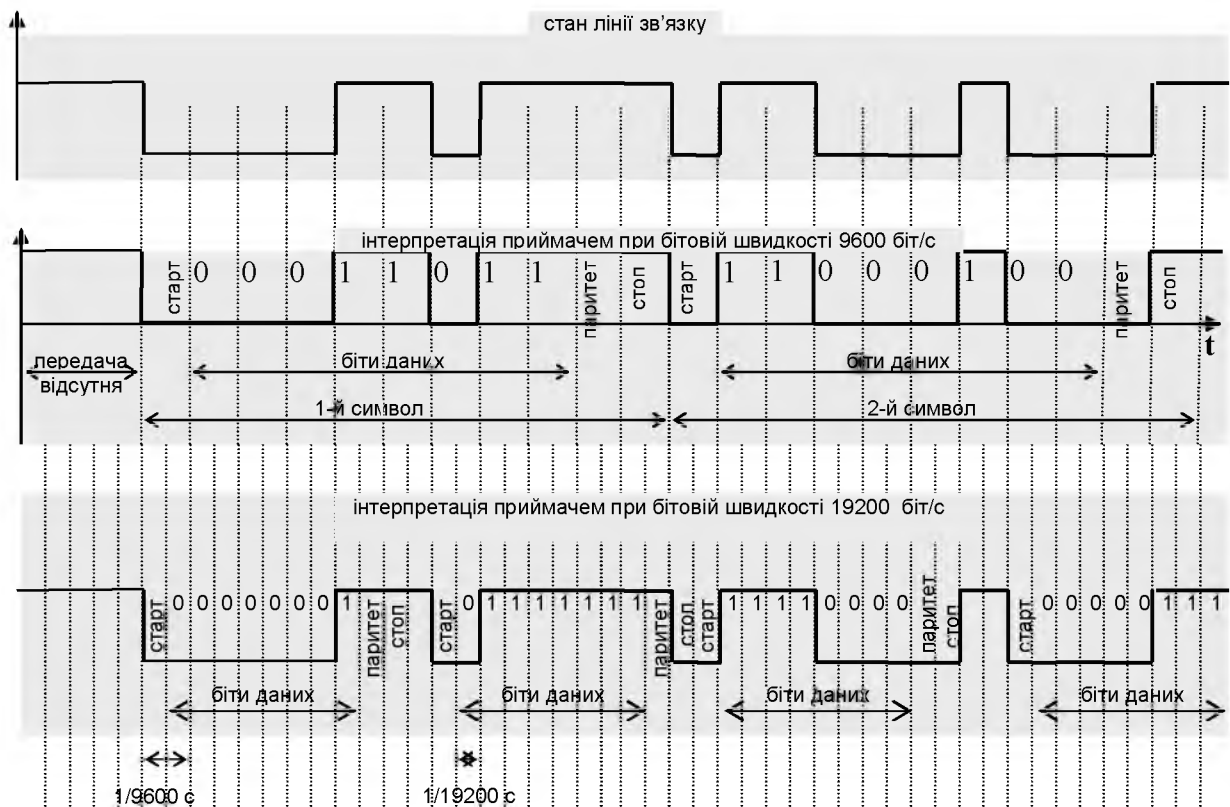


Рис.1.2. Діаграми стану лінії та інтерпретація приймачем сигналу на бітовій швидкості 9600 на прикладі передачі двох символів: 1-й – 11011000; 2-й – 00100011. Біт паритету – непарний; 1 стоповий біт.

- 5) Змінити бітову швидкість на PC2 з 9600 на 19200 біт/с. Повторити крок 3. Нарисувати діаграму з поясненням інтерпретації сигналу приймачем PC2 при бітовій швидкості 19200 біт/с. (рис.1.2.).

### Оформлення роботи.

До захисту готується рисунок з інтерпретацією сигналу приймачем при 9600 біт/с та 19200 біт/с.

### Перевірка виконання роботи та питання до захисту.

Викладачем перевіряється виконання поставленого завдання. Студент повинен пояснити отримані результати, коментуючи діаграми з інтерпретаціями.

1. Яке призначення контактів 9-пінового роз'єму для реалізації RS-232C?
2. Які основні контакти використовуються при нуль-модемному з'єднанні без синхронізуючих сигналів?
3. Чим відрізняється повний та мінімальний нуль-модемний кабель?
4. Навіщо необхідні стартовий та стопові біти при символній асинхронній передачі?
5. Роз'ясніть призначення біту паритету.
6. Роз'ясніть чому саме такі дані отримав PC2 на основі діаграми з кроку 4 та порівнянні відправлених і отриманих даних.
7. Що значить симетричний спосіб передачі по напрузі? Чим він краще за асиметричний?
8. Скільки провідників необхідно для реалізації з'єднання пристроїв по інтерфейсу RS-485 при напівдуплексному зв'язку?

9. Яка принципова відмінність між RS-422 та RS-485?
10. Навіщо використовується захисне зміщення в мережах на базі RS-485?
11. Навіщо потрібні термінатори лінії я які як правило їх номінали для RS-422 та RS-485?
12. Чим викликана необхідність в управлінні станом передавача адаптера-перетворювача інтерфейсів RS-232 – RS-485. Які схеми реалізації адаптерів Ви знаєте?

### Додаток 1.1. Робота з програмою Compt.

**Д1.1.1. Загальні відомості.** Дана програма являється утилітою для роботи з СОМ-портами комп'ютера. Серед її можливостей треба виділити наступні:

- відправляти/приймати послідовність символів представлені в 16-ковому або символьному форматі;
- налаштування СОМ-порту на різні параметри передачі/прийому;
- управляти потоком даних сигналами RTS та DTR в ручному режимі

Після завантаження, як правило видається помилка доступу до СОМ-порту, на яку не слід зважати.

**УВАГА!** Будь-які дві програми не можуть використовувати один і той же СОМ-порт одночасно (за певними винятками). Тому перед використанням Compt.exe необхідно закрити всі програми, які доступуються до СОМ1.

**Д1.1.2. Основні команди.** Якщо програма завантажилась в англomовному варіанті, перейдіть на російськомовний: *Options->Language->Russian* .

Налаштування СОМ-порту: *Настройка->СОМ-порт*

Набрати і відправити послідовність символів: *Сообщение->Отправить*

Управління сигналом RTS відбувається за допомогою відповідних кнопок на панелі інструментів: кнопка ввєрх – увімкнути, кнопка вниз – вимкнути.

Як тільки програма завантажена і налаштований порт, вона прослуховує його і у випадку появи символів, записує їх у вікно "Принято".

## ЛАБОРАТОРНА РОБОТА № 2(модуль 2).

### Формування запитів MODBUS RTU.

**Тривалість:** 2 акад. години (1 пара).

**Мета:** ознайомитись з принципами функціонування протоколу MODBUS; ознайомитись з роботою адаптера-перетворювача інтерфейсів RS-232<->RS-485.

#### Лабораторна установка.

Лабораторна робота виконується на робочих місцях 2-5, 7-10. Одна бригада – два робочі місця.

**Апаратне забезпечення.** Лабораторна установка складається з комп'ютера підключеного до термінального порту ПЛК Twido за допомогою адаптера-перетворювача інтерфейсів RS-232 <-> RS-485. Схема установки показана на рис.2.1.

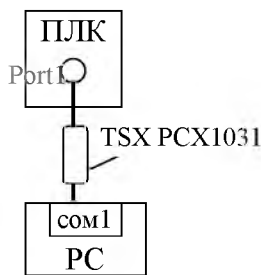


Рис.2.1.

**Програмне забезпечення.** На комп'ютері використовується прикладна програма для роботи з СОМ-портом комп'ютера Compt.exe, а також програма для генерації циклічного перевірного коду CRC16 – crcdll.exe (знаходиться в папці Compt175). На ПЛК написана програма, яка змінює значення змінних відповідно до варіанту. З'єднання ПЛК та ПК реалізується за допомогою Modbus RTU, де ПЛК являється Slave, а ПК разом з Compt.exe являється Master.

**Загальна постановка задачі.** Лабораторна бригада реалізовує роботу MODBUS RTU Master, тобто формує запити для Slave (ПЛК) та обробляє його відповіді. Основні функції, можливі помилки та їх коди наведені в додатках 2.1 та 2.2. Контрольну суму необхідно вираховувати за допомогою програми crcdll.exe, яка в аргументах приймає байти кадру в 16-ковому вигляді, які розділені " " (пробілом), а результатом видає 2 байти суми CRC16.

Необхідно врахувати особливості роботи кабелю TSX PCX 1031 та його вплив на роботу Port1 в Twido. Принципи функціонування кабелю описана в додатку 2.3.

#### Послідовність виконання роботи.

1) Зібрати лабораторну установку. Запустити на виконання програму Compt.exe на PC.

**УВАГА!** Port1 ПЛК Twido і СОМ-порт ПК не ізольовані, тому для безпечного з'єднання необхідно підключати перетворювачі при вимкнених ПЛК, з дозволу викладача.

2) Налаштувати СОМ-порт на роботу з наступними параметрами :

Параметр	робоче місце							
	2	3	4	5	7	8	9	10
Адреса Slave	2	3	4	5	7	8	9	10
бітова швидкість	19200	19200	19200	19200	9600	9600	9600	9600
паритет	парн	парн	парн	парн	непар	непар	непар	непар
стоп біт	1	1	1	1	1	1	1	1

- 3) Виставити кабель в необхідне положення: Port1 Twido працює в режимі визначеному конфігурацією; перетворювач управляється зі сторони RS-485.
- 4) Сформувати та занотувати запит на читання у Slave 3-х внутрішніх регістрів (%MW), починаючи з вказаного (приведений в таблиці нижче в десятковому форматі)

Параметр	робоче місце							
	2	3	4	5	7	8	9	10
номер регістру	123 <sub>10</sub>	68 <sub>10</sub>	34 <sub>10</sub>	176 <sub>10</sub>	34 <sub>10</sub>	11 <sub>10</sub>	67 <sub>10</sub>	95 <sub>10</sub>

- 5) По сформованому в п.4 запиту визначити контрольну суму CRC16 за допомогою програми crcdll.exe. Занотувати повний формат кадру разом з отриманим CRC16.
- 6) Відправити кадр, визначений в п.5. по мережі за допомогою Compt.exe. Отриману відповідь занотувати. Значення змінних перевести в десятковий формат та занотувати.
- 7) Перевірити, чи вірно отримані дані по стану поля CRC16. Результат проаналізувати та занотувати.
- 8) Відправити той самий запит, що у пункті 6, але при іншому біті паритету (якщо був парний – то при непарному і навпаки). Отриманий результат проаналізувати і занотувати.
- 9) Відправити запит на читання 3-х регістрів, починаючи з 1000<sub>10</sub>. Отриманий результат проаналізувати і занотувати.
- 10) Сформувати запит на запис Slave 2-х внутрішніх регістрів, починаючи з вказаного (приведений в таблиці нижче в десятковому форматі). Визначити контрольну суму CRC16 за допомогою програми crcdll.exe. Занотувати формат кадру і відправити по мережі. Отриману відповідь проаналізувати та занотувати.

Параметр	робоче місце							
	2	3	4	5	7	8	9	10
номер регістру	2 <sub>10</sub>	3 <sub>10</sub>	4 <sub>10</sub>	5 <sub>10</sub>	7 <sub>10</sub>	8 <sub>10</sub>	9 <sub>10</sub>	10 <sub>10</sub>
знач. регістр 1	34 <sub>10</sub>	22 <sub>10</sub>	44 <sub>10</sub>	56 <sub>10</sub>	68 <sub>10</sub>	12 <sub>10</sub>	14 <sub>10</sub>	68 <sub>10</sub>
знач. регістр 2	2345 <sub>10</sub>	1234 <sub>10</sub>	3456 <sub>10</sub>	4567 <sub>10</sub>	5678 <sub>10</sub>	9112 <sub>10</sub>	1123 <sub>10</sub>	7654 <sub>10</sub>

### Оформлення роботи.

До захисту готувляться всі занотовані результати з поясненнями.

### Перевірка виконання роботи та питання до захисту.

Викладачем вибірково перевіряється виконання всіх пунктів роботи та занотовані результати. Кожен результат студент повинен пояснити. У випадку виникнення помилок або запитань щодо проведення певного пункту, цей пункт необхідно буде повторити.

1. Розкажіть про основи функціонування MODBUS Application Protocol. Який формат повідомлення MODBUS PDU?

2. Які функції використовуються для доступу до даних процесу?
3. Як формуються повідомлення-запити та повідомлення відповіді для читання та запису діапазону вхідних та вихідних регістрів?
4. Які ситуації можливі при обробці запиту MODBUS Клієнта? Наведіть приклади відповідей про помилку.
5. Прокоментуйте основи функціонування MODBUS RTU/ASCII в контексті моделі OSI. Як пов'язана модель функціонування обміну на прикладному рівні з функціонуванням на каналному?
6. Який метод доступу до шини використовується в MODBUS RTU/ASCII на каналному рівні? Як на цьому рівні проводиться контроль за правильністю доставки бітової послідовності?
7. Поясніть призначення поля адреси. Чому виникає необхідність в адресації? Чи використовується поле адреси при з'єднанні точка-точка в MODBUS RTU?
8. В чому відмінність роботи MODBUS RTU та MODBUS ASCII на каналному та фізичному рівні?
9. Розкажіть про принципи побудови кадрів для MODBUS RTU та MODBUS ASCII.
10. Які інтерфейси, бітова швидкість та топологія використовується для MODBUS RTU/ASCII на фізичному рівні? Яким чином вузли можуть підключатися до загальної шини?
11. Чим викликана необхідність в управлінні станом передавача адаптера-перетворювача інтерфейсів RS-232 <--> RS-485.
12. Якими способами може управлятися стан передавача адаптеру TSX PCX 1031? Як задати кожний із режимів? Коли використовується кожний із них?

## **Додаток 2.1. Формат функцій MODBUS (MBAP)**

Повний список кодів а також специфікацію протоколу можна знайти на офіційному Веб сайті MODBUS-IDA - [www.MODBUS.org](http://www.MODBUS.org). В додатку детально розглянуто тільки найбільш вживані функції MODBUS для обміну даними процесу. Номер функції дається в шістнадцятковому форматі. Скорочення в дужках *Hi* та *Lo* вказують відповідно на старший та молодший байти. Тобто, якщо для вказівки адреси початкової змінної необхідно двобайтове слово, то значення старшого байта буде передаватись в полі з позначенням *Hi*, а молодшого – відповідно *Lo*.

**Д.2.1.1. Код функції 01<sub>16</sub> – читання статусу Coils (дискретних вихідних бітів).** Повідомлення-запит вміщує адресу початкового біту і кількість бітів для читання. Біти нумеруються починаючи з 0. У повідомленні-відповіді кожне значення змінної передається одним бітом, тобто в одному байті пакується статус 8 бітових змінних. Якщо кількість їх не кратно восьми, інші біти в байті заповнюються нулями. Лічильник вміщує кількість байт в полі даних.

Запит:

Код функції	01
Адреса початкового біту (Hi)	0 до FFFF <sub>16</sub>
Адреса початкового біту (Lo)	
Кількість біт (Hi)	1 до 7D0 <sub>16</sub> (2000)
Кількість біт (Lo)	

Відповідь:

код функції	01
лічильник байт	N
Значення бітів (перші 8 біт)	0 до FF <sub>16</sub>
Значення бітів (наступні 8 біт)	0 до FF <sub>16</sub>
...	
Значення бітів (N-ні 8 біт)	0 до FF <sub>16</sub>

**Д.2.1.2. Код функції 02<sub>16</sub> – читання статусу дискретних входів.** Формат даного запиту такий же як попереднього, за винятком поля функції.

**Д.2.1.3. Код функції 03<sub>16</sub> – читання значення вихідних/внутрішніх регістрів.** Повідомлення-запит вміщує адресу початкового вихідного/внутрішнього регістру (двохбайтове слово), і кількість регістрів для читання. Регістри нумеруються починаючи з 0.

Запит:

код функції	03
Адреса початкового регістру (Hi)	від 0 до FFFF <sub>16</sub>
Адреса початкового регістру (Lo)	
Кількість регістрів (Hi)	від 1 до 7D <sub>16</sub> (125)
Кількість регістрів (Lo)	

Відповідь:

код функції	01
лічильник байт	N*2
Значення 1-го регістру (Hi)	0 до FFFF <sub>16</sub>
Значення 1-го регістру (Lo)	
...	
Значення N-го регістру (Hi)	0 до FFFF <sub>16</sub>
Значення N-го регістру (Lo)	

У відповідному повідомленні в полі даних кожний регістр передається двома байтами.

**Д.2.1.4. Код функції 04<sub>16</sub> – читання значення вхідних регістрів.** Формат даного запиту такий же як попереднього, за винятком поля функції.

**Д.2.1.5. Код функції 0F<sub>16</sub> – запис декількох вихідних/внутрішніх бітів.** В запиті вказується початкова адреса біту, кількість біт для запису, лічильник байтів і безпосередньо значення. В ширококомовній передачі біти записуються всім серверам. Розглянемо приклад для встановлення наступних бітових вихідних/внутрішніх змінних:

Байт 1								Байт 2							
26	25	24	23	22	21	20	19	--	--	--	--	--	--	28	27
1	1	0	0	1	1	0	1							0	1

В таблиці показана відповідність адреси змінної, починаючи з 19-ї, і значення біту. Для зручності біти розміщені у тому порядку, що і передаються. В другому байті корисні тільки 2 перші біти, значення інших не буде прийнято до уваги, оскільки кількість бітів вказані у кадрі. Запит та відповідь будуть мати такий вигляд:

Запит:

Функція	0F
Адреса початкового біту (Hi)	00
Адреса початкового біту (Lo)	13
Кількість бітів (Hi)	00
Кількість бітів (Lo)	0A
Лічильник байтів	02
Дані(змінні 19-26)	CD
Дані(змінні 27-28)	01

Відповідь:

Функція	0F
Адреса початкового біту (Hi)	00
Адреса початкового біту (Lo)	13
Кількість бітів (Hi)	00
Кількість бітів (Lo)	0A

### Д.2.1.6. Код функції 10<sub>16</sub> – запис декількох вихідних/внутрішніх реєстрів.

Запит:

Функція	10 <sub>16</sub>
Адреса початкового реєстру (Hi)	0 до
Адреса початкового реєстру (Lo)	FFFF <sub>16</sub>
Кількість реєстрів (Hi)	1 до
Кількість реєстрів (Lo)	007B <sub>16</sub> (123)
Лічильник байтів	2*N
Дані (1-й реєстр Hi)	0 до
Дані (1-й реєстр Lo)	FFFF <sub>16</sub>
...	
Дані (N-й реєстр Hi)	0 до
Дані (N-й реєстр Lo)	FFFF <sub>16</sub>

Відповідь:

Функція	10
Адреса початкового реєстру (Hi)	0 до FFFF <sub>16</sub>
Адреса початкового реєстру Lo	
Кількість реєстрів (Hi)	1 до 007B <sub>16</sub>
Кількість реєстрів (Lo)	(123)

### Додаток 2.2. Повідомлення про помилки MODBUS (MBAP)

Ці повідомлення стосуються всіх типів MODBUS, але першопочатково були визначені для MODBUS Serial (RTU/ASCII).

При запиті Клієнта до Серверу, можуть мати місце наступні ситуації:

- якщо Сервер прийняв запит без комунікаційних помилок, і може нормально розпізнати запит, він повертає нормальну відповідь;
- якщо Сервер не прийняв запит, відповідь не повертається. Клієнт очікує відповіді протягом певного тайм-ауту;
- якщо Ведений (для MODBUS Serial) прийняв кадр, але знайшов комунікаційну помилку (паритет, помилка контрольної суми), то кадр-відповідь не повертається, а Ведучий чекає відповіді на запит протягом певного тайм-ауту;
- якщо Сервер прийняв запит без комунікаційної помилки, але не може виконати замовлену функцію (наприклад, читання не існуючих виходів або реєстрів), Сервер повертає повідомлення про помилку і її причини.

Таблиця Д2.1

Список кодів помилок

Код	Назва	Опис
01	ILLEGAL FUNCTION	Прийнятий код функції не може бути оброблений на Сервері
02	ILLEGAL DATA ADDRESS	Адреса даних вказана в запиті не доступна даному Серверу .
03	ILLEGAL DATA VALUE	Величина, вміщена в полі даних запиту являється не допустимою величиною для Серверу .
04	SLAVE DEVICE FAILURE	Невиправна помилка мала місце поки Сервер намагався виконати дію запиту.
05	ACKNOWLEDGE	Сервер прийняв запит і обробляє його, але необхідний певний час. Ця відповідь захищає Клієнта від генерації помилки тайм-ауту.
06	SLAVE DEVICE BUSY	Сервер зайнятий обробкою команди, Клієнт повинен повторити запит пізніше.
07	NEGATIVE ACKNOWLEDGE	Невдалий програмний запит (для функцій 13 і 14).
08	MEMORY PARITY ERROR	Сервер хоче читати розширену пам'ять, але знайшов помилку паритету.

Повідомлення про помилку має два поля які відрізняються від полів нормальної відповіді:



ПОЛЕ КОДУ ФУНКЦІЇ: при нормальній відповіді сервер повертає в цьому полі той номер функції, який потребував Клієнт. У всіх кодах функції старший біт встановлений в 0. При поверненні повідомлень про помилку, Сервер встановлює цей біт в 1, по чому Клієнт може ідентифікувати наявність помилки.

ПОЛЕ ДАНИХ: В цьому полі при помилці повертається її код.

### Додаток 2.3. Робота адаптеру перетворювача TSX PCX 1031

Адаптер-перетворювач TSX PCX 1031 (див.рис.2.2) використовується для підключення пристроїв з RS-232C до термінального порту TSX Twido, TSX Micro, TSX Premium, або коробки підключення TSX P ACC01. Управління станом трансмітера RS-485 в адаптері-перетворювачі може управлятися одним із двох способів, які задаються перемикачем: зі сторони RS-232C сигналом RTS (позиція "**MULTI**"), або зі сторони термінального порту ПЛК (позиція "**DIRECT**"). Оскільки в стандарті RS-485 немає виділених ресурсів для режиму DIRECT, то він використовується *тільки для підключення адаптера безпосередньо до термінального порту ПЛК*, тобто у випадку з'єднання точка-точка.

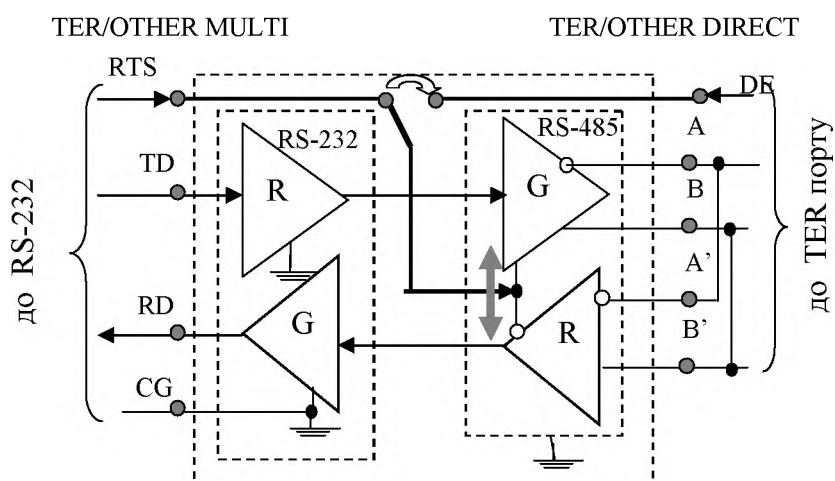


Рис.2.2. Режими роботи адаптера перетворювача RS-232<->RS-485 TSX PCX

Враховуючи, що TSX PCX 1031 використовується в якості кабелю програмування вищезазначених контролерів, він надає можливість переводити підключений термінальний порт ПЛК в режим *програмування*. Перемикач кабелю в позиції "**TER**" переводить підключений порт ПЛК в режим програмування, незалежно від дійсної його конфігурації. Позиція "**OTHER**" переводить термінальний порт в той режим, який визначається його дійсною конфігурацією.

## ЛАБОРАТОРНА РОБОТА № 3 (модуль 2).

### Використання шини MODBUS RTU для зв'язку SCADA програм з ПЛК.

#### ЧАСТИНА 1.

**Тривалість:** 1 акад. година (0.5 пари).

**Мета:** Навчитись створювати конфігурацію SCADA-програм для роботи з мережами, основаними на асинхронних послідовних інтерфейсах на прикладі MODBUS RTU і SCADA VijeoCitect в режимі точка-точка.

#### Лабораторна установка.

Лабораторна робота виконується на робочих місцях 2-5, 7-10. Одна бригада – два робочі місця.

**Апаратне забезпечення.** Лабораторна установка складається з комп'ютера підключеного до термінального порту ПЛК Twido за допомогою адаптера-перетворювача інтерфейсів RS-232 <-> RS-485. Схема установки показана на рис.3.1.

**Програмне забезпечення.** На комп'ютері використовується SCADA-програма VijeoCitect. На ПЛК написана програма, яка змінює значення змінних відповідно до варіанту. З'єднання ПЛК та ПК реалізується за допомогою MODBUS RTU, де ПЛК являється Slave, а драйвер для I/O Server VijeoCitect являється Master.

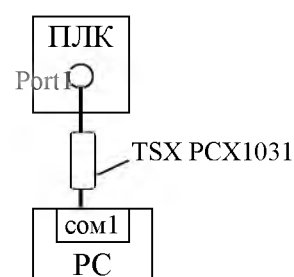


Рис.3.1.

**Загальна постановка задачі.** Ціль роботи – реалізувати поставлену задачу, оформлену у вигляді схеми інформаційних потоків. Ця задача передбачає обмін даними між SCADA програмою VijeoCitect (вірніше його I/O Server) та TSX Twido.

У лабораторній роботі використовується класичний підхід, де SCADA являється Клієнтом, тобто ініціює обмін. Для MODBUS RTU Клієнт завжди виконується на вузлі з правами Ведучого, тому драйвер MODBUS RTU для VijeoCitect повинен реалізовувати протокол MODBUS Master. Налаштування I/O Server та драйверів MODBUS в інструментальній середовищі VijeoCitect описані в додатку 3.1.

Для такого з'єднання адаптер-перетворювач може працювати в режимі управління трансмітером з боку контролера .

**УВАГА!** Термінальний порт ПЛК і СОМ-порт ПК не ізолювані, тому для безпечного з'єднання необхідно підключати перетворювачі при вимкнених ПЛК, з дозволу викладача.

**Деталізований опис постановки задачі.** В лабораторній роботі необхідно реалізувати задачу, показану на схемі інформаційних потоків (рис.3.2).

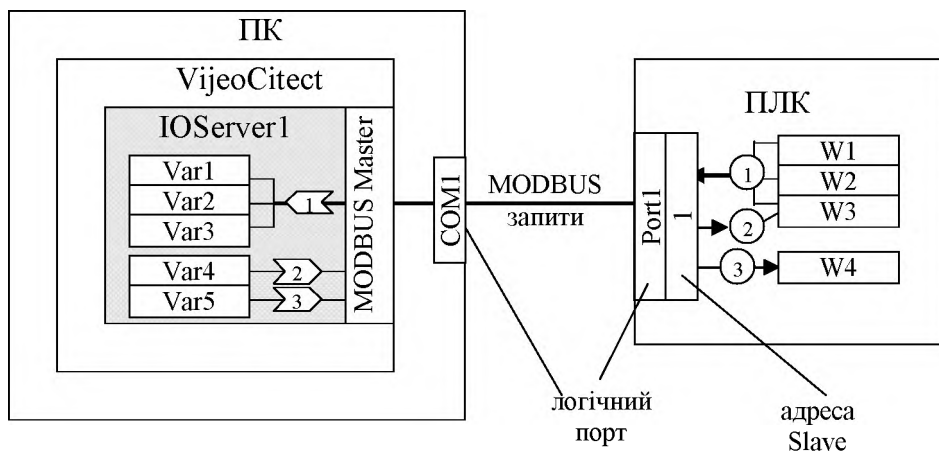


Рис.3.2. Схема інформаційних потоків до частини 1 лабораторної роботи №3.

Коротка характеристика інформаційних потоків (за номерами):

1) Клієнтською стороною для цього потоку (так як і для інших двох) є SCADA, отже умовно можна сказати, що починається він саме з IOMaster. Даний потік реалізовує зчитування даних для змінних Var1, Var2, Var3 1 раз/5с. Для Var1 джерелом буде W1 (вибирається відповідно до варіанту), для Var2 - W2, Var3 - W3. Реалізується потік сервісом MODBUS запитів (у MODBUS RTU інших сервісів не існує). Продовження потоку йде через термінальний порт Port1 ПЛК. На цьому портові ПЛК являється MODBUS Slave з адресою 1 (залежить від робочого місця). Потік закінчується змінними, з вказівкою серверної частини потоку (круг).

2) Другий потік повинен реалізувати запис вихідного значення змінної Var4 в змінну W3, при його зміні в Citect.

3) Третій потік повинен реалізувати запис вихідного значення змінної Var5 в W4, при зміні його значення в Citect.

### Послідовність виконання роботи.

1) Зібрати лабораторну установку.

**УВАГА! Port1 ПЛК Twido і COM-порт ПК не ізолювані, тому для безпечного з'єднання необхідно підключати перетворювачі при вимкнених ПЛК, з дозволу викладача.**

2) Завантажити Vijeo Citect Explorer. Створити I/O Server, Board, Port, Device та VariableTags відповідно до завдання (схема інформаційних потоків) та наступної таблиці:

Параметр	робоче місце							
	2	3	4	5	7	8	9	10
Адреса Slave	2	3	4	5	7	8	9	10
бітова швидкість	19200	19200	19200	19200	9600	9600	9600	9600
паритет	парн	парн	парн	парн	непар	непар	непар	непар
стоп біт	1	1	1	1	1	1	1	1
адреса регістру W1	123	68	34	176	34	11	67	95
адреса регістру W2	124	69	35	177	35	12	68	96
адреса регістру W3	125	70	36	178	36	13	69	97
адреса регістру W4	2	3	4	5	7	8	9	10

- 3) В Graphics Builder створити сторінку з елементами для відображення (Var1, Var2, Var3) та вводу змінних (Var4, Var5).
- 4) Запустити виконавчу систему Vijeo Citect, попередньо налаштувавши правильні параметри запуску (Tools-> Computer Setup Wizard). Оцінити орієнтовну швидкість відновлення змінних, та занотувати.
- 5) Активізувати кеш та змінити час відновлення кешу на 5 с. Перезапустити виконавчу систему, оцінити орієнтовну швидкість відновлення змінної та занотувати.

### **Оформлення роботи.**

До захисту готовляться всі занотовані результати з поясненнями.

### **Перевірка виконання роботи та питання до захисту.**

Викладачем вибірково перевіряється виконання всіх пунктів роботи та занотовані результати. Кожен результат студент повинен пояснити. У випадку виникнення помилок або запитань щодо проведення певного пункту, цей пункт необхідно буде повторити.

1. Ким як правило є база даних реального часу SCADA при з'єднанні її з контролером по MODBUS RTU: клієнтом чи сервером? Скільки клієнтських та серверних прикладних процесів може функціонувати в одній мережі MODBUS RTU?
2. Яким чином на схемі можуть показуватися клієнтські та серверні частини інформаційних потоків? Покажіть та прокоментуйте шлях всіх потоків на робочій схемі до лабораторної роботи .
3. Розкажіть про відмінність в реалізації адресного простору серверів MODBUS.

## **ЧАСТИНА 2.**

**Тривалість:** 1 акад.година (0.5 пари).

**Мета:** Навчитись створювати конфігурацію SCADA для роботи з MODBUS RTU в шинних топологіях; отримати навички аналізу мережного обміну з використанням програм сканерів.

### **Лабораторна установка.**

Лабораторна робота виконується на робочих місцях 7-9. В ній одночасно приймають участь студенти всіх бригад підгрупи.

**Апаратне забезпечення.** Схема установки показана на рис.3.3.

ПЛК7, ПЛК8, ПЛК9 – контролери Twido LMDA20DTK.

КП1, КП2, КП3 – коробка підключення TSX P ACC01.

**Програмне забезпечення.** На комп'ютері PC8 використовується SCADA-програма VijeoCitect, на PC7 та PC9 програма-сканер COM-порта compt.exe. З'єднання ПЛК та PC7 реалізується за допомогою MODBUS RTU, де ПЛК 7,8,9 являються Slave, а драйвер для I/O Server VijeoCitect являється Master.

**Загальна постановка задачі.** Ціль роботи – реалізувати поставлену задачу, оформлену у вигляді схеми інформаційних потоків, та за допомогою програми-сканера проаналізувати роботу мережі. Ця задача передбачає обмін даними між SCADA програмою VijeoCitect (вірніше його I/O Server) та декількома TSX Twido.

У лабораторній роботі SCADA являється Клієнтом, тому драйвер MODBUS RTU для VijeoCitect повинен реалізовувати протокол MODBUS Master. ПЛК являються Веденими.

Враховуючи мультиточкове з'єднання через спеціальні коробки TSX P ACC01, адаптер-перетворювач повинен працювати в режимі управління трансмітером з боку RS-232 (див. Додаток 3.2). Крім того, необхідно вірно налаштувати опції VijeoCitect, для управління сигналом RTS.

**УВАГА! Термінальний порт ПЛК і COM-порт ПК не ізольовані, тому для безпечного з'єднання необхідно підключати перетворювачі при вимкнених ПЛК, з дозволу викладача.**

**Деталізований опис постановки задачі.** Необхідно реалізувати наступну схему інформаційних потоків (рис.3.4):

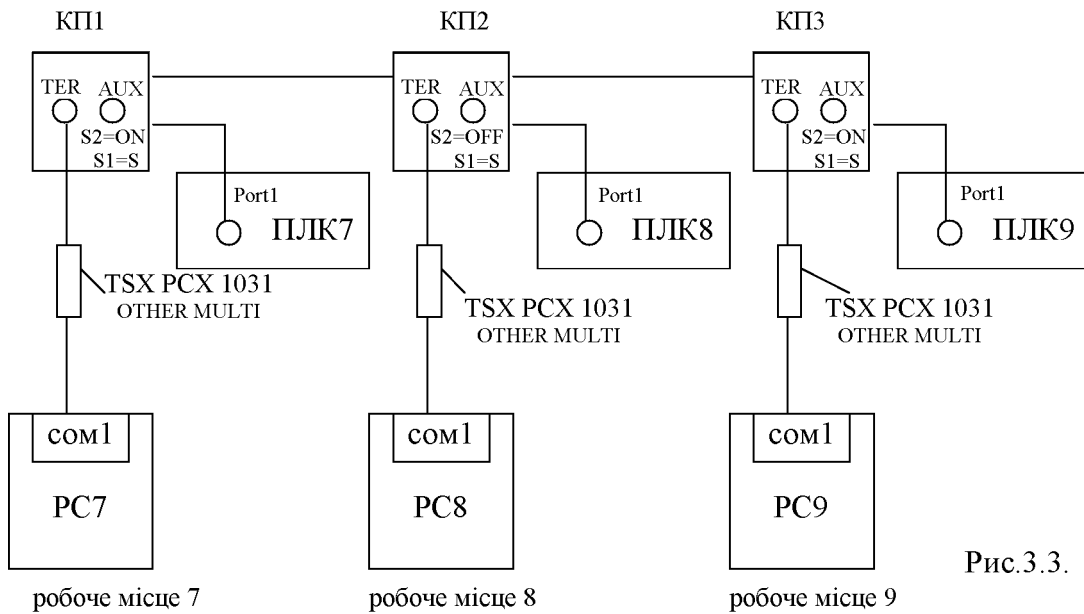


Рис.3.3.

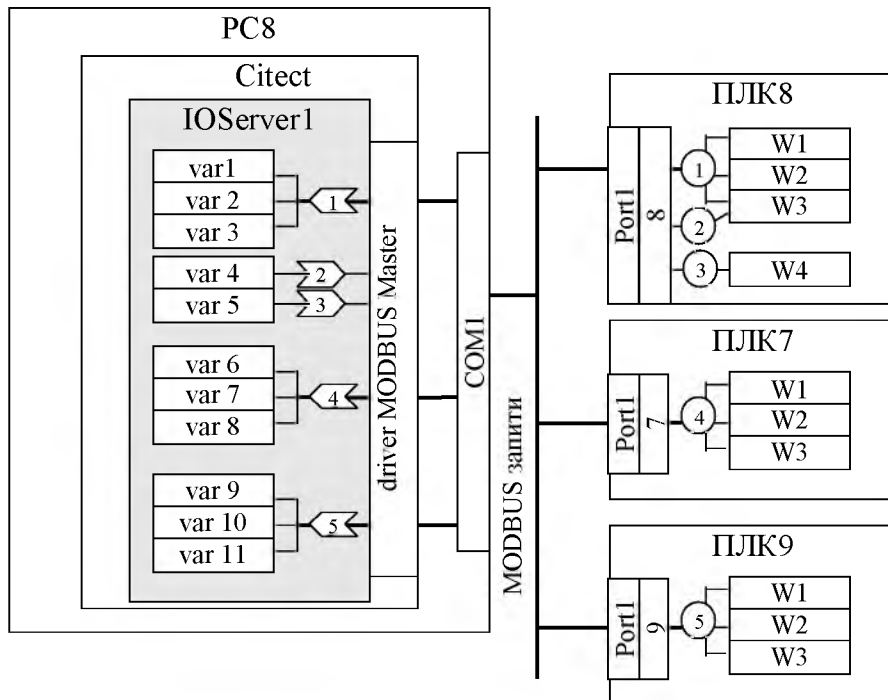


Рис.3.4.

Коротка характеристика інформаційних потоків (за номерами):

Характеристика інформаційних потоків співпадає з ЧАСТИНОЮ 1 даної лабораторної роботи:

- потоки 1,2,3 аналогічно робочому місцю 8;
- потік 4 аналогічно як потік 1 для роб. місця 7;
- потік 5 аналогічно як потік 1 для роб. місця 9.

### **Послідовність виконання роботи.**

1) Зібрати лабораторну установку.

**УВАГА! Port1 ПЛК Twido і COM-порт ПК не ізолювані, тому для безпечного з'єднання необхідно підключати перетворювачі при вимкнених ПЛК, з дозволу викладача.**

2) Запустити на виконання Citect Explorer і створити проект, відповідно до завдання, якщо змінні по вузлам (в десятковому форматі):

- ПЛК7: W1,W2,W3 відповідно реєстри 34,35,36.
- ПЛК8: W1,W2,W3,W4 відповідно реєстри 11,12,13,8.
- ПЛК9: W1,W2,W3 відповідно реєстри 67,68,69.

Звернути увагу, що система повинна управляти перетворювачем інтерфейсів. При створенні проекту врахувати, що час відновлення змінних повинен дорівнювати 5с.

3) В Graphics Builder створити сторінку з елементами для відображення та вводу змінних. Запустити виконавчу систему Vijeo Citect. Досягнути позитивного виконання задачі відповідно до рис.3.4.

4) На PC7 та PC9 запустити на виконання сканер COM-порту compt.exe з такими ж параметрами передачі, що і для системи з MODBUS (RTS в compt.exe повинен бути вимкнений, щоб не вмикати трансмітер перетворювача!).

5) Проаналізувати отримані дані з compt.exe. Звернути увагу на те, яким чином VijeoCitect проводить опитування реєстрів для кожного ПЛК, скільки запитів необхідно для реалізації читання всіх реєстрів з всіх ПЛК. Зробити копію екрана з вікном compt.exe, в якому відображаються отримані протягом 1 хв запити. Результати аналізів занотувати.

6) Збільшити адресу реєстру var3 на 10 (наприклад з %MW13 до %MW23). Повторити пункт 3.

7) Повторювати пункти 3 та 4 потрібно до тих пір, поки для читання реєстрів з ПЛК8 не будуть йти по 2 запити. Визначити приблизну максимальну величину різниці між номерами адрес реєстрів для реалізації групових запитів в SCADA VijeoCitect для протоколу MODBUS. Результат запишіть.

### **Оформлення роботи.**

До захисту готувляться всі занотовані результати з поясненнями.

#### **Перевірка виконання роботи та питання до захисту.**

Викладачем вибірково перевіряється виконання всіх пунктів роботи та занотовані результати. Кожен результат студент повинен пояснити. У випадку виникнення помилок або запитань щодо проведення певного пункту, цей пункт необхідно буде повторити.

1. Розкажіть про необхідність використання активної коробки TSX P ACC01. Які правила її підключення?

2. В якому режимі повинен працювати адаптер TSX PCX1031 при підключенні до шини через коробку TSX P ACC01?
3. Якими Веденими (адреси) на шині були PC7 та PC9?
4. Роз'ясніть відмінність понять: Клієнт-Сервер та Ведучий-Ведений.
5. Чи можливо в MODBUS RTU забезпечити передачу від Slave до Slave? Якщо можливо, то яким чином?
6. Яким чином в VijeoCitect настроюється управління адаптером-перетворювачем інтерфейсів? Навіщо його настроювати в даній частині лабораторної роботи?
7. Роз'ясніть результати пунктів 4-6.

### Додаток 3.1. Конфігурування VijeoCitect для роботи з MODBUS RTU.

**Д.3.1.1. Кластерна система VijeoCitect.** Система виконання VijeoCitect складається із клієнтської (наприклад Дисплейні клієнти) та серверної частини. Всі сервери (I/O Server, Alarm Server, Trend Server, Report Server) запускаються в межах визначених кластерів – логічне об'єднання серверів на різних комп'ютерах. Найбільш простий варіант системи - *Standalone*, тобто коли всі типи серверів знаходяться в межах одного комп'ютеру, а система має один кластер. Саме такий тип використовується в лабораторній роботі. В цьому випадку на одному ПК може знаходитись тільки один основний сервер кожного виду. Тим не менше необхідно створити один кластер, до якого всі сервери будуть прив'язуватись по замовченню. Тобто при створенні серверів в системах такого типу крім його імені нічого вводити не потрібно.

#### Д.3.1.2. Загальні концепції настройки обміну даними вводу/виводу в VijeoCitect.

В VijeoCitect за обмін даними вводу/виводу (база даних реального часу) відповідає I/O Server. Структура комунікацій має наступний вигляд (рис.3.5):

- I/O Servers
  - Boards
  - Ports
  - Modems
  - I/O Devices

Тобто в CitectExplorer в розділі Communications створюється I/O Server, в межах якого визначаються логічні або фізичні комунікаційні плати (Boards), на кожній з яких є комунікаційні порти (Ports). При необхідності використання модемів вони налаштовуються в Modems. Порти являють собою виходи на промислову мережу, де знаходяться пристрої, які в VijeoCitect конфігуруються в розділі I/O Devices.



Рис.3.5.Настройка комунікацій в VijeoCitect

Для зв'язку з зовнішніми засобами вводу/виводу в лабораторній роботі не використовується OFS. Обмін по MODBUS RTU проводиться через драйвер протоколу MODBUS.

Для автоматичного конфігурування всієї структури потрібної комунікації бажано спочатку скористатися помічником *Express I/O Device Setup*, після чого всі

інші настройки підрихтувати під задану задачу. В наступних підпунктах розписано призначення настройок всіх елементів структури комунікації для MODBUS RTU.

#### **Д.3.1.3. Boards.**

*Board Name* – довільне унікальне ім'я плати

*Board Type* – тип плати, при використанні COM-портів вибирається тип COMx

*Address*, *I/O Port*, *Interrupt*, *Special Opt* - при використанні COM-портів залишається

#### **Д.3.1.4. Ports.**

*PortName* - довільне унікальне ім'я порта

*Port Number* - номер порта, для COM1=1 для COM2=2 і т.д.

*Board Name* – ім'я плати

*Baud Rate* - швидкість передачі даних (наприклад 9600)

*Data Bits* – кількість бітів даних в символі, для MODBUS RTU = 8, для MODBUS ASCII = 7.

*Stop Bits* - кількість стопових бітів в символі

*Parity* – тип біту паритету

*Special Opt* - Спеціальні настройки для порту. Для COM портів вони необхідні для настройки поведінки сервісних сигналів RS-232. Нижче наведені опції для настройки поведінки сигналу RTS:

-t: драйвер включити RTS тільки при передачі

-ti: RTS включається тоді, коли достатньо місця в приймальному буфері, щоб отримати вхідні символи, і відключається коли недостатньо.

-to: RTS включається тоді, коли у вихідному буфері є символи для передачі

-tPRE,POST: перед передачею повідомлення драйвер включає RTS на PRE мілісекунд, передає повідомлення, очікує POST мілісекунд перед відключенням RTS.

#### **Д.3.1.5. I/O devices.**

*Name* – довільне унікальне ім'я пристрою (вузла в промисловій мережі)

*Number* – унікальний номер для пристрою

*Address* – адреса пристрою на мережі, для MODBUS RTU/ASCII відповідає адресі Веденого

*Protocol* – тип протоколу, для MODBUS RTU/ASCII вибирається один із типів MODBUS пристроїв (див. Д.3.1.7)

*Port Name* – ім'я порту, до якого підключений даний вузол

*Memory* – у випадку створення внутрішніх змінних SCADA, вибирається TRUE

#### **Д.3.1.6. Розширені настройки I/O devices**

Для розширеної настройки I/O devices існують додаткові поля. Поява додаткових опцій настройки у вкладці I/O devices (і у будь яких інших вікнах настройки) доступна по кнопці "F2".

Серед полів розширеної настройки слід виділити настройки кеша. При відсутності кешування даних, I/O Server при кожному запиті на оновлення даних анімації від Дисплейних клієнтів, будуть проводити обмін з джерелом даних, тобто з I/O Device. Якщо для даного I/O Device буде виділений та налаштований кеш, то Дисплейні клієнти будуть отримувати дані саме з кешу а не безпосередньо з пристрою. Оновлення кешу буде проводитись з періодичністю *Cache Time*, а.

*Cache* – для включення кешування виставляється TRUE.

*Cache Time* – при включеному кешуванні, виставляється період відновлення кешу в мілісекундах (рекомендується не менше 300 мс)



### Д.3.1.7. Особливості вибору протоколу MODBUS в I/O devices.

Реалізація MODBUS протоколу може відрізнятися в пристроях різних виробників, наприклад в адресації змінних або в розміщенні бітів в пам'яті і т.д. Крім того адреси змінних (поле Address) пристроїв в інтерфейсі настройки тегів VijeoCitect (Variable Tag) теж відрізняються. Тому в VijeoCitect передбачені декілька типів протоколу MODBUS, які враховують ці особливості. Ці протоколи відрізняються:

- номером першого регістру/катушки в пристрої; деякі пристрої (TSX Quantum або Momentum) використовують регістр/катушку з номером 1 як самий перший, а інші (наприклад TSX Premium) використовують номер 0;
- яким чином адресується самий перший біт в межах регістру: від 0 до 15, чи від 1 до 16.
- яким чином задається адреса змінної пристрою в Variable Tag (див. Д.3.1.8);

Для вузлів з адресацією як в TSX Twido, TSX Micro та TSX Premium тип протоколу вибирається MODBUS30.

**Д.3.1.8. Конфігурування Variable Tag.** Для ідентифікації джерела даних для VariableTag при виборі типу протоколу MODBUS30 в поле Address вписується адреса в форматі PL7/UNITY (наприклад %MW100).

### Додаток 3.2. Коробка підключення TSX P ACC01.

Реалізація з'єднання контролерів TSX Micro/ TSX Premium/ TSX Twido по мережі MODBUS RTU/ASCII (крім TSX Premium) або UNITELWAY (крім TSX Twido) з використанням термінальних портів можливе з використанням спеціальної коробки TSX P ACC01. Ця коробка гальванічно розв'язує трансивер RS-485 контролера від загальної шини, що дозволяє безпечно реалізовувати шинні топології на відстані до 1000 м (замість 10 м при відсутності розв'язки). Крім того дана коробка дозволяє підмикати адаптери перетворювачі TSX PCX 1031 до загальної шини.

Коробка TSX P ACC01 має бути підключена до джерела живлення 5 В, яке присутнє на вбудованому термінальному порті ПЛК. Порт TER на коробці відповідає порту Port1 (для Twido) або TER (для Premium) на контролері, який живить дану коробку. Винятком є відсутність сигнальної лінії для управління станом трансмітера на перетворювачі з боку RS-485. Тобто перетворювач, який підключається до TSX P ACC01 (наприклад TSX PCX 1031) повинен управлятися зі сторони RS-232 сигналом RTS.

На рис.3.6 показане внутрішнє наповнення коробки: клеми для підключення шинного кабелю, перемикач вибору режиму контролера (S1), перемикач термінатору лінії (S2).

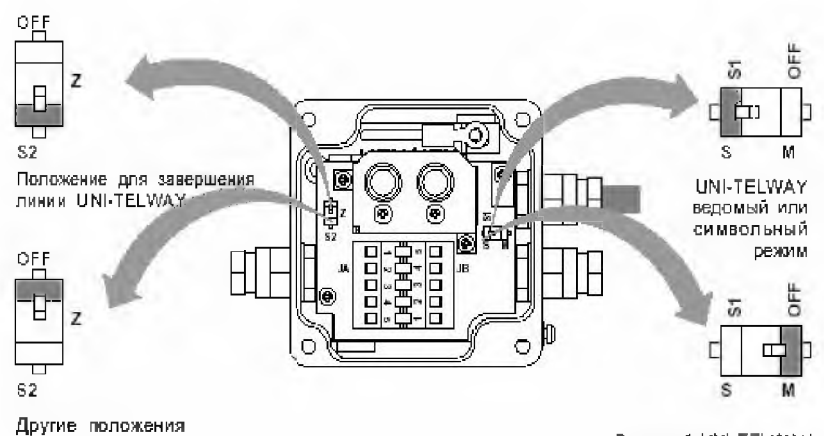


Рис.3.6.

S1 задає операційний режим роботи TER-порту, аналогічно перемикачу TER/OTHER стану перетворювача в режимі точка-точка (див. додаток 2.3).

Перемикачем S2 виставляється в кінцевих, тобто термінальних коробках. На лабораторній установці перемикачі на всіх коробках S1=S, а S2=OFF (окрім КП1). На рис.3.6 показані S2=ON в коробках КП1 та КП3, але в лабораторії за коробкою КП3 вправо йдуть інші коробки відгалуження, тому вона не являється термінальною.

### **Додаток 3.3. Особливості роботи драйверів при реалізації запитів в MODBUS RTU.**

Сучасні SCADA при створенні змінних намагаються оптимізувати мережний трафік між базою даних реального часу та ПЛК. Необхідність такої оптимізації розглянемо на прикладі.

Читання змінних %MW5, %MW10 та %MW15 можна реалізувати трьома окремими запитами на читання однієї змінної, або одним запитом на читання 11 змінних, оскільки MODBUS дозволяє робити запити як на одиночне так і на групове читання та запис. При груповому читанні, 8 змінних (16 байт) є лишніми, однак сумарний час транзакцій буде менший порівняно з одиночним. Тому доцільно змінні в пам'яті ПЛК розміщувати по порядку, щоб організувати їх групове читання. Не зважаючи на "дірки" в просторі змінних, драйвер SCADA, як правило, все одно зчитує змінні груповими запитами. Тим не менше, загальноприйнятою мірою для зменшення трафіку та мінімального часу обміну між пристроями є послідовне розміщення мережних змінних в пам'яті контролерів.

## ЛАБОРАТОРНА РОБОТА № 4 (модуль 2).

### Використання шини MODBUS RTU та MODBUS TCP/IP для зв'язку між ПЛК. Перевірка розрахованого часу транзакції.

#### ЧАСТИНА 1.

**Тривалість:** 1 акад. година (0.5 пари).

**Мета:** Навчитись реалізовувати комунікаційний обмін між ПЛК з використанням комунікаційних функцій; дослідна перевірка розрахованого часу транзакції по мережі MODBUS RTU.

#### Лабораторна установка.

Лабораторна робота виконується на робочих місцях 4 та 7. Одна бригада – одне робоче місце. Інші дві бригади в цей час виконують частину 2.

**Апаратне забезпечення.** Лабораторна установка складається з: контролерів M340 та Twido, які з'єднані між собою по MODBUS RTU на базі інтерфейсу RS-485; комп'ютера, підключеного до M340 через USB-інтерфейс (Рис.4.1).

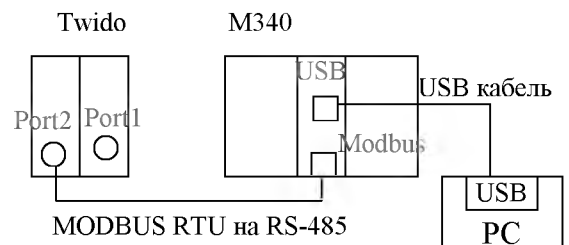


Рис.4.1.

Підключення M340 до MODBUS RTU проводиться через інтегрований в CPU P341000 послідовний порт з позначенням Modbus. Даний порт реалізує інтерфейси RS-232/RS-485 з підтримкою протоколів MODBUS RTU/ASCII Master/Slave та символного режиму. Підключення до ліній інтерфейсу проводиться через RJ-45 з'єднувач.

Підключення Twido до MODBUS RTU проводиться через адаптер TWD NAC 485T, який включений в модуль TWD XCP ODM. Цей адаптер являється другим комунікаційним портом контролера (Port2) і реалізує інтерфейс RS-485 з підтримкою протоколів MODBUS RTU/ASCII Master/Slave, Remote Link та символного режиму. Підключення до ліній інтерфейсу проводиться через клемну колодку на адаптері.

Фізичне з'єднання портів контролерів M340 та Twido проводиться через клемну колодку.

**УВАГА!** Порти ПЛК M340 та Twido не ізолювані, тому для їх безпечного з'єднання підключення необхідно проводити при вимкненому живленні одного з ПЛК, з дозволу викладача.

**Програмне забезпечення.** На комп'ютері використовується середовище UNITY PRO для програмування M340. З'єднання ПК та M340 проводиться через драйвер USB. У Twido записана програма, яка змінює значення змінних відповідно до завдання.

**Загальна постановка задачі.** Цілі роботи:

- 1) реалізувати поставлену задачу, оформлену у вигляді схеми інформаційних потоків (рис.4.2); ця задача передбачає обмін даними між M340 та Twido;
- 2) перевірити розрахований час транзакції з отриманими в експерименті.

**Деталізований опис постановки задачі.** Реалізація клієнтських запитів в M340 для читання вихідних/внутрішніх регістрів по протоколу MODBUS проводиться через функцію READ\_VAR, яка описана в додатку 4.1. Постановка задачі передбачає виконання запиту по ініціації користувача. Необхідно створити таку програму, яка б розраховувала час між моментом ініціації клієнтського запиту та повернення відповіді. Загальна методика розрахунку такого часу наведена в додатку 4.2.

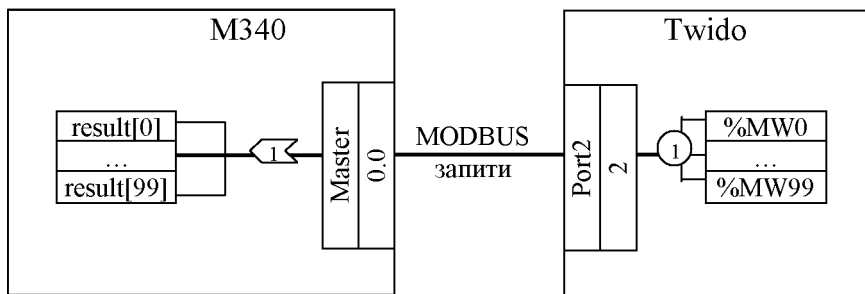


Рис.4.2. Схема інформаційних потоків до частини 1 лабораторної роботи №4.

### Послідовність виконання роботи.

- 1) Зібрати лабораторну установку.

**УВАГА! Порти ПЛК M340 та Twido не ізолювані, тому для їх безпечного з'єднання підключення необхідно проводити при вимкненому живленні одного з ПЛК, з дозволу викладача.**

- 2) Завантажити UNITY PRO. Створити новий проект: при виборі контролера виставити опцію "Show All Version"; вибрати BMX P34 1000 (OS.2.0).
- 3) Створити конфігурацію для послідовного порту (в Project Browser в розділі Configuration -> BMX P34 1000 -> Serial Port):
  - Function=Modbus link;
  - Type = Master;
  - Transmission Speed = 19200;
  - DATA=RTU;
  - Stop=1 bit;
  - Parity=Even;
  - Physical=RS485;
- 4) Створити або імпортувати секцію з програмою та змінними (див.рис.4.6 та рис.4.7).
- 5) Для імпорту виділити в Project Browser в розділі Program->Tasks->Mast->Sections-> контекстне\_меню->Import..., вказати файл PKIS4\_RTU.XST. Після імпорту необхідно перевірити створення змінних в розділі Variables & FB instances згідно з рис.4.7. Необхідно також виставити опції: **Tools->Project Settings -> Variables -> Directly represented array variables = max; Tools->Project Settings -> Variables -> Allow dynamic arrays (ANY\_ARRAY\_XXX) = max.**
- 6) Скомпілювати проект (Build->Rebuild All Project). При виникненні помилок повторити правильність виконання пунктів 2-4.
- 7) Завантажити проект в ПЛК M340 та запустити контролер на виконання:
  - в меню PLC виставити Standard Mode;
  - в меню PLC->Set Address вибрати: Media=USB; Address=SYS

- перевірити зв'язок кнопкою Test Connection, якщо зв'язок відсутній – перевірити з'єднання;
  - провести з'єднання UNITY PRO з M340: PLC->Connect;
  - завантажити проект в ПЛК: PLC->Transfer Project to PLC;
  - запустити контролер на виконання (PLC->RUN)
- 8) Створити таблицю анімацій відповідно рис.4.8: виділити всю секцію програми і в контекстному меню вибрати Initialize Animation Table.
  - 9) Зчитати 100 змінних з ПЛК Twido (%MW0-%MW99) шляхом зміни змінної send\_cmd в 1. Результат роботи комунікаційної функції перевірити в змінних para, а зчитані змінні в масиві result. При позитивному результаті (відсутність помилок в para) перейти до наступних пунктів.
  - 10) Занотувати значення Tm\_c1, Tm\_c2, Tm\_c3, Tsl\_c1 та trans\_time в таблицю формату Таб.4.1.
  - 11) Повторити пункти 8-9 ще 9 разів.
  - 12) Змінити кількість регістрів читання на 50 (%MW0-%MW49). Повторити пункти 8-10.
  - 13) Розрахувати Tsend та Trecv відповідно до формату запитів MODBUS та швидкісних характеристик мережі, занотувати результати в поле "кільк.регістрів/Tsend/Trecv" для 100 та 50 регістрів. Вибрати зі створеної таблиці записи з найбільшим, найменшим та середнім (орієнтовно) значеннями trans\_time для 100 та 50 регістрів (всього 6 записів). Розрахувати для цих записів Trans\_time<sub>min</sub> та Trans\_time<sub>max</sub> по формулам відповідно (4.1) та (4.2) та занотувати у відповідні поля, при умові що Tsl\_c2=Tsl\_c1.

### Оформлення роботи.

До захисту готується таблиця типу Таб.4.1.

Таб.4.1.

№ досліду	кільк. регістри в / Tsend/ Trecv	Tm_c1 мс	Tsl_c1 мс	Tm_c2 мс	Tm_c3 мс	trans_time мс	Trans_time <sub>min</sub> мс	Trans_time <sub>max</sub> мс
1	100/							
...	Tsend/							
10	Trecv							
11	50/							
...	Tsend/							
20	Trecv							

### Перевірка виконання роботи та питання до захисту.

Викладачем вибірково перевіряється виконання всіх пунктів роботи та занотовані результати. Кожен результат студент повинен пояснити. У випадку виникнення помилок або запитань щодо проведення певного пункту, цей пункт необхідно буде повторити.

Перед захистом роботи необхідно пояснити постановку задачі та кінцеві результати.

1. Поясніть методику розрахунку часу транзакції, яка використовується в даній частині лабораторної роботи. Від чого залежить час транзакції?

2. Яким чином в програмі для лабораторної роботи визначається час циклів Ведучого та Веденого?
3. Яким чином реалізуються комунікаційні функції читання та запису внутрішніх регістрів по протоколу MODBUS в контролерах Premium та M340? Поясніть призначення кожного параметру в цих функціях.
4. Яким чином контролюється результат виконання комунікаційних функцій? Що таке біт активності?
5. Прокоментуйте результати з таблиці 4.1? Поясніть чому вони відрізняються? Що змінилося при зменшенні кількості регістрів, які читаються, і чому?

## ЧАСТИНА 2.

**Тривалість:** 1 акад. година (0.5 пари).

**Мета:** Навчитись реалізовувати комунікаційний обмін між ПЛК з використанням комунікаційних функцій; дослідна перевірка розрахованого часу транзакції по мережі MODBUS/TCP.

### Лабораторна установка.

Лабораторна робота виконується на робочих місцях 3 та 8. Одна бригада – одне робоче місце. Інші дві бригади в цей час виконують частину 1.

**Апаратне забезпечення.** Лабораторна установка складається з: контролерів Premium (Premium1 та Premium2), які з'єднані між собою по Ethernet з використанням MODBUS/TCP; комп'ютера, підключеного до Premium1 через адаптер-перетворювач TSX PCX1031 та протоколу UNI-TELWAY (Рис.4.3).

Підключення Premium до Ethernet проводиться через інтегрований Ethernet порт з позначенням ETU PORT Ethernet 10/100. Даний порт реалізує мережу EthernetII/802.3 з підтримкою багатьох протоколів, зокрема MODBUS/TCP Клієнт та Сервер (MODBUS/TCP Сервер є інтегрованим на рівні операційної системи). Фізичне з'єднання між контролерами відбувається через Ethernet Switch.

**Програмне забезпечення.** На комп'ютері використовується середовище UNITY PRO для програмування Premium1. З'єднання ПК та Premium1 проводиться через драйвер UNITELWAY. У Premium2 записана програма, яка змінює значення змінних відповідно до завдання.

**Загальна постановка задачі.** Цілі роботи:

- 1) реалізувати поставлену задачу, оформлену у вигляді схеми інформаційних потоків (рис.4.4); ця задача передбачає обмін даними між Premium1 та Premium2;
- 2) перевірити розрахований час транзакції з отриманими в експерименті.

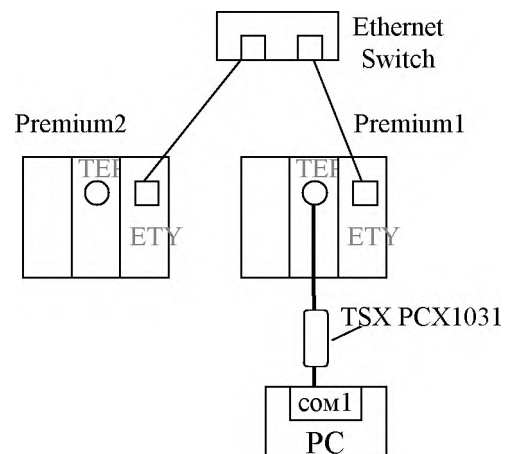
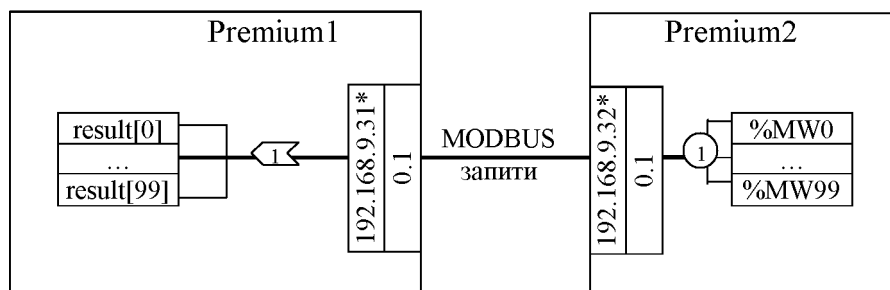


Рис.4.3.

**Деталізований опис постановки задачі.** MODBUS/TCP Сервер реалізований в Premium на рівні операційної системи (достатньо тільки сконфігурувати порт Ethernet). Реалізація клієнтських запитів в Premium для читання вихідних/внутрішніх регістрів по протоколу MODBUS проводиться через функцію READ\_VAR, яка описана в додатку 4.1. Постановка задачі передбачає виконання запиту по ініціалії користувача. Необхідно створити таку програму, яка б розраховувала час між моментом ініціалії клієнтського запиту та повернення відповіді. Загальна методика розрахунку такого часу наведена в додатку 4.3.



- \* - для робочого місця 3: IP Premium1=192.168.9.31; IP Premium2=192.168.9.32
- для робочого місця 8: IP Premium1=192.168.9.42; IP Premium2=192.168.9.41

Рис.4.4. Схема інформаційних потоків до частини 2 лабораторної роботи №4.

### Послідовність виконання роботи.

1. Зібрати лабораторну установку.
2. Завантажити UNITY PRO. Створити новий проект: при виборі контролера виставити опцію "Show All Version"; вибрати TSX P57 2634M (OS.2.0).
3. Створити конфігурацію для мережі Ethernet:
  - 3.1. Створити нову мережу: в Project Browser в розділі Communication->Networks -> контекстне меню->New Network-> вибрати Ethernet.
  - 3.2. Для створеної мережі налаштувати:
    - Model Family=TCP/IP Regular Connection.
    - IP адресу Premium1 відповідно до робочого місця (вкладка IP Configuration), маска підмережі 255.255.255.0;
    - відповідність логічних адрес типу Network.Station реальним фізичним адресам IP (див.Д.4.1.5);
  - 3.3. Прив'язати створену конфігурацію до реального порту: в Project Browser в розділі Configuration -> TSX RKY 6EX -> TSX ETY Port
    - Channel0->Function = ETH TCP/IP
    - Channel0->Net Link = "назва створеної мережі в п.3.1." (наприклад Ethernet\_1)
  - 3.4. Закінчити конфігурацію: меню Edit->Validate
4. Створити або імпортувати секцію з програмою та змінними (див.рис.4.10 та рис.4.7).

Для імпорту виділити в Project Browser в розділі Program->Tasks->Mast->Sections-> контекстне\_меню->Import..., вказати файл PKIS4\_TCP.XST. Після імпорту необхідно перевірити створення змінних в розділі Variables & FB instances згідно з рис.4.7. Необхідно також виставити опції: **Tools->Project Settings -> Variables ->**

**Directly represented array variables = max; Tools->Project Settings -> Variables -> Allow dynamic arrays (ANY\_ARRAY\_XXX) = max.**

5. Скомпілювати проект (Build->Rebuild All Project). При виникненні помилок повторити правильність виконання пунктів 2-4.
6. Завантажити проект в ПЛК Premium1 та запустити контролер на виконання:
  - в меню PLC виставити Standard Mode;
  - в меню PLC->Set Address вибрати: Media= UNTLW01; Address=SYS
  - перевірити зв'язок кнопкою Test Connection, якщо зв'язок відсутній – перевірити з'єднання;
  - провести з'єднання UNITY PRO з Premium1: PLC->Connect;
  - завантажити проект в ПЛК: PLC->Transfer Project to PLC;
  - запустити контролер на виконання (PLC->RUN)
7. Створити таблицю анімацій відповідно до рис.4.8: виділити всю секцію програми і в контекстному меню вибрати Initialize Animation Table.
8. Зчитати 100 змінних з ПЛК Premium2 (%MW0-%MW99) шляхом зміни змінної send\_cmd в 1. Результат роботи комунікаційної функції перевірити в змінних para, а зчитані змінні в масиві result. При позитивному результаті (відсутність помилок в para) перейти до наступних пунктів.
9. Занотувати значення Tm\_c1, Tm\_c2, Tm\_c3, Tsl\_c1 та trans\_time в таблицю формату Таб.4.2.
10. Повторити пункти 8-9 ще 9 разів.
11. Змінити кількість регістрів читання на 50 (%MW0-%MW49). Повторити пункти 8-10.
12. Вибрати зі створеної таблиці записи з найбільшим, найменшим та середнім (орієнтовно) значеннями trans\_time для 100 та 50 регістрів (всього 6 записів). Розрахувати для цих записів Trans\_time<sub>min</sub> та Trans\_time<sub>max</sub> по формулам відповідно (4.4) та (4.5) та занотувати у відповідні поля, при умові що:
  - для Trans\_time<sub>min</sub> - NAT1=NAT2=1 мс;
  - для Trans\_time<sub>max</sub> - NAT1=NAT2=10 мс,
  - Tsl\_c2=Tsl\_c3.

### Оформлення роботи.

До захисту готується таблиця типу Таб.4.2.

Таб.4.2.

№ досліду	кількість регістрів	Tm_c1 мс	Tsl_c1 мс	Tm_c2 мс	Tm_c3 мс	trans_time мс	Trans_time <sub>min</sub> мс	Trans_time <sub>max</sub> мс
1	100							
...								
10								
11	50							
...								
20								



## Перевірка виконання роботи та питання до захисту.

Викладачем вибірково перевіряється виконання всіх пунктів роботи та занотовані результати. Кожен результат студент повинен пояснити. У випадку виникнення помилок або запитань щодо проведення певного пункту, цей пункт необхідно буде повторити.

Перед захистом роботи необхідно пояснити постановку задачі та кінцеві результати.

1. Поясніть методику розрахунку часу транзакції, яка використовується в даній частині лабораторної роботи. Яким чином в програмі для лабораторної роботи визначається час циклів Клієнту та Серверу?
2. Яким чином реалізуються комунікаційні функції читання та запису внутрішніх реєстрів по протоколу MODBUS в контролерах Premium та M340? Поясніть призначення кожного параметру в цих функціях.
3. Яким чином контролюється результат виконання комунікаційних функцій? Що таке біт активності?
4. Прокоментуйте результати з таблиці 4.2? Поясніть чому вони відрізняються? Що змінилося при зменшенні кількості реєстрів, які читаються, і чому?

### Додаток 4.1. Комунікаційні функції UNITY для читання/запису внутрішніх/вихідних реєстрів.

**Д.4.1.1. Використання комунікаційних функцій в UNITY.** Для роботи з комунікаціями в контролерах Quantum, Premium та M340, які програмуються в середовищі UNITY PRO використовуються комунікаційні функції. Для зчитування внутрішніх/вихідних реєстрів MODBUS в M340 та Premium використовується функція READ\_VAR, для запису - WRITE\_VAR.

**Д.4.1.2. Функція READ\_VAR.** Для зчитування внутрішніх/вихідних реєстрів або бітів MODBUS в M340 та Premium використовується функція READ\_VAR. Синтаксис функції в ST:

READ_VAR(Address, Object_Type, First_Object, Object_Number, Management_Param, Receiving_Array);		
Address	ARRAY [0.. 5] OF <a href="#">INT</a> для Premium  ARRAY [0.. 7] OF <a href="#">INT</a> для M340	Адреса вузла отримувача запиту; для отримання цього масиву в M340 використовується функція ADDM (див.Д.4.1.4) а в Premium функція ADDR (див.Д.4.1.5).
Object Type	<a href="#">STRING</a>	Тип об'єкта зчитування: '%M' або '%MW'
First Object	<a href="#">DINT</a>	Адреса початкового об'єкту
Object Number	<a href="#">INT</a>	Кількість об'єктів
Management_Param	ARRAY [0.. 3] OF <a href="#">INT</a>	Таблиця управління обміном (див.Д.4.1.6)
Receiving_Array	ARRAY [n.. m] OF <a href="#">INT</a>	Масив, який вміщує значення прочитаних змінних

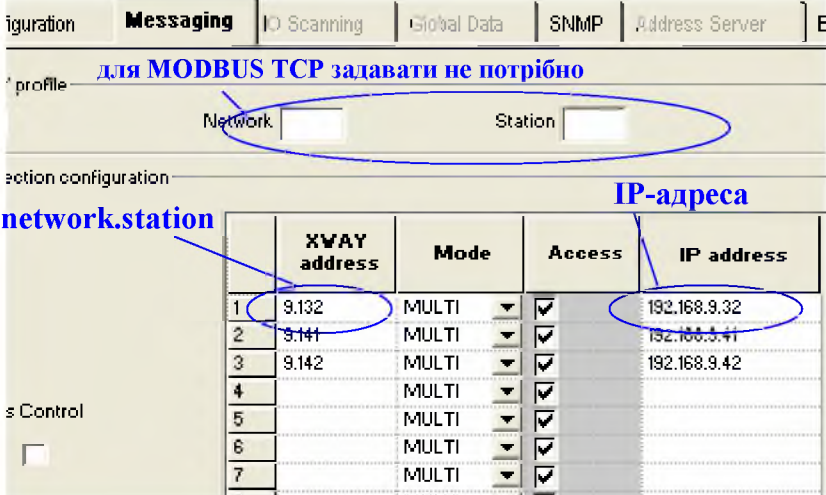
**Д.4.1.3. Функція WRITE\_VAR.** Для запису внутрішніх/вихідних регістрів або бітів MODBUS в M340 та Premium використовується функція WRITE\_VAR. Синтаксис функції в ST:

<b>WRITE_VAR(Address, Object_Type, First_Object, Object_Number, Data_to_Write, Management_Param);</b>		
Address	ARRAY [0.. 5] OF <a href="#">INT</a> для Premium  ARRAY [0.. 7] OF <a href="#">INT</a> для M340	Адреса вузла отримувача запиту; для отримання цього масиву в M340 використовується функція ADDM (див.Д.4.1.4) а в Premium функція ADDR (див.Д.4.1.5).
Object_Type	<a href="#">STRING</a>	Тип об'єкта запису: '%M' або '%MW'
First_Object	<a href="#">DINT</a>	Адреса початкового об'єкту
Object_Number	<a href="#">INT</a>	Кількість об'єктів
Data_to_Write	ARRAY [n.. m] OF <a href="#">INT</a>	Масив, який вміщує значення даних, які треба записати
Management_Param	ARRAY [0.. 3] OF <a href="#">INT</a>	Таблиця управління обміном (див.Д.4.1.6)

**Д.4.1.4. Функція ADDM (для M340).** Функція ADDM призначена для перетворення символічної строки в адресу (у вигляді масиву), яка використовується у ряді комунікаційних функцій контролерів M340, зокрема READ\_VAR та WRITE\_VAR.

<b>Addr_Array := ADDM(Addr String);</b>		
Addr_String	<a href="#">STRING</a>	Адреса вузла отримувача в символічному форматі: для протоколу MODBUS:  <b>'r.m.c.e'</b>  де: <ul style="list-style-type: none"> <li>– r: номер шасі з комунікаційним модулем (для інтегрованого порту Modbus завжди = 0)</li> <li>– m: номер комунікаційного модуля (для інтегрованого порту Modbus завжди = 0)</li> <li>– c: номер каналу (для інтегрованого порту Modbus завжди = 0)</li> <li>– e: адреса Веденого (від 1-го до 247)</li> </ul>
Addr_Array	ARRAY [0.. 7] OF <a href="#">INT</a>	Адреса вузла отримувача запиту у вигляді масиву;

**Д.4.1.5. Функція ADDR (для Premium).** Функція ADDR призначена для перетворення символічної строки в адресу (у вигляді масиву), яка використовується у ряді комунікаційних функцій контролерів Premium, зокрема READ\_VAR та WRITE\_VAR.

<b>Addr_Array := ADDR(Addr_String);</b>		
Addr_String	<a href="#">STRING</a>	<p>Адреса вузла отримувача в символічному форматі: для протоколу MODBUS:</p> <p style="text-align: center;"><b>'{network.station}SYS'</b></p> <p>де:</p> <ul style="list-style-type: none"> <li>– network: номер мережі</li> <li>– station: номер станції</li> <li>– SYS: символічна константа (системний Сервер)</li> </ul> <p>Специфіка синтаксису визначена універсальністю комунікаційних функцій Premium. Зв'язок логічної адреси типу "network.station" з дійсною IP-адресою отримувача налаштовується при створенні в UNITY PRO комунікації Ethernet, в таблиці на вкладці Messaging. Адреси Station обов'язково повинні знаходитись в межах 100-164.</p> 
Addr_Array	ARRAY [0.. 5] OF <a href="#">INT</a>	Адреса вузла отримувача запиту у вигляді масиву;

**Д.4.1.6. Таблиця управління обміном (Management\_Parameter).** Таблиця управління обміном (іншими словами комунікаційні параметри обміну) потрібні для управління та контролю за роботою комунікаційної функції. Ця таблиця включає 4-ри слова:

Номер слова в таблиці	Старший байт слова	Молодший байт слова
<b>1</b>	<b>номер обміну (Exchange number)</b>	0-й біт – <b>біт активності (Activity Bit)</b> , переводиться системою в 1 при початку обробки, в 0 – при поверненні результату
<b>2</b>	<b>operation report</b> 00 – немає помилок при обробці; FF – повідомлення помилки;	<b>communication report:</b> 00 – немає помилок при обробці; xx – номер помилки (додаткова інформація в довідковій системі UNITY PRO);
<b>3</b>	<b>Time-out:</b> дозволений тайм-аут для очікування відповіді, задається в 100 мс (10 = 1с), 0 – нескінченне очікування	
<b>4</b>	<b>Length:</b> для READ VAR – кількість отриманих байтів	

## Додаток 4.2. Методика та програма розрахунку часу транзакції в мережі MODBUS RTU з контролерами M340 та Twido.

**Д.4.2.1. Методика розрахунку.** Методика розрахунку часу транзакції, яка використовується в даній роботі аналогічна як для інших систем, в яких обмін з мережею MODBUS RTU прив'язаний до циклу контролера, тобто обробка вхідних повідомлень проходить на початку циклу, а відправка вихідних – в кінці (рис.4.5). Час транзакції ( $Trans\_time$ ) залежить від: тривалості циклів Ведучого ( $Tm\_c1$ ,  $Tm\_c2$ ,  $Tm\_c3$ ); тривалості циклів Веденого ( $Tsl\_c1$ ,  $Tsl\_c2$ ); часу передачі по мережі запиту ( $Tsend$ ) та відповіді ( $Trecv$ ); часу надходження запиту до Веденого та відповіді до Ведучого, який буде впливати на  $Tsl\_dev$  та  $Tm\_dev$ . Таким чином, загальний час транзакції при тих самі  $Tm\_c1$ ,  $Tm\_c2$ ,  $Tm\_c3$ ,  $Tsl\_c1$  та  $Tsl\_c2$ , але при різних  $Tsl\_dev$  та  $Tm\_dev$  може приймати значення:

- 1) мінімальне – при  $Tsl\_dev=Tsl\_c1$  та  $Tm\_dev=Tm\_c2$ , тобто кадри приходять перед зчитуванням вхідів циклів  $sl\_c2$  та  $m\_c3$ :

$$Trans\_time_{min}=Tm\_c1 + Tsend + Tsl\_c2 + Trecv + Tm\_c3; \quad (4.1)$$

- 2) максимальне – при  $Tsl\_dev=0$  та  $Tm\_dev=0$ , тобто кадри приходять відразу після зчитування вхідів  $sl\_c1$  та  $m\_c2$ :

$$Trans\_time_{max}=Tm\_c1 + Tsend + Tsl\_c1 + Tsl\_c2 + Trecv + Tm\_c2 + Tm\_c3; \quad (4.2)$$

- 3) між максимальним та мінімальним:

$$Trans\_time_{min} < Trans\_time < Trans\_time_{max} \quad (4.3)$$

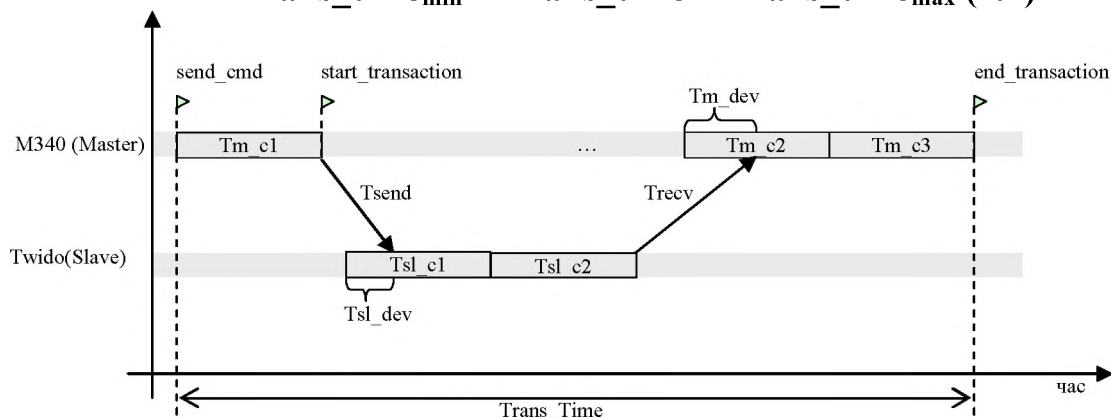


Рис.4.5. Часова діаграма до розрахунку часу транзакції.

**Д.4.2.2. Варіант програми для відправки запиту на читання, з фіксацією часу транзакції.** На рис. 4.6 представлений лістинг програми для M340, яка відправляє запит на читання 100 регістрів (%MW0-%MW99) у Веденого з адресою 2. Виклик комунікаційної функції, що формує даний запит (READ\_VAR) проводиться шляхом зведення змінної *send\_cmd* в логічну "1", скидання значення даної змінної в 0 проводиться програмно.

Фіксація часу транзакції проводиться шляхом використання таймеру (Timer). Час початку передачі запиту позначається міткою *start\_transaction* а кінець транзакції - *end\_transaction* (див. рис.4.5). Загальний час транзакції розраховується як час між командою *send\_cmd* та міткою *end\_transaction* (рис.4.5). Останнє значення часу циклу Веденого записується в його 0-вий регістр (%MW0). Змінні які використовуються в програмі наведені на рис. 4.7.

```

(*таймер працює при команді на читання та протягом транзакції*)
Timer (IN := start_transaction or send_cmd,
      PT := t#100s);

(*про тривалість циклу в мс можна дізнатися в наступному циклі через системну змінну %SW30*)
if send_cmd and para[0].0 then (*на наступний цикл після відправки запиту*)
  Tm_c1:=%sw30;
  send_cmd:=false;
end_if;

if end_transaction then (*на наступний цикл після отримання запиту*)
  Tm_c3:=%sw30;
  end_transaction:=false;
end_if;

if send_cmd then (*відправляємо запит на читання*)
  READ_VAR (ADR := ADDR ('0.0.0.2'),(*адреса Веденого=2*)
           OBJ := '%MW', (*тип об'єкту*)
           NUM := 0,(*початковий номер*)
           NB := 100,(*кількість*)
           GEST := para,(*комунікаційні параметри*)
           RECP => result); (*масив для отриманих даних*)
  start_transaction:=true; (*мітка початку транзакції*)
end_if;

(*при запуску функції 0-й біт 0-го слова таблиці -para- виставляється в 1*)
(*при отриманні результату обробки функції 0-й біт 0-го слова таблиці -para- виставляється в 0*)
if start_transaction and not para[0].0 then
  end_transaction:=true; (*мітка закінчення транзакції*)
  trans_time:=Timer.et; (*при закінченні транзакції запам'ятовуємо час*)
  start_transaction:=false;
  Tm_c2:=%sw30; (*запам'ятовуємо попередній цикл*)
  Tsl_c1:=result[0]; (*в 0-му слові Веденої відправляє значення останнього циклу*)
end_if;

```

Рис.4.6. Лістинг програми для відправки запиту на читання 100 внутрішніх регістрів з фіксацією часу транзакції.

Name	Type	Address	Value	Comment
end_transaction	BOOL			мітка закінчення транзакції
para	ARRAY[0..3] OF INT			
para[0]	INT			лічильник запитів/біт активності
para[1]	INT			операційний звіт/комунікаційний звіт
para[2]	INT			тайм-аут
para[3]	INT			довжина
result	ARRAY[0..99] OF INT			отримані дані
send_CMD	BOOL			команда на відправку
start_transaction	BOOL			мітка початку транзакції
Tm_c1	INT			час циклу Ведучого при відправці запиту
Tm_c2	INT			час циклу Ведучого перед отриманням відповіді
Tm_c3	INT			час циклу Ведучого в якому він отримує відповідь
trans_time	TIME			час транзакції
Tsl_c1	INT			час циклу Веденого

Name	no.	Type
Timer		TON

Рис.4.7. Опис змінних для програми .

Для перевірки роботи програми слід створити таблицю анімацій, як на рис.4.8. Для можливості зміни значення змінних в таблиці анімацій необхідно натиснути кнопку Modification. Зведення змінної send\_cmd в "1", приведе до виклику функції READ\_VAR. Результат обробки функції можна проконтролювати через змінну para[1] (див. Д.4.1.6).

Name	Value	Type
para		ARRAY[0..3] OF INT
para[0]	16#1400	INT
para[1]	16#0000	INT
para[2]	0	INT
para[3]	100	INT
result		ARRAY[0..99] OF I...
send_CMD	0	BOOL
Tm_c1	2	INT
Tm_c2	1	INT
Tm_c3	1	INT
Tsl_c1	19	INT
trans_time	106ms	TIME

Рис.4.8. Таблиця анімацій.

- УВАГА!** Для роботи програми в UNITY PRO 4 необхідно виставити наступні опції:
- Tools->Project Settings -> Variables -> Directly represented array variables = так
  - Tools->Project Settings -> Variables -> Allow dynamic arrays (ANY\_ARRAY\_XXX) = так

### Додаток 4.3. Методика та програма розрахунку часу транзакції в мережі MODBUS/TCP з контролерами Premium.

**Д.4.3.1. Методика розрахунку.** Методика розрахунку часу транзакції, яка використовується в даній частині роботи аналогічна як для інших систем, в яких обмін з мережею MODBUS TCP прив'язаний до циклу контролера, тобто обробка вхідних повідомлень проходить на початку циклу, а відправка вихідних – в кінці (рис.4.6). Час транзакції (Trans\_time) залежить від: тривалості циклів Клієнта (Tm\_c1, Tm\_c2, Tm\_c3); тривалості циклів Серверу (Tsl\_c1, Tsl\_c2); часу доступу комунікаційного модуля до мережі (NAT, для інтегрованого ЕТУ порту Premium NAT<10мс); часу надходження запиту до Серверу та відповіді до Клієнта, який буде впливати на Tsl\_dev та Tm\_dev. Таким чином, загальний час транзакції при тих самі Tm\_c1, Tm\_c2, Tm\_c3, Tsl\_c1 та Tsl\_c2, але при різних Tsl\_dev та Tm\_dev може приймати значення:

- 1) мінімальне – при Tsl\_dev=Tsl\_c1 та Tm\_dev=Tm\_c2, тобто пакети приходять перед зчитуванням входів циклів sl\_c2 та m\_c3:

$$\mathbf{Trans\_time_{min}=Tm\_c1 + NAT1 + Tsl\_c2 + NAT2+ Tm\_c3; (4.4)}$$

- 2) максимальне – при Tsl\_dev=0 та Tm\_dev=0, тобто пакети приходять відразу після зчитування входів sl\_c1 та m\_c2:

$$\mathbf{Trans\_time_{max}=Tm\_c1 + NAT1 + Tsl\_c1 + Tsl\_c2 + NAT2 + Tm\_c2 + Tm\_c3; (4.5)}$$

- 3) між максимальним та мінімальним:

$$\mathbf{Trans\_time_{min} < Trans\_time < Trans\_time_{max} (4.6)}$$

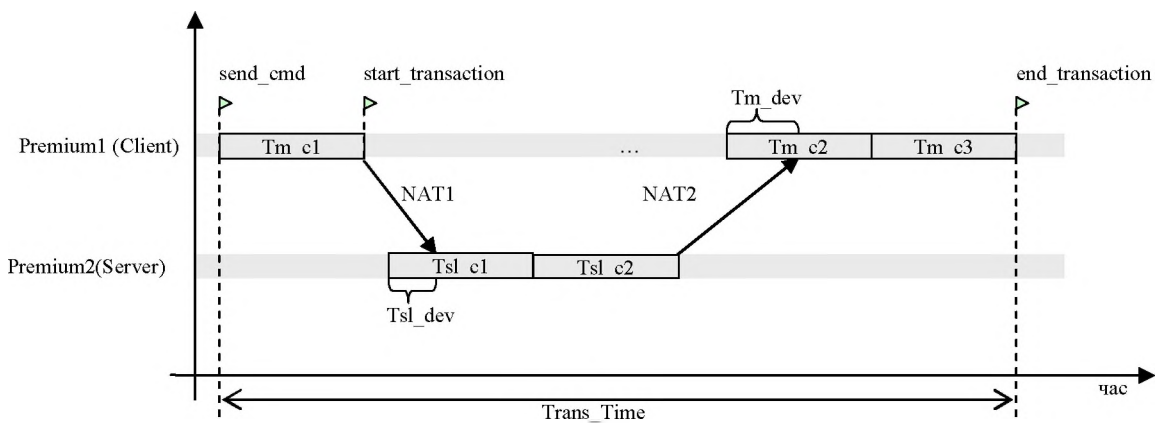


Рис.4.9. Часова діаграма до розрахунку часу транзакції.

**Д.4.2.2. Варіант програми для відправки запиту на читання, з фіксацією часу транзакції.** На рис. 4.10 представлений лістинг програми для Premium1, яка відправляє запит на читання 100 регістрів (%MW0-%MW99) у Клієнта з логічною адресою 9.132 (відповідне призначення адреси IP див. Д.4.1.5). Виклик комунікаційної функції, що формує даний запит (READ\_VAR) проводиться шляхом зведення змінної send\_cmd в логічну "1", скидання значення даної змінної в 0 проводиться автоматично.

Фіксація часу транзакції проводиться шляхом використання таймеру (Timer). Час початку передачі запиту позначається міткою start\_transaction а кінець транзакції - end\_transaction (див. рис.4.9). Загальний час транзакції розраховується як час між командою send\_cmd та міткою end\_transaction (рис.4.9). Останнє значення часу циклу Сервера записується в його 0-вий регістр (%MW0). Змінні які використовуються в програмі наведені на рис. 4.7.

Для перевірки роботи програми слід створити таблицю анімацій, як на рис.4.8. Для можливості зміни значення змінних в таблиці анімацій необхідно натиснути кнопку Modification. Зведення змінної send\_cmd в "1", приведе до виклику функції READ\_VAR. Результат обробки функції можна проконтролювати через змінну para[1] (див. Д.4.1.6).

- УВАГА!** Для роботи програми в UNITY PRO 4 необхідно виставити наступні опції:
- Tools->Project Settings -> Variables -> Directly represented array variables = так
  - Tools->Project Settings -> Variables -> Allow dynamic arrays (ANY\_ARRAY\_XXX) = так

```

(*таймер працює при команді на читання та протягом транзакції*)
Timer (IN := start_transaction or send_cmd,
       PT := t#100s100ms );

(*про тривалість циклу в мс можна дізнатися в наступному циклі через системну змінну %SW30*)
if send_cmd and para[0].0 then (*на наступний цикл після відправки запиту*)
    Tm_c1:=%sw30;
    send_cmd:=false;
end_if;

if end_transaction then (*на наступний цикл після отримання запиту*)
    Tm_c3:=%sw30;
    end_transaction:=false;
end_if;

if send_cmd then (*відправляємо запит на читання*)
    READ_VAR (ADR := ADDR ('9.132}sys'), (*логічна адреса вузла Сервера 9.132*)
             OBJ := '%MW', (*тип об'єкту*)
             NUM := 0, (*початковий номер*)
             NB := 100, (*кількість*)
             GEST := para, (*комунікаційні параметри*)
             RECP => result); (*масив для отриманих даних*)
    start_transaction:=true; (*мітка початку транзакції*)
end_if;

(*при запуску функції 0-й біт 0-го слова таблиці -para- виставляється в 1*)
(*при отриманні результату обробки функції 0-й біт 0-го слова таблиці -para- виставляється в 0*)
if start_transaction and not para[0].0 then
    end_transaction:=true; (*мітка закінчення транзакції*)
    trans_time:=Timer.et; (*при закінченні транзакції запам'ятовуємо час*)
    start_transaction:=false;
    Tm_c2:=%sw30; (*запам'ятовуємо попередній цикл*)
    Tsl_c1:=result[0]; (*в 0-му слові Сервер відправляє значення останнього циклу*)
end_if;

```

Рис.4.10. Лістинг програми для відправки запиту на читання 100 внутрішніх регістрів з фіксацією часу трнзакції.



**ЛАБОРАТОРНА РОБОТА № 6 (модуль 4).  
Використання технології OPC для зв'язку SCADA+PLC, SCADA+SCADA.**

**ЧАСТИНА 1.**

**Тривалість:** 2 акад. години (1 пара).

**Мета:** Навчитись користуватися технологією OPC для зв'язку між мікропроцесорними засобами автоматизації в гетерогенних системах (між пристроями різних виробників).

**Лабораторна установка.**

Лабораторна робота виконується на робочих місцях 2,3 (одна бригада), 4,5 (друга бригада), 6,7 (третья бригада) та 9,10 (четверта бригада). Лабораторна установка для робочих місць 2 та 3 зображена на рис.6.1.

**Апаратне забезпечення.** Лабораторна установка складається з двох пар:

ПЛК Twido – MODBUS RTU – ПК  
ПК між собою з'єднані за допомогою Ethernet з використанням комутатора (Switch).

**Програмне забезпечення.** На комп'ютері використовується SCADA-програма VijeoCitect, OPC Сервер OFS (OPC Factory Server, вир. Шнейдер Електрик) з вбудованими драйверами зв'язку MODBUS RTU Master, тестова програма OFS Client (вир. Шнейдер Електрик).

**Загальна постановка задачі.** Цілі роботи:

- за допомогою тестової програми OPC Клієнта перевірити основні механізми роботи з локальним та віддаленим OPC Сервером;
- реалізувати поставлену задачу, оформлену у вигляді схеми інформаційних потоків, яка передбачає обмін даними між Сервером вводу/виводу SCADA VijeoCitect та ПЛК та між серверами VijeoCitect з використанням технології OPC;

**Деталізований опис постановки задачі.** У лабораторній роботі необхідно реалізувати задачу, показану на моделі (рис.6.2) та схемі (рис.6.3) інформаційних потоків (рис.3.2). Система повинна забезпечити обмін даними між PLC і SCADA (потoki 1,2,4) а також між двома SCADA (потік 3). Параметри налаштувань програмно-технічних засобів відповідно до варіанту наведені в таб. 6.1.

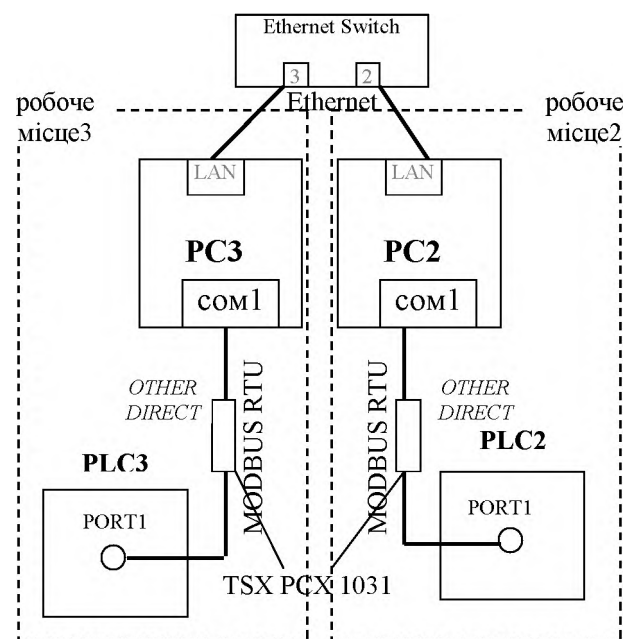


Рис.6.1.

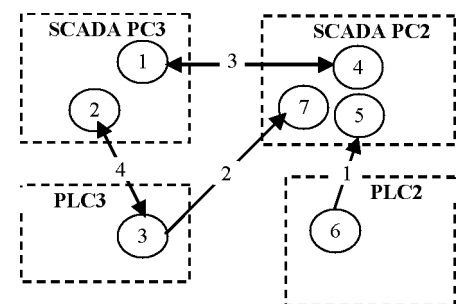


Рис.6.2.

Дана система являється гомогенною, оскільки використовуються продукти одного виробника (Шнейдер Електрик). У таких системах зв'язок між засобами може проходити з використанням закритих але більш ефективних технологій. SCADA програма VijeoCitect 7 базується на кластерній системі серверів, яка дозволяє будувати мультиклієнтські та мультисерверні системи з використанням власних протоколів обміну. У даній лабораторній роботі для цих цілей використовується відкрита технологія OPC DA2.0, яка не така ефективна в гомогенних системах ІШЕ, але являється стандартом технології інтеграції в гетерогенних системах (системах з використанням засобів різних виробників).

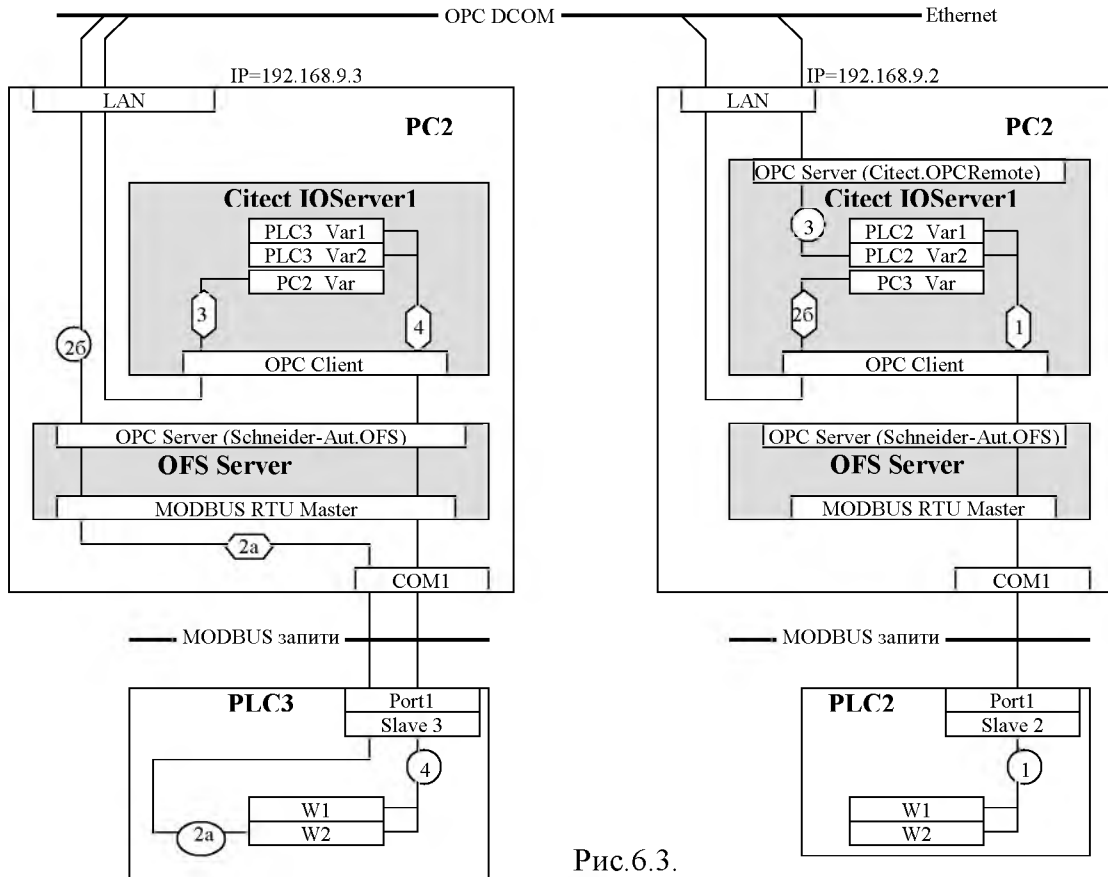


Рис.6.3.

Доступ до серверів вводу/виводу VijeoCitect надається через два типи OPC Серверів: локальний OPC з ProgID="Citect.OPC" та віддалений OPC з ProgID="Citect.OPCRemote". Таким чином при з'єднанні через OPC з Сервером VijeoCitect на тому самому ПК використовується ProgID="Citect.OPC", а з Сервером на іншому ПК - ProgID="Citect.OPCRemote".

Таб.6.1.

Параметр		робоче місце							
		2	3	4	5	7	8	9	10
потік І	Адреса Slave	2	3	4	5	7	8	9	10
	бітова швидкість	19200	19200	19200	19200	9600	9600	9600	9600
	паритет	парн	парн	парн	парн	непар	непар	непар	непар
	стоп біт	1	1	1	1	1	1	1	1
змінні PLC	W1 (%MWx)	123 <sub>10</sub>	68 <sub>10</sub>	34 <sub>10</sub>	176 <sub>10</sub>	34 <sub>10</sub>	11 <sub>10</sub>	67 <sub>10</sub>	95 <sub>10</sub>
	W2 (%MWx)	2 <sub>10</sub>	3 <sub>10</sub>	4 <sub>10</sub>	5 <sub>10</sub>	7 <sub>10</sub>	8 <sub>10</sub>	9 <sub>10</sub>	10 <sub>10</sub>
PC	значення x в IP адресі 192.168.9.x	2	3	4	5	12	13	14	15
	Ім'я вузла	Comp2	Comp3	Comp4	Comp5	Comp12	Comp13	Comp14	Comp15

## Послідовність виконання роботи.

1) Зібрати лабораторну установку.

**УВАГА! Port1 ПЛК Twido і СОМ-порт ПК не ізолювані, тому для безпечного з'єднання необхідно підключати перетворювачі при вимкнених ПЛК, з дозволу викладача.**

- 2) За допомогою OFSConfiguration сконфігурувати OFS для реалізації поставленої задачі. Вказати необхідність розширеного діагностичного режиму запуску сервера (OFS Server Settings-> Verbose).
- 3) Запустити на виконання програму OFS Клієнт, для перевірки локального серверу "Schneider-Aut.OFS". Після завантаження OFS Сервера відкрити в ньому вікно DiagnosticWindow. Проаналізувати останній запис у вікні, визначити Клієнта, що підключився останнім, та загальну кількість клієнтів. Результат занотуйте.
- 4) Повторити пункт 3 ще для одного OFS Клієнта. Після проведеного аналізу і запису результатів, закрити його.
- 5) Створити одну групу з періодом відновлення 1000 мс, а другу 2000 мс. У кожній групі створити по одному Item, який би відповідав відновленню змінної W1. Після створення проаналізувати, як відновлюються дані в різних групах. Проаналізувати Value, TimeStamp, Quality для всіх Item. Результат занотувати.
- 6) Вимкнути ПЛК, після чого проаналізувати Value, TimeStamp та Quality для всіх Item в OPC Group. Результат занотувати. Після запису результатів включити ПЛК.
- 7) Створити проект в VijeoCitect, з двома змінними, що відповідають W1 та W2 на PLC (поток 1 та 4). Забезпечити їх відображення та можливість зміни значення на мнемосхемі. Перевірити роботу системи в режимі виконання (ран-тайм) VijeoCitect.
- 8) Запустити на виконання програму OFS Клієнт, для перевірки локального OPC Серверу "Citect.OPC". Створити групу та додати із списку змінних змінну PLCx\_Var2. Задати змінній довільне значення в OFS Клієнті та проконтролювати її зміну в ран-тайм системі VijeoCitect. Після вдалого результату закрити OFS Клієнт.
- 9) Повторити пункти 3-5 для віддаленого OFS Серверу (інше робоче місце бригади).
- 10) Повторити пункт 8 для віддаленого серверу вводу/виводу VijeoCitect (інше робоче місце бригади).
- 11) Закінчити реалізацію поставленої задачі на рис.6.3.

## Оформлення роботи.

До захисту готовляться всі занотовані результати з поясненнями.

## Перевірка виконання роботи та питання до захисту.

Викладачем вибірково перевіряється виконання всіх пунктів роботи та занотовані результати. Кожен результат студент повинен пояснити. У випадку виникнення помилок або запитань щодо проведення певного пункту, цей пункт необхідно буде повторити.

1. Поясніть основні принципи функціонування технології OPC. На якій технології міжпрограмного обміну вона базується?

2. Назвіть специфікації стандарту OPC, які функціонують на сьогоднішній день, та їх призначення. Яка специфікація використовувалась в даній лабораторній роботі?
3. Яку послідовність необхідно зробити, щоб забезпечити доступ в програмі OPC-Клієнта до змінної в контролері через OPC-Сервер?
4. Як визначається джерело даних в OPC-Серверах та в OFS Сервері зокрема?
5. Як ідентифікуються дані OPC-Серверу програмою OPC-Клієнта? Яким чином формуються ItemID для OFS-Серверу.
6. Як OPC-Клієнт ідентифікує необхідний OPC-Сервер? Яка програма може називатись OPC-Клієнтом а яка OPC-Сервером?
7. Чи може одна і та сама програма бути одночасно і OPC-Клієнтом і OPC-Сервером? Назвіть які програми OPC-Клієнти та OPC-Сервери використовувались в лабораторній роботі.
8. Розкажіть як відбувається доступ до даних через об'єкти OPC-Item. Чи може надаватися доступ OPC-Клієнтам до одних і тих самих даних OPC-Сервера? Чи можуть декілька OPC-Клієнтів користуватися одним і тим самим OPC-Item?
9. Яка інформація про дані надається через об'єкт OPC-Item? Прокоментуйте це на прикладах пунктів 5 та 6 даної лабораторної роботи.
10. Розкажіть про призначення об'єктів OPC-Group. Які групові операції для об'єктів OPC-Item проводяться через OPC-Group? Прокоментуйте це на прикладі пункту 5.
11. Які способи читання з джерела даних доступні в OPC-DA 2.0? Як вони функціонують? Яке з них на Вашу думку найбільш підходить до опитування даних процесу в засобах SCADA/HMI?
12. Які способи запису даних доступні в OPC-DA 2.0? Як вони функціонують? Яке з них на Вашу думку найбільш підходить до супервізорної зміни даних в засобах SCADA/HMI?
13. Які вимоги ставляться до мережної системи для можливості зв'язку OPC-Клієнта з віддаленим OPC-Сервером?
14. Що додатково необхідно вказати в OPC-Клієнті для ідентифікації віддаленого OPC-Сервера, з яким необхідно з'єднатись? Прокоментуйте це на прикладі пунктів 9 та 10.
15. Розкажіть про область застосування OPC-технології на прикладі даної лабораторної роботи.

### **Додаток 6.1. Робота з OFS Client.**

**Д6.1.1. Загальні положення.** Програма OFS Client призначена для тестування роботи OPC DA Серверів, в тому числі OFS Server. Програма підтримує наступні функції:

1. роботу з локальними або віддаленими OPC Серверами, які підтримують специфікацію OPC DA;
2. визначення переліку підтримуваних інтерфейсів об'єкту OPCServer та OPCGroup ;
3. створення OPC групи з вказаними ім'ям, часом відновлення, зоною нечутливості;

4. доступ до властивостей OPC груп, зокрема управління активацією групи (періодичне читання), управління підпискою (асинхронні операції);
5. створення OPC Item та зміна їх властивостей;
6. проведення асинхронних та синхронних операцій над OPC Item;
7. доступ до часових відміток зчитування/запису всіх змінних

#### **Д6.1.2. З'єднання з OPC-Сервером.**

Після завантаження програми, необхідно вказати OPC Сервер, з яким повинен зв'язатися OFS Client. У списку ProgID OPC Серверів доступні тільки ті, які підтримують вказану версію специфікації над списком (при All – сервери підтримують 1.0 і 2.0 одночасно). Вибраний OPC Сервер висвітлюється в полі ProgID. У випадку з'єднання з віддаленим OPC сервером, в полі Node додатково вказується назва віддаленого ПК або його IP (наприклад [\\192.168.9.1](#) або \\COMP1).

#### **Д6.1.3. Робота з OPC-Group та OPC-Item.**

Меню Group->NewGroup – створення нової групи;

Меню Item->New – створення нового OPCItem; в списку вибирається конкретний ItemID;

При створенні, OPC Group по замовченню автоматично проводить періодичне читання з повідомленням. У цьому випадку, Item будуть відображати значення та властивості з періодом UpdateRate OPC групи.

В меню Group можна проводити групові синхронні/асинхронні операції над Item а також міняти властивості групи (Properties): активність(Active), режим повідомлення(Notifications) і т.д.

### **Додаток 6.2. Налаштування VijeoCitect на роботу з OPC серверами.**

Для конфігурування Citect/VijeoCitect як OPC-клієнтів, для кожного OPC-серверу створюється окремий Device, послідовність створення якого аналогічна як для інших типів пристроїв. Особливості конфігурування показані на рисунку.

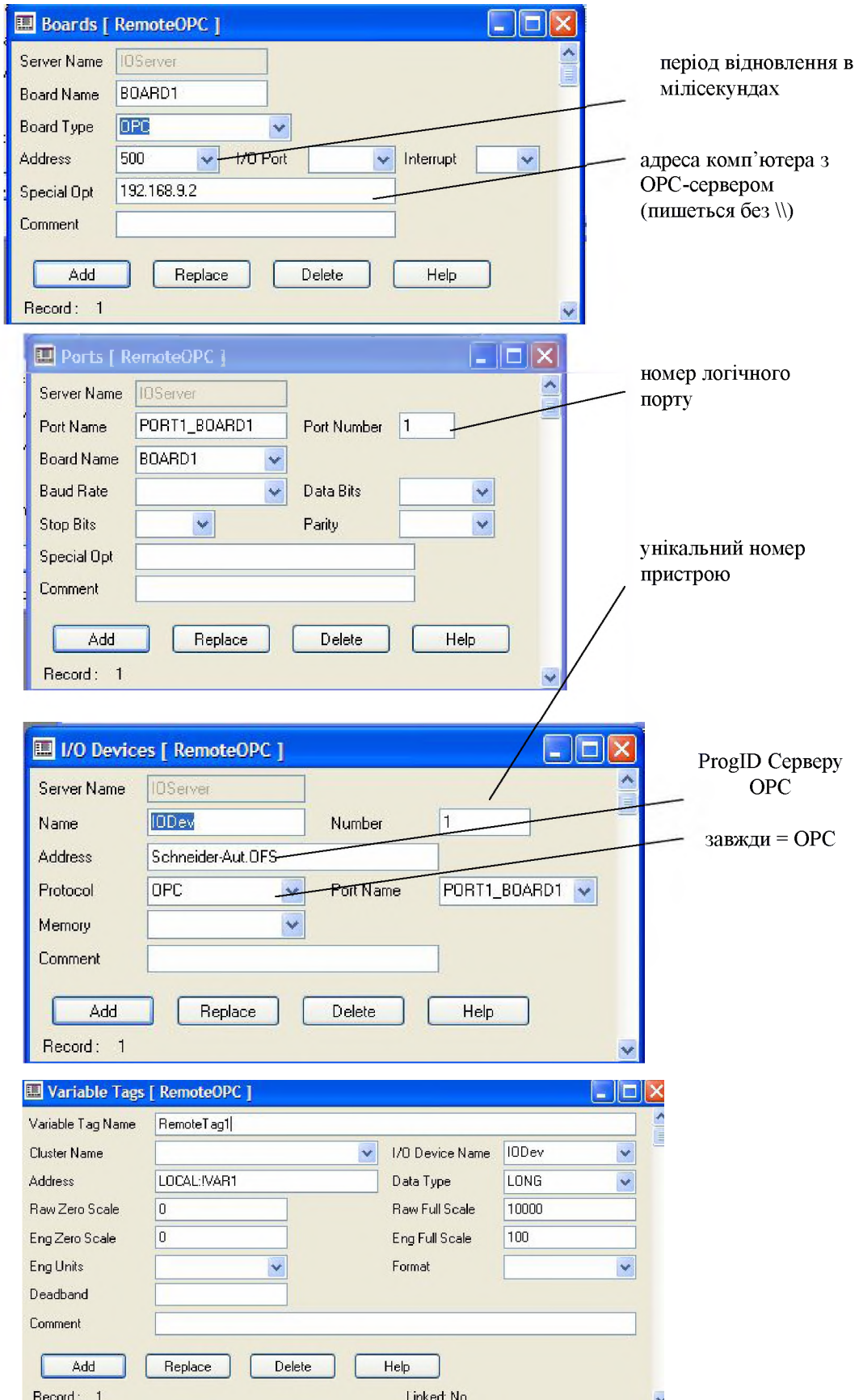


Рис.6.4 Вигляд вікна конфігурації OPC Серверу в VijeoCitect.

## ЛАБОРАТОРНА РОБОТА № 7 (модуль 4).

### Використання відкритих технологій доступу до баз даних в комп'ютерно-інтегрованих системах управління.

**Тривалість:** 4 акад. години (2 пари).

**Мета:** Навчитись користуватися технологіями доступу до баз даних для їх використання при побудові комп'ютерно-інтегрованих систем управління.

#### Лабораторна установка.

Лабораторна робота виконується на робочих місцях 2,3 (перша бригада), 4,5 (друга бригада), 7,8 (третя бригада) та 9,10 (четверта бригада). Лабораторна установка для робочих місць 2 та 3 зображена на рис.7.1. **Комп'ютер PC1 використовується для всіх робочих місць в якості технологічного сервера.**

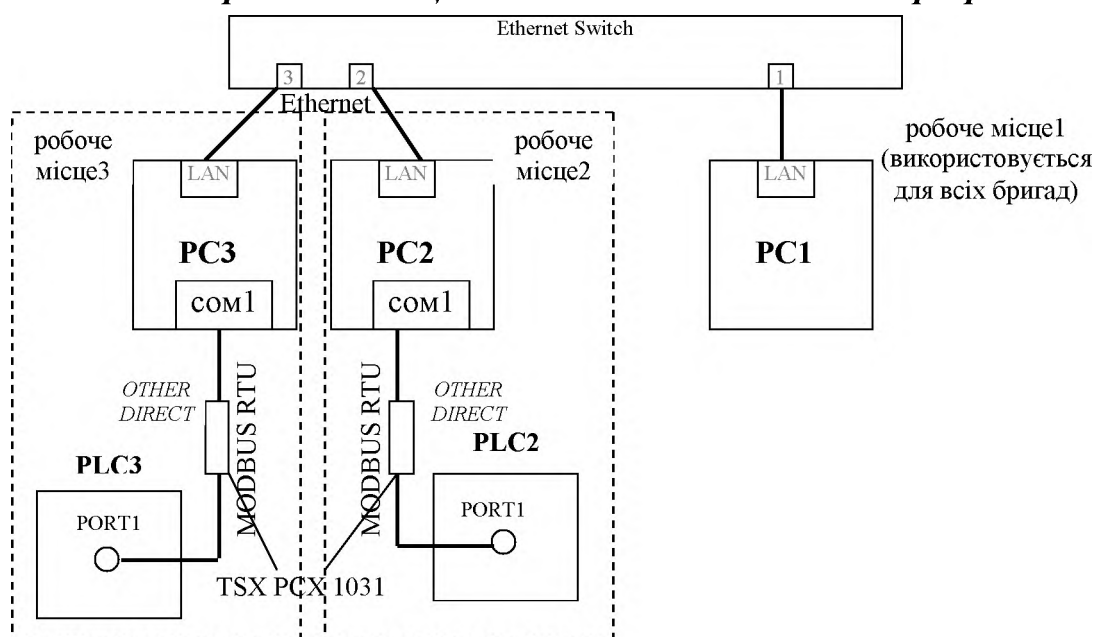


Рис.7.1. Лабораторна установка для першої бригади.

**Апаратне забезпечення.** Лабораторна установка складається з двох пар: ПЛК Twido – MODBUS RTU – ПК та ПК робочого місця 1 (PC1). ПК між собою з'єднані за допомогою Ethernet з використанням комутатора (Switch).

**Програмне забезпечення.** На комп'ютерах використовується SCADA-програма VijeoCitect (Шнейдер Електрик), Microsoft Excel, вбудований в ОС Windows XP Адміністратор ODBC, Microsoft® SQL Server™ Management Studio Express (SSMSE, Microsoft), на PC1 - Microsoft SQL Server 2000 (Microsoft).

**Загальна постановка задачі.** Цілі роботи:

- навчитись створювати БД для забезпечення ведення централізованого архіву;
- навчитись використовувати технології OLEDB та ODBC для доступу до джерел даних на прикладі SCADA VijeoCitect та Microsoft Excel;
- реалізувати поставлену задачу, оформлену у вигляді схеми інформаційних потоків, яка передбачає ведення централізованого архіву, та доступ до архівних даних з боку офісного прикладного забезпечення;

**Деталізований опис постановки задачі.** У лабораторній роботі необхідно реалізувати задачу, показану на моделі (рис.7.2) та схемі (рис.7.3) інформаційних потоків на прикладі варіанту для бригади 1. Система складається з трьох підсистем: підсистема 1, підсистема 2 та технологічний сервер.

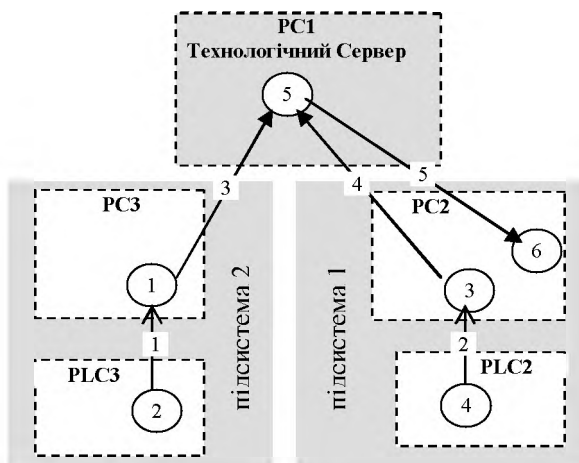


Рис.7.2. Постановка задачі у вигляді графової моделі мережних інформаційних потоків

**Функції:** 1,3 – збір та відображення технологічних параметрів витрат пари; 2,4 – збір та обробка даних; 5 – ведення централізованого архіву; 6 – розрахунок середніх витрат пари;

**Інформаційні потоки:** 1 – витрати пари F31, F32 з період. 2с; 2 – витрати пари F21, F22 – з період. 2с; 3 – витрати пари F31, F32 з період. 10 с; 4 – витрати пари F21, F22 з період. 10 с; 5 – архівні F21, F31 за останні 5 хв з період. 1 хв.

Реалізацію системи умовно можна поділити на чотири етапи.

1. У SCADA VijeoCitect кожної підсистеми реалізується зчитування значень витрат пари з використанням вбудованих драйверів Modbus RTU (потоки 1,2): F31, F32 з PLC3 (підсистема 2); F21, F22 з PLC2 (підсистема 1). Діапазон витрат ( $0-10000 \text{ м}^3/\text{год}$ ), періодичність зчитування 2 с;
2. З використанням середовища Microsoft® SQL Server™ Management Studio Express, на віддаленому вузлі технологічного серверу, що базується на Microsoft SQL Server, створюється база даних (DB1), таблиця T2 для підсистеми 1, та таблиця T3 для підсистеми 2, зі значеннями колонок відповідно F21, F22 та F31, F32. Таблиці необхідно спроектувати таким чином, щоб в одній з колонок зберігався час запису (див. додаток 7.1).
3. У SCADA VijeoCitect кожної підсистеми необхідно реалізувати архівування значень витрат (потоки 3,4) у відповідній таблиці технологічного серверу. Для ведення архіву використовується технологія ODBC та вбудована в VijeoCitect мова CiCode. Спочатку налаштовується ODBC DSN, який посилається на потрібну базу даних. У SCADA VijeoCitect створюється DEVICE з посиланням на потрібну таблицю джерела даних DSN (див. додаток 7.2). Створюється CiCode функція для добавлення запису в таблицю з плинними значеннями змінної. Для періодичності запису 10 с, в VijeoCitect створюється EVENT, який буде викликати функцію добавлення запису.
4. У програмі Microsoft Excel комп'ютера підсистеми 1, необхідно реалізувати зчитування архівних значень F21 та F31 за останні 5 хвилин (потік 5) та порівняння їх середніх значень у вигляді кругової діаграми. Періодичність вибірки – 1 хв. Доступ до архівних даних проводиться через механізми імпорту (див. додаток 7.4). Вибірка необхідних даних проводиться через SQL запити (див. додаток 7.3). Середні значення розраховується вбудованими функціями Excel, порівняння середніх значень  $F21_{\text{сер}}$  та  $F31_{\text{сер}}$  проводиться через кругову діаграму.



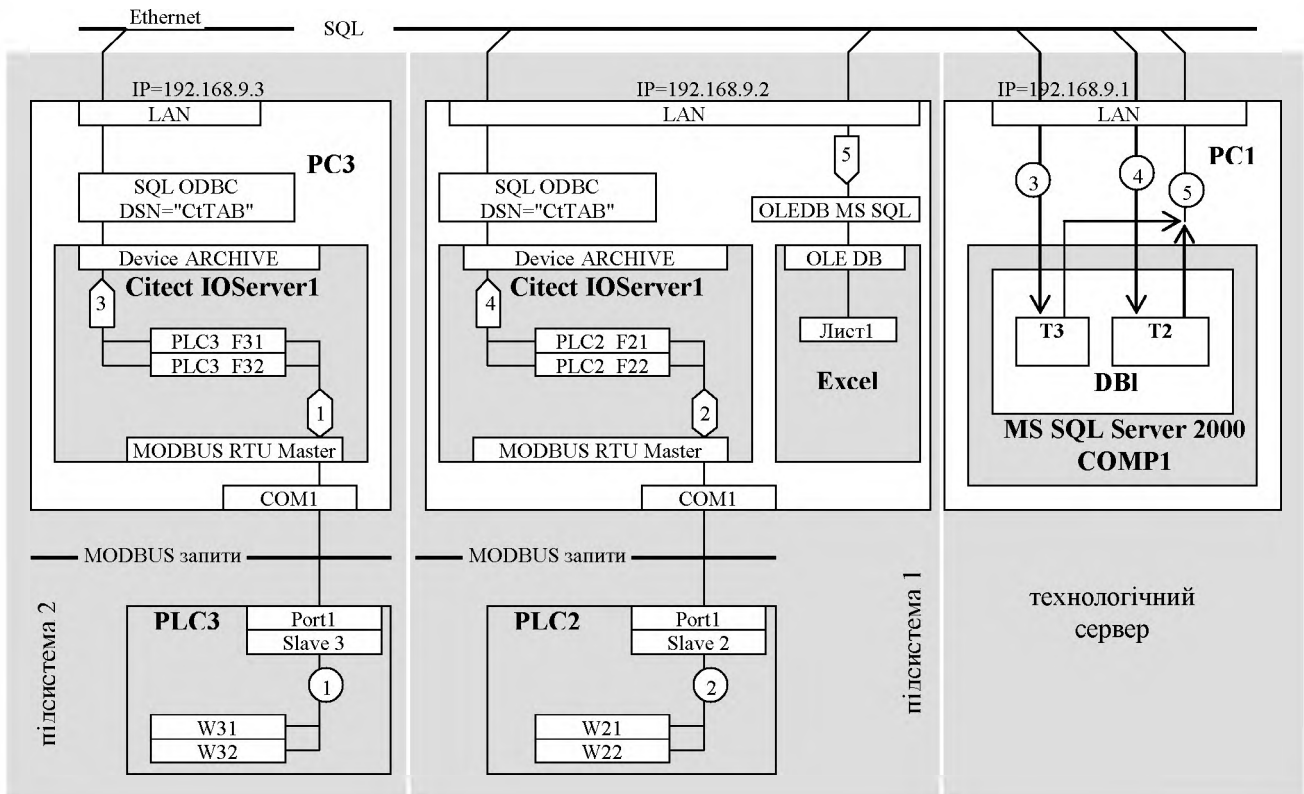


Рис. 7.3. Постановка задачі у вигляді схеми мережних інформаційних потоків

Таб. 7.1.

Параметр	бригада 1		бригада 2		бригада 3		бригада 4	
	підсист. 1	підсист. 2	підсист. 1	підсист. 2	підсист. 1	підсист. 2	підсист. 1	підсист. 2
робоче місце	2	3	4	5	7	8	9	10
Адр. Slave	2	3	4	5	7	8	9	10
бітова швидк.	19200	19200	19200	19200	9600	9600	9600	9600
паритет	парн	парн	парн	парн	непар	непар	непар	непар
стоп біт	1	1	1	1	1	1	1	1
змін.1 PLC	W21 %MW123	W31 %MW68	W21 %MW34	W31 %MW176	W21 %MW34	W31 %MW11	W21 %MW67	W31 %MW95
змін.2 PLC	W22 %MW124	W32 %MW69	W22 %MW35	W32 %MW177	W22 %MW35	W32 %MW12	W22 %MW68	W32 %MW96
назва БД, користувач, пароль	DB1, користувач G1, пароль 1		DB2, користувач G2, пароль 1		DB3, користувач G3, пароль 1		DB4, користувач G4, пароль 1	

### Послідовність виконання роботи.

#### 1. Забезпечення читання змінних з ПЛК.

1.1) Зібрати лабораторну установку.

**УВАГА! Port1 ПЛК Twido і COM-порт ПК не ізолювані, тому для безпечного з'єднання необхідно підключати перетворювачі при вимкнених ПЛК, з дозволу викладача.**

1.2) У VijeoCitect реалізувати потоки 1 (підсистема 2) та 2 (підсистема 1). Результат перевірити в режимі виконання.

#### 2. Створення БД та таблиць на технологічному сервері.

2.1) На одному з робочих місць бригади (ПК підсистеми 1 або 2) запуснути на виконання Microsoft® SQL Server™ Management Studio Express (SSMSE). З'єднатися з сервером: ім'я COMP1, схема аутентифікації - "SQL Server", користувач та пароль - відповідно до варіанту; опцію заповнити пароль **не виставляти**.

2.2) Перевірити наявність бази даних відповідно до варіанту (наприклад для бригади 1 – DB1). У випадку наявності такої – видалити її.

2.3) Створити базу даних, відповідно до варіанту.

2.4) Створити таблицю T2 з наступними властивостями:

Стовпчик 1: Ім'я = F21, тип = char(10), дозволити значення null;

Стовпчик 2: Ім'я = F22, тип = char(10), дозволити значення null;

Стовпчик 3: Ім'я = DateTime, тип = datetime, дозволити значення null, Значення або прив'язка по замовченню – GetDate();

Вигляд конфігураційного вікна зафіксувати скріншотом.

2.5) Створити таблицю T3 з наступними властивостями:

Стовпчик 1: Ім'я = F31, тип = char(10), дозволити значення null;

Стовпчик 2: Ім'я = F32, тип = char(10), дозволити значення null;

Стовпчик 3: Ім'я = DateTime, тип = datetime, дозволити значення null, Значення або прив'язка по замовченню – GetDate();

Вигляд конфігураційного вікна зафіксувати скріншотом.

### 3. Архівування даних в VijeoCitect.

3.1) Сконфігурувати джерело даних ODBC з ім'ям DSN=CtTAB, що по замовченню повинна посилатися на потрібну базу даних, відповідно до варіанту.

- "Пуск" -> "Настройка" -> "Панель Управления" -> "Администрирование" -> "Источники данных (ODBC)";
- "Системный DSN", "Добавить";
- вибрати драйвер SQL Server;
- налаштувати драйвер на відповідне джерело даних:
  - ім'я (DSN-ім'я): CtTab;
  - аутентифікація: SQL;
  - користувач та пароль, відповідно до варіанту;
- перевірити з'єднання за допомогою тестової кнопки;

3.2) У VijeoCitect створити DEVICE (System->Devices) типу SQL з іменем "Archive", зв'язаний з ODBC, створеним в п 3.1 (DSN=CtTAB). Вказати ім'я таблиці відповідно до варіанту. Вказати формат таблиці, відповідно до варіанту за прикладом:

{F21,10} {F22,10}

Вигляд конфігураційного вікна зафіксувати скріншотом.

3.3) У VijeoCitect створити новий файл Cicode (Активний Проект->Cicode Files->Create a new Cicode Page) з довільною назвою і записати в ньому функцію з іменем *FnWriteToSQL* за прикладом, наведеним в додатку 7.2. Зберегти файл.

3.4) У VijeoCitect створити генератор подій EVENT (System->Events) для виклику функції *FnWriteToSQL* 1 раз/10с (відповідно до параметрів інформаційного потоку). У параметрах новоствореного EVENT задати:

*Name* = EventWriteDB  
*Action* =FnWriteToSQL()  
*Period* = 00:00:10

Вигляд конфігураційного вікна зафіксувати скріншотом.

3.5) Скопіювати проект VijeoCitect. Після вдалої компіляції необхідно активувати генератор подій для системи виконання. У Citect Explorer запустити ComputerSetupWizard (Tools->ComputerSetupWizard), вибрати режим CustomSetup, у вікні Network Setup вибрати "No Networking", у вікні EventsSetup

виставити опцію "Enable Events on This Computer" та "EventWriteDB". Запустити виконавчу систему VijeoCitect.

3.6) Спостерігати за роботою системи протягом 1 хв, у випадку з'явлення повідомлення "Не вдається відкрити базу даних" перевірити всі підпункти пунктів 2 та 3, та досягнути вдалого результату.

3.7) Не зупиняючи виконавчу систему VijeoCitect, переглянути зміст таблиць T2 та T3 в базі даних технологічного серверу, використовуючи SSMSE. Якщо даних в таблицях немає перевірити всі підпункти пунктів 2 та 3, та досягнути вдалого результату.

3.8) Періодично оновлюючи таблицю (наприклад через "Выполнить код SQL") дочекатися, коли з'являться записи в таблицях T2 та T3 старіше за 5 хвилин. Вигляд таблиць T2 та T3 зафіксувати скріншотом.

3.9) У SSMSE для таблиці T2 показати область SQL-кода. Відредагувати запит змінивши його на вибірку полів F21 та DateTime таблиці T2 за останні 5 хвилин. Вигляд таблиці T2 та SQL-запиту зафіксувати скріншотом.

3.10) У SSMSE для таблиці T3 показати область SQL-кода. Записати запит на вибірку полів F31 та DateTime таблиці T3 за останні 5 хвилин. Вигляд таблиці T3 та SQL-запиту зафіксувати скріншотом.

#### **4. Реалізація вибірки архівних даних в Excel.**

4.1) На PC підсистеми 1 завантажити Excel. Імпортувати дані з таблиць T2 та T3 використовуючи OLE DB. У властивостях зовнішніх даних "Изменить запрос", визначить рядок підключення, тип команди та текст команди. Результат зафіксуйте скріншотом та занотуйте. У властивостях діапазону зовнішніх даних визначить фонове оновлення екрану кожну хвилину.

4.2) Повторити пункт 4.1 для таблиці T3 але в інших комірках.

4.3) У параметрах зовнішніх даних "Изменить запрос" змініть тип команди на SQL, в полі тексту команди введіть запит як в пункті 3.9. Результат зафіксуйте скріншотом.

4.4) Повторити пункт 4.3 для таблиці зовнішніх даних з таблиці T3, з запитом аналогічним як в пункті 3.10.

4.5) Для перетворення текстових даних в числові в сусідніх колонках напроти полів зі значеннями F21 та F31 використати функцію Excel "ЗНАЧЕН", яка знаходиться в бібліотеці категорії "текстовые". Новостворені колонки назвати F21ч та F31ч. На основі створених даних побудувати діаграму типу "График". Результат зафіксуйте скріншотом.

4.6) Для підрахунку середнього арифметичного по кожній із колонок F21ч та F31ч використати функцію Excel "СРЗНАЧ", яка знаходиться в бібліотеці категорії "Статистические". На основі середніх даних побудувати кругову діаграму (необхідні дані вибрати Ctrl).

#### **Оформлення роботи.**

До захисту готуються всі зафіксовані сріншоти та занотовані результати з поясненнями.

#### **Перевірка виконання роботи та питання до захисту.**

Викладачем вибірково перевіряється виконання всіх пунктів роботи та занотовані результати. Кожен результат студент повинен пояснити. У випадку виникнення помилок або запитань щодо проведення певного пункту, цей пункт необхідно буде повторити.

1. Які програмні засоби і навіщо доступуються до архівних даних? Який тип СУБД використовувався в даній лабораторній роботі в якості джерела даних? Які програмні засоби в даній лабораторній роботі доступалися до даного джерела? Які відкриті технології доступу до баз даних при цьому використовувалися?
2. Як вирішується питання стандартизації інтерфейсу до СУБД через використання мови SQL? Що таке діалект SQL, який діалект використовувався в СУБД в даній лабораторній роботі? Які особливості СУБД були використані в лабораторній роботі?
3. Поясніть синтаксис оператора вибірки SELECT? Де і навіщо в даній лабораторній роботі використовувався даний оператор? Поясніть використання оператора SELECT на основі занотованих результатів.
4. На яких технологіях базується ODBC. Як ідентифікується джерело даних в ODBC? Поясніть всю послідовність створення DSN, яку Ви проводили в лабораторній роботі.
5. Наведіть послідовність операцій, яку необхідно зробити користувачеві для вказівки джерела даних програмі клієнту ODBC на прикладі підключення його в VijeoCitect.
6. Поясніть призначення об'єктів Connection (Connection String) та Command в технології ADO/OLE DB? Яким чином формується Рядок Підключення? Прокоментуйте дані та інші поля на прикладі пункту 4.3 даної лабораторної роботи.
7. Чи можна доступитись до джерел даних ODBC через інтерфейс OLE DB та ADO? Обґрунтуйте свою відповідь.
8. Поясніть пункти 3.3 та 3.4 даної лабораторної роботи. Поясніть програму написану на CiCode.

## Додаток 7.1. Робота з Microsoft SQL Server Management Studio Express (SSMSE).

**Д7.1.1. Загальні положення.** Середовище Microsoft® SQL Server™ Management Studio Express (SSMSE) надає графічний інтерфейс для управління більшістю реляційних СУБД SQL Server версії 2005 та нижче. Середовище SSMSE можна безкоштовно завантажити з офіційного сайту Microsoft.

У середовищі SSMSE всі операції з базами даних можна проводити з використанням як графічних засобів так і SQL запитів.

**Д7.1.2. З'єднання з необхідним Сервером.** По замовченню ярлик для SSMSE знаходиться в "Пуск->Програми->Microsoft SQL Server 2005-> SQL Server Management Studio Express ". Підключення до необхідного екземпляру SQL Server проводиться наступним чином:

"меню Файл->Підключити к обозревателю объектов..."

У діалоговому вікні вказати необхідне ім'я екземпляру Серверу, тип аутентифікації ("перевірка подлинности"), ім'я користувача та пароль (рис.7.4). При необхідності зв'язку з віддаленим SQL Сервером його ім'я вказується в форматі:

ім'я\_комп'ютера\ім'я\_екземпляру\_сервера

У даній лабораторній роботі для підключення до MS SQL Server 2000, достатньо вказати тільки ім'я комп'ютера, тобто COMP1.

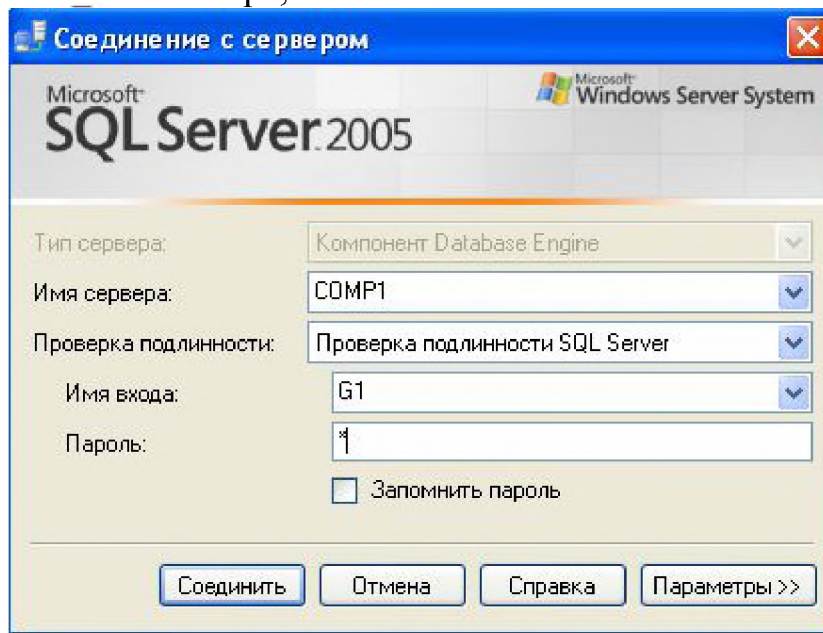


Рис.7.4. Вікно вибору сервера

**Д7.1.3. Додавання та знищення бази даних.** Додавання бази даних (БД) в графічному редакторі SSMSE можна зробити через контекстне меню "Базы данных" вибраного екземпляру Серверу. У діалоговому вікні (рис.7.5) достатньо вказати ім'я новостворюваної БД і натиснути "Ок", база даних створиться з параметрами по замовченню, що задовольняє умовам даної лабораторної роботи.

Видалення БД проводиться в її контекстному меню. Враховуючи, що користувачам в даній лабораторній роботі не надані адміністративні права на знищення журналів, у діалоговому вікні перед знищенням необхідно зняти опцію "Удалить журналы резервного копирования и восстановления базы данных", інакше БД не буде видалена.

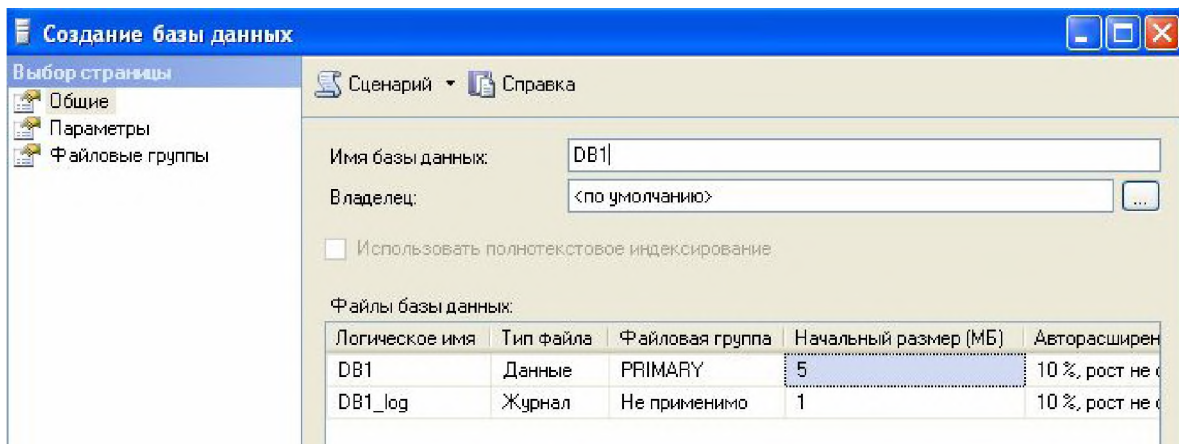


Рис.7.5. Вікно створення БД

**Д7.1.4. Створення та видалення таблиць в базі даних.** У SSMSE Таблиці створюються в контекстному меню бази даних "Таблиці". При створенні викликається конструктор таблиць (рис.7.6) в якому задається структура таблиці: стовпці та при необхідності їх зв'язки. Збереження структури таблиці проводиться в меню "Файл->Сохранить Таблицу". Конструктор таблиці також можна викликати вже після створення таблиці, викликавши в її контекстному меню команду "Изменить".

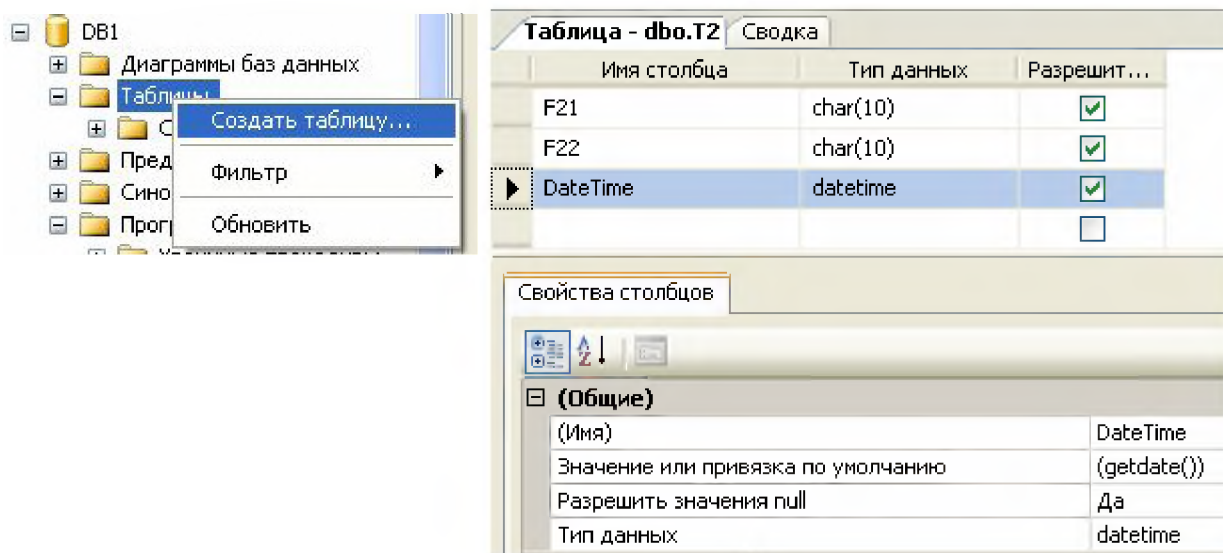


Рис.7.6. Вікно конструктора таблиць

Знищення таблиці проводиться через її контекстне меню.

**Д7.1.5. Вибірка та модифікація даних в SSMSE.** Перегляд та редагування записів таблиць графічними засобами проводиться в конструкторі запитів, через контекстне меню "Открыть таблицу" (рис.7.7). При цьому в головному меню з'являється пункт "Конструктор запросов", в якому в підменю "Область" можна вибрати різні типи відображення (Області). Так вибірку потрібних даних можна проводити за допомогою SQL запитів в області SQL-кода. Перевірку SQL-коду та запуск його на виконання можна проводити через відповідні команди меню "Конструктора запросов".

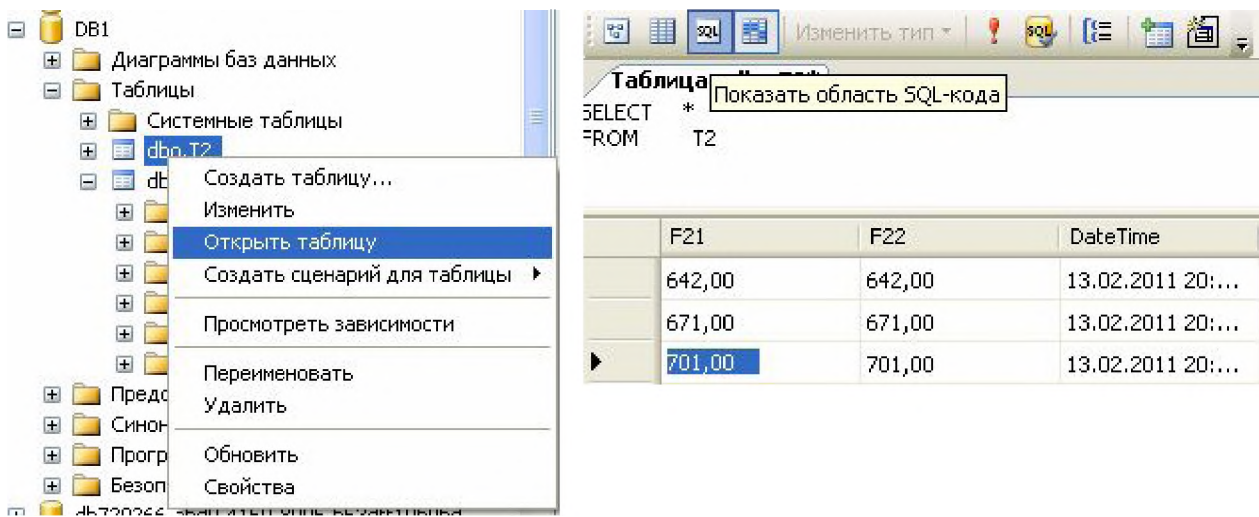


Рис.7.7. Вікно конструктора запитів з активованою областю SQL-коду

## Додаток 7.2. Робота VijeoCitect з базами даних SQL.

**Д7.2.1. Загальні положення.** У VijeoCitect є декілька механізмів роботи з джерелами даних типу SQL. Один з найпростіших – це використання спеціальних об'єктів для роботи з зовнішніми пристроями – **DEVICE** разом з **CiCode** командами.

Об'єкти DEVICE створюються в Project Editor в розділі System->Devices. Одним із типів DEVICE є ODBC джерело даних типу таблиця. VijeoCitect підтримує роботу тільки з символьним (текстовим) типом даних. Базу даних та таблицю в ній необхідно створити перед її використанням в Device.

**Д7.2.2. Налаштування параметрів DEVICE для роботи з SQL джерелами даних.** Для роботи з DEVICE типу SQL необхідно сконфігурувати наступні поля (рис.7.8):

*Name* – довільна назва пристрою, наприклад *DevDB*

*Format* – формат таблиці, тобто назва та ширина в символах колонок таблиці. Наприклад, формат - *{Name,16}{Water,8}{Sugar,8}{Flour,8}{Salt,8}{Yeast,8}{Milk,8}* буде визначати таблицю наступного вигляду:

Name	Water	Sugar	Flour	Salt	Yeast	Milk

*Header* – для SQL типу пристрою вказується рядок підключення, наприклад:  
`DSN=ExmplDataBase; UID=G1; PWD=1`

вказує на ім'я DSN "ExmplDataBase", ім'я користувача (UID) рівним "G1", пароль (PWD) рівним "1".

*FileName* – ім'я таблиці в базі даних, до якої необхідно підключитись

*Type* – для SQL типу пристрою вибирається SQL\_DEV

*NoFiles* – кількість файлів, для SQL типу пристрою дорівнює -1

Поля *Time*, *Period*, *Prpcess*, *ClusterName*, *Comment* в даній лабораторній роботі необхідно залишити порожніми.

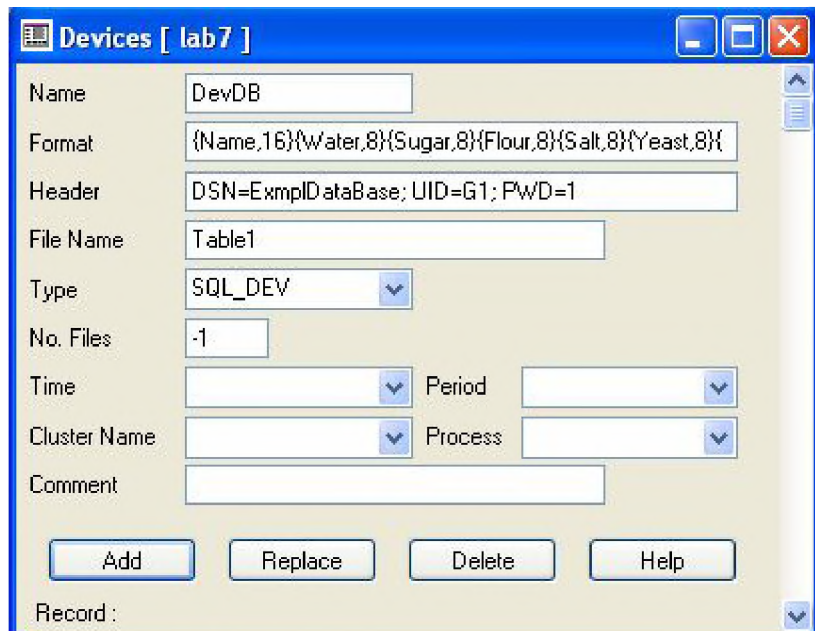


Рис. 7.8. Вікно конфігурації Device в VijeoCitect

**Д7.2.3. Додавання записів в джерело даних DEVICE з використанням CiCode.** Додавання нових записів в CiCode проводиться через функції DevAppend, зміна значень полів запису через функцію DevSetField. Робота даних функцій з конкретним Device проводиться через дескриптор (handle), який повертається функцією DevOpen.

Роботу функцій розглянемо через функцію користувача FnWriteToSQL, яка заносить новий запис в DEVICE з ім'ям Archive, та полями F21 та F22, які вміщують значення витрат.

```

FUNCTION FnWriteToSQL()
  INT hDev; //дескриптор пристрою
  ErrSet(1); //обробку помилок робить програма користувача

  //відкрити DEVICE з іменем Archive, який буде посилатись на таблицю T2
  hDev=DevOpen("Archive",0);

  IF hDev = -1 THEN          //якщо помилка відкриття
    Message("Не вдається відкрити базу даних","Не вдається відкрити базу даних",0);
  ELSE                      //якщо помилки немає
    //добавити новий запис в таблицю
    DevAppend(hDev);

    //записати в поле F21 значення змінної PLC2_F21
    DevSetField(hDev, "F21", RealToStr(PLC2_F21,5,2));
    //де F21 – назва колонки в таблиці бази даних
    //PLC2_F21 – назва змінної в VijeoCitect, значення якої треба записувати

    //записати в поле F22 значення змінної PLC2_F22
    DevSetField(hDev, "F22", RealToStr(PLC2_F22,5,2));
    //де F22 – назва колонки в таблиці бази даних
    //PLC2_F22 – назва змінної в VijeoCitect, значення якої треба записувати

  END
DevClose(hDev);
END

```



## Додаток 7.3. Використання SQL запитів.

**Д7.3.1. SQL запити на вибірку.** Загальний вигляд SQL-запита на вибірку рядків(записів) з таблиць:

```
SELECT список_полів_через_кому  
FROM таблиця_БД  
WHERE умова_вибірки_записів
```

Наприклад, наступний запит повертає всі поля таблиці T2, які задовольняють умові більшості поля DateTime за '2006-09-19 00:00:00':

```
SELECT *  
FROM T2  
WHERE T2.DateTime>'2006-09-19 00:00:00'
```

**Д7.3.2. Використання вбудованих функцій MS SQL Server.** MS SQL Server дає можливість створювати власні та використовувати існуючі функції (Function) та збережені процедури (Stored Procedure). В лабораторній роботі в запитах SQL використовуються вбудовані функції SQL Server для роботи із датою та часом: DATEADD та GETDATE.

DATEADD (*частина, кількість, дата\_час*) – додає до вказаної дати та часу вказану кількість частин (секунд, хвилин, годин і т.д.)

GETDATE() – повертає плинне значення дати та часу. Формат часових даних – *datetime*, може мати вигляд 'уууу-мм-дд hh:mi:ss', наприклад: '2006-10-10 12:36:48'. Параметр *частина* в функціях задається відповідними буквами, тобто "уууу"-рік, "mm" – місяць, "dd" – хвилини, "hh"- години, "mi" – хвилини, "ss" – секунди.

Наприклад, наступний запит повертає поля F21 та DateTime таблиці T2 за останні 5 хвилин, при умові що в полі DateTime зберігається дата та час збереження.

```
SELECT F21, DateTime  
FROM T2  
WHERE T2.DateTime> DATEADD ("mi",-5,GETDATE())
```

## Додаток 7.4. Імпорт даних в Excel.

У таблиці Excel можна імпортувати зовнішні дані, наприклад із баз даних, використовуючи інтерфейс OLE DB або ODBC.

В лабораторній роботі використовується імпорт даних із баз даних та файлів використовуючи інтерфейс OLE DB. Імпорт даних через OLE DB проводиться наступним чином: "Данные->Импорт внешних данных->Импортировать данные...". При виборі цього способу імпорту необхідно вказати джерело даних, настройки якого збережені в файлі, або створити нове Підключення за допомогою кнопки "Создать". Для повної настройки джерела даних в меню майстра Підключення треба вибрати пункт "Дополнительно", після чого вибрати необхідного провайдера даних та налаштувати параметри Підключення. Після вставки даних в таблицю Excel можна змінити Рядок Підключення, Тип та Текст Команди через контекстне меню вставлених даних "Изменить запрос". Наприклад можна змінити тип запиту на SQL, і вписати SQL-запит в поле "Текст команды"

## ЛАБОРАТОРНА РОБОТА № 9 (модуль 3).

### Конфігурування мережі CANOpen, використання сервісів PDO та SDO. Використання профілю CiA-402 для управління пристроями PDS.

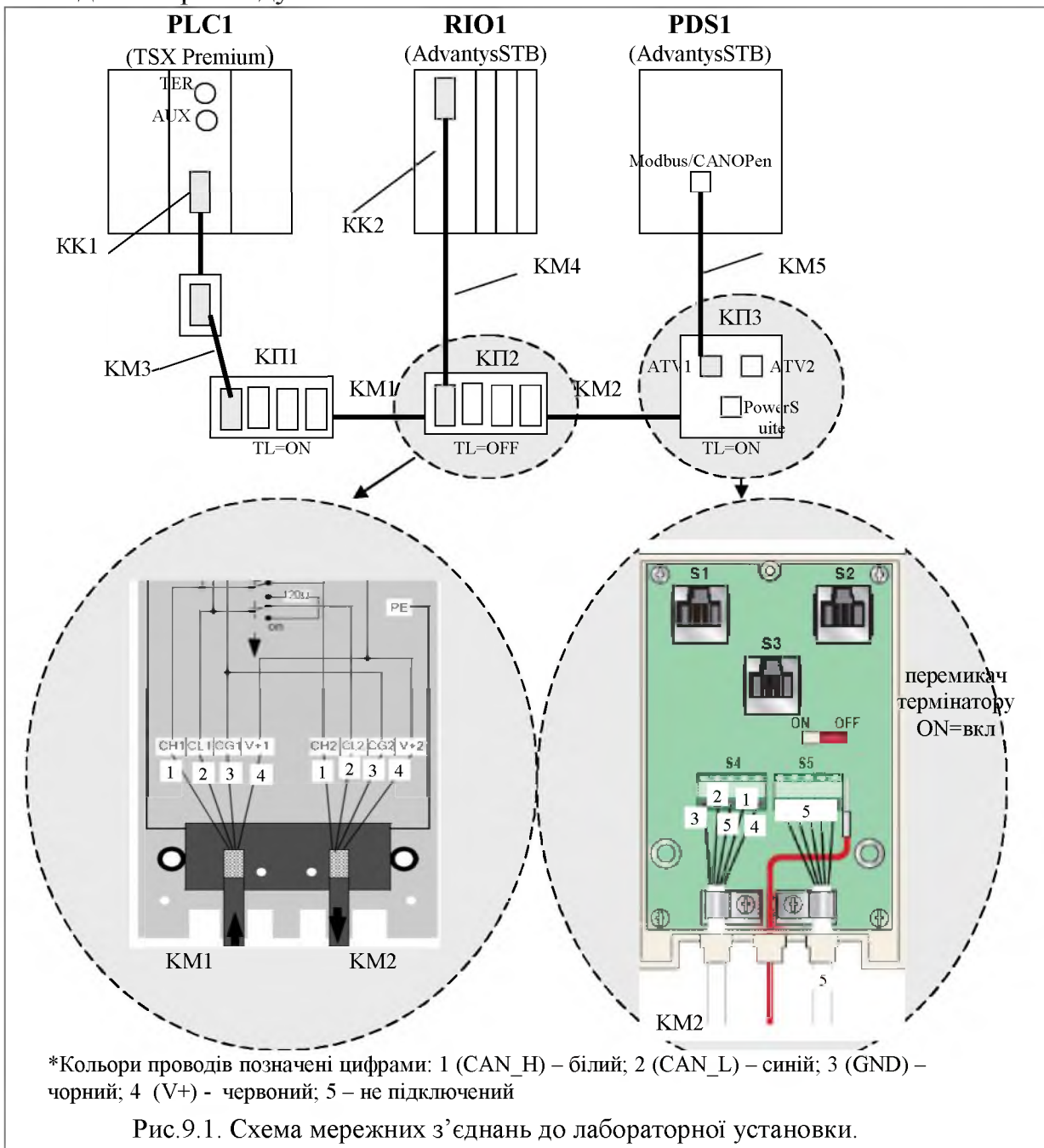
#### ЧАСТИНА 1.

**Тривалість:** 2 акад. години (1 пара).

**Мета:** Навчитись створювати конфігурацію мережі CANOpen відповідно до завдання.

#### Лабораторна установка.

Лабораторна робота виконується на робочих місцях 2-5 двома бригадами, інші дві бригади в цей час виконують другу частину лабораторної роботи №9. Лабораторна установка зображена на рис.9.1, для частини 1 використовується тільки в якості наглядного прикладу.



**Апаратне забезпечення.** В таблиці наведена специфікація мережних засобів. Опис роботи засобів та способи їх підключення наведені в додатках 9.1 – 9.3.

№	Позначення	Назва	Тип	Кількість	Фірма
1	PLC1	ПЛК TSX Premium	TSX 57204M	1 шт.	SE
2	RIO1 (KK2)	Острів віддаленого вводу/виводу Advantys STB, з комунікаційним модулем CANOpen AdvantysSTB	STB NCO 2212	1 шт.	-/-
3	PDS1	Частотний перетворювач Altivar71 для управління асинхронним двигуном	ATV71H037M3	1 шт.	-/-
4	KK1	Комунікаційна PCMCIA карта CANOpen для TSX Micro/Premium	TSX CPP110	1 шт.	-/-
5	KM1, KM2	Магістральний кабель (подвійна екранована вита пара для CANOpen)	TSX CAN CA100	40 м	-/-
6	KM3, KM4	Кабель для підключення пристроїв по CANOpen, з обох боків 9-пінова SUB-D розетка, 1 м.	TSX CAN CADD1	1 шт.	-/-
7	KM5	Кабель для підключення пристроїв по CANOpen, з обох боків RJ-45 вилка, 1 м	TSX CAN CARR1	1 шт.	-/-
8	КП1, КП2	Коробка підключення пристроїв до магістральної шини CANOpen з 4-ма портами типу 9-пінова SUB-D вилка	TSX CAN TDM4	1 шт.	-/-
9	КП3	Коробка підключення частотних перетворювачів Altivar до магістральної шини CANOpen, 3 порти RJ-45 розетка (2 – CAN, 1 – Modbus RTU)	VW3 CAN TAP 2	1 шт.	-/-

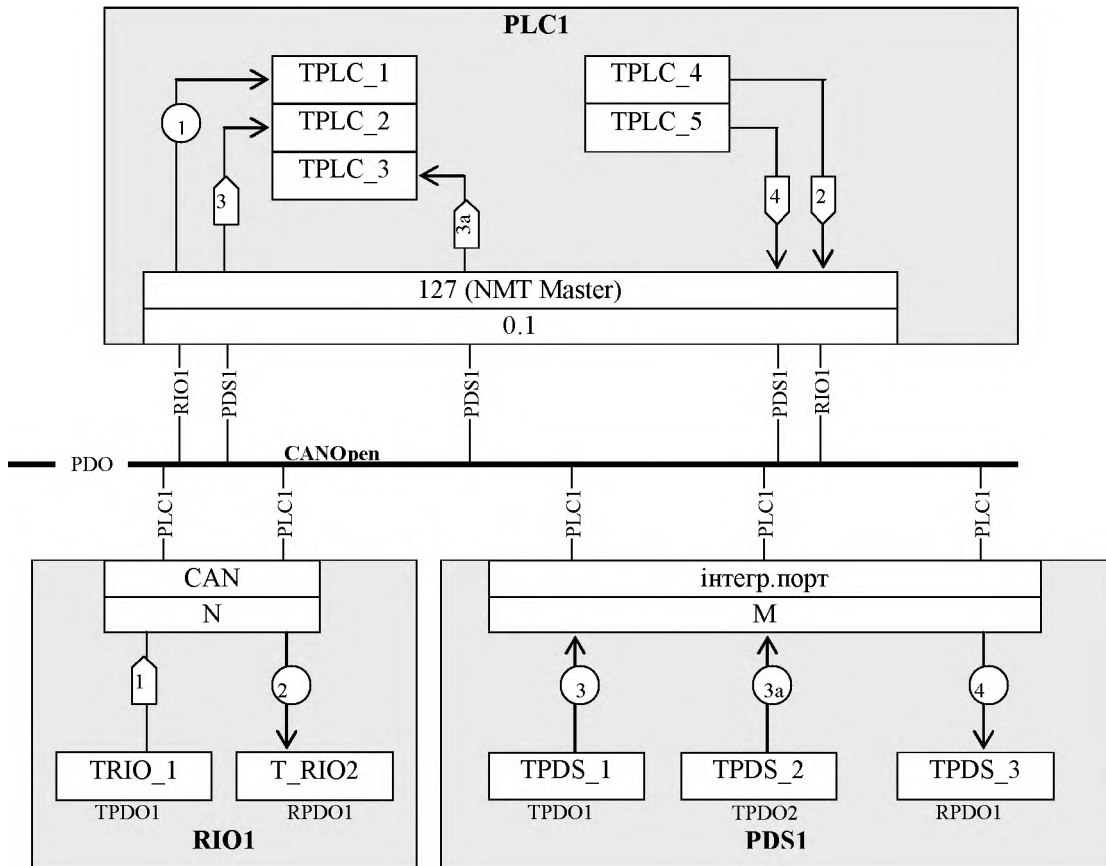


Рис.9.2. Схема мережних інформаційних потоків до лабораторної роботи

**Програмне забезпечення.** Для створення конфігурації для PLC1 використовується UNITY PRO, для створення конфігурації для TSX CPP 110 використовується SyCon.

**Загальна постановка задачі.** Ціль роботи – створити конфігурацію мережі для TSX CPP 110 та PLC Premium відповідно до поставленої задачі, оформлену у вигляді схеми інформаційних потоків (рис 9.2) та варіанту. В даній роботі вважається, що пристрої RIO1 та PDS1 вже сконфігуровані.

**Послідовність виконання роботи.**

1. Запустити SyCon та створити конфігурацію мережі відповідно до таблиці

Параметр		робоче місце			
		2	3	4	5
*.EDS RIO1		STB NCO 2212			
*.EDS PDS1		ATV 71 V2.6			
Адреса RIO1 (N)		1	2	3	4
Адреса PDS1 (M)		2	3	4	5
бітова швидкість		1 Мбіт/с	500 Кбіт/с	250 Кбіт/с	100 Кбіт/с
періодичність SYNC		100 мс	200 мс	250 мс	500 мс
управління RIO1 при старті (Д9.4.3)		всі стадії активні	всі стадії активні	без NODE RESET та перевірки профілю	без конфігурації пристрою
управління PDS1 при старті (Д9.4.3)		без NODE RESET	без перевірки профілю	всі стадії активні	всі стадії активні
протокол контролю за помилками		для всіх вузлів Node Guard	для всіх вузлів Heartbeat	для всіх вузлів Node Guard	для всіх вузлів Heartbeat
PLC1	TPLC_1	%MW10	%MW20	%MW30	%MW40
	TPLC_2	%MW11-%MW14	%MW21-%MW22	%MW31-%MW34	%MW41-%MW44
	TPLC_3	%MW15-%MW17	%MW23-%MW25	%MW35-%MW36	відсутній
	TPLC_4	відсутній	%MW50	%MW60	%MW70
	TPLC_5	%MW40	%MW51	%MW61-%MW62	%MW71-%MW72
RIO1	TRIO_1	DI0-DI7	DI0-DI7	DI0-DI7	DI0-DI7
	TRIO_2	відсутній	DO0-DO7	DO0-DO7	DO0-DO7
	Об'єкт Словника*	відсутній	відсутній	Index/SubInd=3000/0/0 значення FF <sub>16</sub>	Index/SubInd=3001/0/0 значення FF <sub>16</sub>
PDS1	TPDS_1	Status Word 6041/0	Control Effort	Frequency Ref	Status Word
		Control Effort 6044/0	Status Word	Status Word	Control Effort
		Frequency Ref 2002/4		Motor Current	Frequency Ref
		Motor Current 2002/5		Motor Torque	Motor Current
	TPDS_2	Motor Torque 2002/6	Frequency Ref	Mains Voltage	відсутній
		Mains Voltage 2002/8	Motor Current	Motor Voltage	
		Motor Voltage 2002/9	Motor Torque		
TPDS_3	Control Word 6040/0	Target Velocity 6042/0	Control Word	Target Velocity	
			Target Velocity	Control Word	
Об'єкт Словника*	Index/SubInd=2000/13 значення 10 <sub>16</sub>	Index/SubInd=2000/14 значення 5 <sub>16</sub>	відсутній	відсутній	
Параметри ін форм. потоків	потік 1	синхронно через кожні 20 Об'єктів SYNC	при запиті RTR, через кожні 10 циклів NMT-Ведучого	синхронно, при зміні значення	при запиті RTR, через кожні 15 циклів NMT-Ведучого
	потік 2	відсутній	визначено профілем асинхронно через кожні 5 циклів NMT-Ведучого	синхронно при зміні значення	синхронно через кожні 20 Об'єктів SYNC
	потік 3	при запиті RTR, через кожні 30 циклів NMT-Ведучого	визначено профілем пристрою, кожні 2 с	при запиті RTR, через кожні 20 циклів NMT-Ведучого	визначено профілем пристрою, кожні 500 мс
	потік 3а	визначено профілем пристрою, кожні 500 мс	синхронно через кожні 20 Об'єктів SYNC	визначено профілем пристрою, кожні 1 с	відсутній
	потік 4	визначено профілем асинхронно через кожні 10 циклів NMT-Ведучого	синхронно при зміні значення	синхронно через кожні 20 Об'єктів SYNC	асинхронно через кожні 10 циклів NMT-Ведучого

\* - додатковий Об'єкт Словника та його значення, який необхідно змінити при конфігурації NMT-Ведучим в NMT-Веденому вузлі

2. Порівняти SDO таблицю з наведеними значеннями в таблиці T9.1 (відповідно до варіанту), якщо вона не співпадає повернутися до пункту 1.
3. Зберегти файл конфігурації на диску під назвою "lab9.co".
4. Завантажити UNITY PRO та створити конфігурацію ПЛК TSX 572634M (OS2.0), з комунікаційною картою TSX CCP 110. Налаштувати параметри комунікаційної карти відповідно до варіанту. Параметри конфігурації записати в таблицю T9.2.

### **Оформлення роботи.**

До захисту готуються дві таблиці T9.1 та T9.2 (див. додаток 9.5) відповідно до варіанту. В полі коментар таблиці T9.1 вписується призначення запису та пояснення числового значення.

### **Перевірка виконання роботи та питання до захисту.**

Викладачем перевіряється виконання поставленого завдання шляхом звірки значень отриманої студентом SDO Table та таблиці T9.1, а також будь який із пунктів завдання по вибору. Захист роботи проводиться по T9.1 та T9.2. Студент повинен пояснити призначення всіх полів, та їх значення.

1. На яких стандартах на фізичному рівні базується CANOpen і які додаткові вимоги ставляться на цьому рівні? Наведіть основні характеристики CANOpen на фізичному рівні: спосіб передачі бітів, топологія, правила термінування, максимальна довжина ліній зв'язку та відгалужень, бітова швидкість, максимальна кількість пристроїв. Як це реалізовано в лабораторній роботі.
2. Розкажіть про призначення Словнику Об'єктів. Яким чином визначається місце розміщення Об'єктів в Словнику?
3. Перерахуйте комунікаційні сервіси, які використовуються в CANOpen. Поясніть їх призначення. Які Комунікаційні Об'єкти використовуються для реал-тайм обміну невеликими обсягами даних процесу? Які Комунікаційні Об'єкти використовуються для обміну великими обсягами даних?
4. Що забезпечують функції сервісу PDO? Які комунікаційні Об'єкти використовуються для функціонування даного сервісу? Поясніть призначення T-PDO та R-PDO. Як забезпечується їх взаємний зв'язок та зберігається інформація про нього?
5. Як забезпечується зв'язок даних T-PDO/R-PDO з Об'єктами Словнику? Як називається цей механізм? Де зберігаються настройки таких зв'язків?
6. Яке призначення NMT-сервісів? Яке призначення NMT-Ведучого в CANOpen та які функції він виконує? Поясніть різницю між NMT-Ведучим та Ведучим шини. Як ідентифікується конкретний вузол в CANOpen і для чого це потрібно?
7. Яким чином забезпечується зв'язок між Комунікаційними Об'єктами вузлів до їх функціонування в операційному режимі? Поясніть механізм розподілу Ідентифікаторів по замовченню. Яке правило наперед-визначеного статичного зв'язування PDO в мережі?

8. Які режими ініціації обміну об'єктів PDO є в CANOpen? Поясніть їх функціонування? Як забезпечується синхронізація передачі PDO в чітко визначені моменти часу? Який вузол генерує синхронні об'єкти? Які комунікаційні режими передачі (Transmission Type) визначені в CANOpen? Прокоментуйте призначення кожного режиму. Де зберігається налаштування режиму для кожного PDO?
9. Яким чином в CANOpen вузол повідомляє про наявність помилки? Які сервіси використовуються для визначення станів NMT-Ведених та NMT-Ведучих вузлів на шині?
10. Які переваги дає профілювання пристроїв для використання їх в CANOpen? Які типи профілів пристроїв для CANOpen Ви можете назвати? Яким чином в Об'єкті Словнику вказується його приналежність до конкретного профілю?

## ЧАСТИНА 2.

**Тривалість:** 2 акад. години (1 пара).

**Мета:** Навчитись користуватися сервісами PDO, SDO та управляти частотними перетворювачами по профілю CiA 402.

### Лабораторна установка.

Лабораторна робота виконується на робочому місці 8 двома бригадами, інші дві бригади в цей час виконують першу частину лабораторної роботи №9. Лабораторна установка зображена на рис.9.1.

**Апаратне забезпечення.** В таблиці наведена специфікація мережних засобів. Опис роботи засобів та способи їх підключення наведені в додатках 9.1 – 9.3.

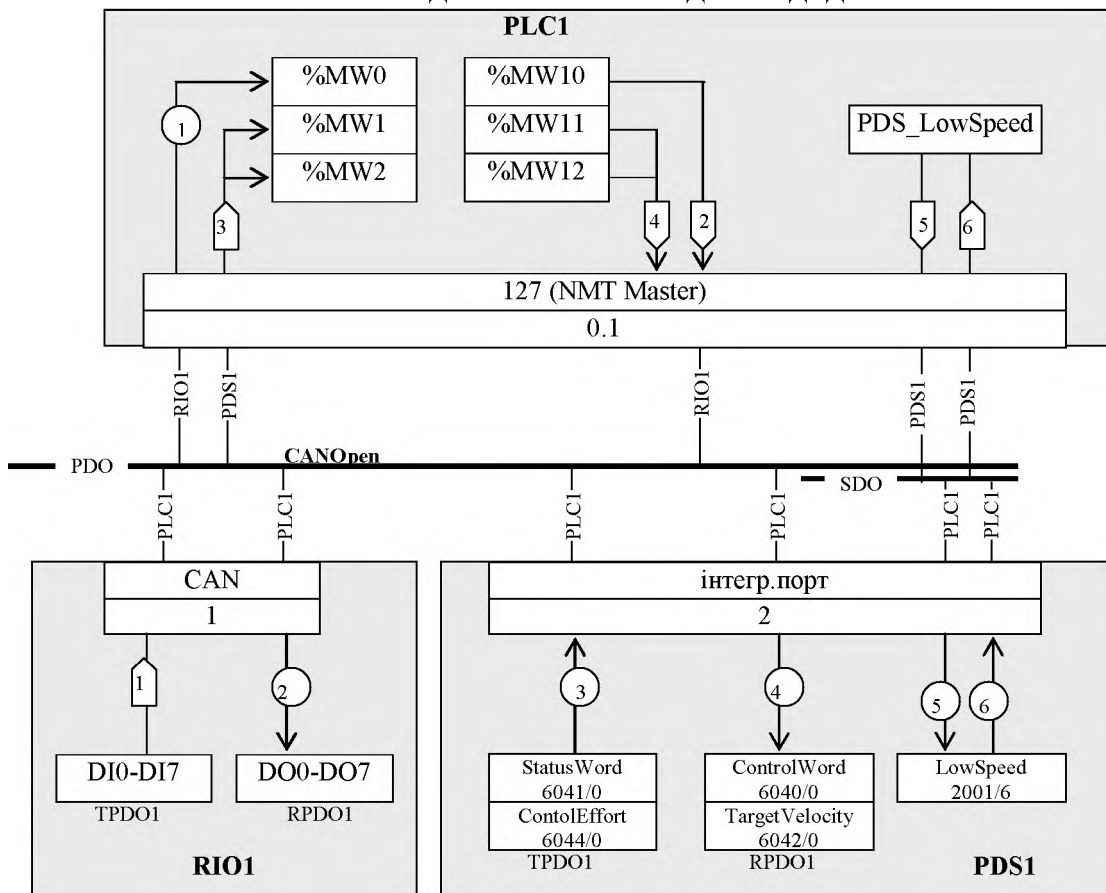


Рис.9.3. Схема мережних інформаційних потоків до лабораторної роботи

**Програмне забезпечення.** Для створення конфігурації для PLC1 використовується UNITY PRO, для створення конфігурації для TSX CPP 110 використовується SyCon.  
**Загальна постановка задачі.** Цілі роботи: створити конфігурацію мережі для TSX CPP 110 та PLC Premium відповідно до поставленої задачі, оформлену у вигляді схеми інформаційних потоків (рис 9.3); перевірити функціонування сервісів PDO та SDO; навчитись управляти частотним перетворювачем через сервіс CiA-402.

В даній роботі пристрої RIO1 та PDS1 вже сконфігуровані.

### **Послідовність виконання роботи.**

- 1) Запустити на ПК8 SyCon та створити конфігурацію мережі відповідно до схеми інформаційних потоків, що зображена на рис.9.3 (за винятком потоків 5 та 6, які оформляються через сервіс SDO). Бітова швидкість – 1 Мбіт/с, профіль RIO1 – тільки дискретні входи та дискретні виходи. Всі інші параметри, що не вказані на схемі інформаційних потоків та не уточнені в задачі залишити по замовченню. Зберегти файл конфігурації на диску під назвою "lab9\_2.co" але не закриваючи SyCon.
- 2) Завантажити UNITY PRO та створити конфігурацію ПЛК TSX 572634M (**OS2.0**), з комунікаційною картою TSX CPP 110. Налаштувати параметри комунікаційної карти відповідно до схеми інформаційних потоків та підключити файл "lab9\_2.co" в режимі SyCon. Створити конфігурацію для Ethernet відповідно до лабораторної роботи №4 (частина 2), та прив'язати її до інтегрованого в процесорний модуль каналу Ethernet (інакше UNITY PRO не скомпілює проект).
- 3) Створити або імпортувати секцію з програмою та змінними (див. Д9.6). Для імпорту виділити в Project Browser в розділі Program->Tasks->Mast->Sections->контекстне\_меню->Import..., вказати файл PKIS9\_SDO.XBD. Імпортувати змінні, для чого виділити в Project Browser в розділі Variables->контекстне\_меню->Import, вказати файл PKIS9\_CAN.XSY. Після імпорту необхідно перевірити створення змінних в розділі Variables & FB instances.

Імпортувати таблицю анімацій, для чого виділити в Project Browser в розділі Animation Tables->контекстне\_меню->Import, вказати файл PKIS9\_CAN.XTB.

Скомпілювати проект та завантажити його в контролер.

- 4) Завантажити із SyCon активну конфігурацію в карту TSX CPP110 (див. Д9.4.7). Після завантаження запустити ПЛК на виконання.
- 5) Необхідно перевірити роботу RIO1: стан індикаторів (CANRunn повинен світитися, CANERR не світитися); в таблиці анімацій послідовно змінювати змінні RIO1\_Outputs.0 до RIO1\_Outputs.5, при цьому слідкувати за індикаторами каналів модуля та змінними RIO1\_Inputs.0... RIO1\_Inputs.5. Якщо зворотній зв'язок присутній – перейти до пункту 6, якщо ні – до пункту 1.
- 6) Орієнтуючись на змінну стану частотного перетворювача (див. додаток 9.7) необхідно виставити задану частоту на 1500 об/хв. та запустити двигун. Перевірити індикатори на частотному перетворювачі та значення змінної дійсної швидкості.
- 7) Виставити задану частоту на 500 об/хв. Перевірити індикатори на частотному перетворювачі та значення змінної дійсної швидкості. Зупинити двигун.
- 8) Зчитати значення мінімальної частоти частотного перетворювача через сервіс SDO. Результат зчитування перевірити через змінну PARA. Виставити задану частоту рівною 0. Запустити двигун. Перевірити індикатори на частотному

перетворювачі та значення змінної дійсної швидкості. Результати проаналізувати та записати.

- 9) Використовуючи сервіс SDO записати значення мінімальної частоти частотного перетворювача на 1 Гц більше ніж було до того, але більше 5. Результат зчитування перевірити через змінну PARA1. Виставити задану частоту рівною 0. Перевірити індикатори на частотному перетворювачі та значення змінної дійсної швидкості. Результати проаналізувати та записати.
- 10) Використовуючи сервіс SDO записати значення мінімальної частоти частотного перетворювача рівною 5 Гц. Зупинити двигун.

### **Оформлення роботи.**

В оформлення роботи включається SDO-таблиця SyCon та копія сторінки конфігурації мережної карти TSX CPP 110.

### **Перевірка виконання роботи та питання до захисту.**

Викладачем перевіряється виконання поставленого завдання шляхом перевірки пунктів 5-10. Студент повинен розуміти роботу прикладної програми, яка використовується в даній роботі, функціонування сервісів PDO, SDO та вміти пояснити автомат станів частотного перетворювача згідно профілю CiA-402.

1. В чому необхідність знання структури Словнику Об'єктів конкретного засобу? Який ресурс необхідно використати для її визначення? Прокоментуйте загальноприйнятий розподіл Індексів різних типів Об'єктів в Словнику.
2. Що забезпечують функції сервісу SDO? Які комунікаційні Об'єкти використовуються для функціонування даного сервісу? На якій моделі прикладного обміну базуються сервіси SDO?
3. Порівняйте функціонування PDO та SDO сервісів. В яких випадках використовуються одні та інші?
4. Яким чином забезпечується обмін даними за допомогою об'єктів SDO? Як дані SDO-об'єктів зв'язуються з даними Прикладних Об'єктів Словнику? Як забезпечується передача даних великого обсягу?
5. Яким чином реалізований сервіс SDO в контролерах TSX Premium? Прокоментуйте комунікаційні функції, що використовуються в UNITY PRO для читання Об'єктів Словника через сервіс SDO.
6. Розкажіть про наявні профілі пристроїв PDS та їх відображення на промисловій мережі в стандартах MEK 61800-7?
7. Поясніть відмінність поняття профілю PDS від промислової мережі, що підтримуються даним пристроєм. Чи визначає факт використання конкретної промислової мережі наявність конкретного профілю PDS та навпаки?
8. Прокоментуйте функціонування профілю CiA402 в контексті функціональних елементів. Які основні відмінності даної реалізації від стандартної MEK 61800-7-1?
9. Як реалізуються в CiA402 змінні COMMAND та STATUS, визначені в MEK 61800-7-1? З якими мережами використовується даний профіль?
10. Прокоментуйте схему функціонування автомату станів базових функціональних елементів для профілю CiA402.
11. Прокоментуйте схему функціонування операційної роботи приводу в режимі управління швидкістю для профілю CiA402.



## Додаток 9.1. Використання та конфігурування карти TSX CPP 110.

**Д9.1.1. Загальні положення.** Комунікаційна карта TSX CPP110 (формат PCMCIA III) призначена для підключення ПЛК TSX Premium та TSX Micro до мереж CAN та CANOpen. Карта підтримує стандарт CANOpen - DS301 V4.01, а саме:

- підтримує бітову швидкість до 1Мбіт/с;
- може виконувати функції NMT-Ведучого;
- дає можливість проводити динамічне PDO-Відображення та PDO-Зв'язування;
- доступні функції "Heartbeat";
- може бути SYNC-Producer;
- Node-ID карти потрібен тільки для функцій "Heartbeat".

Карта також підтримує можливість безпосереднього обміну CAN-кадрами стандарту CAN V2.0B. Конфігурування карти проводиться утилітою SyCon  $\geq$ V2.8 (див. додаток 9.4).

**Д9.1.2. Підключення.** Карта TSX CPP110 поставляється разом з кабелем відгалуження, який в кінці має коробку підключення з вбудованим 9-піновим SUB-D роз'ємом (рис.9.4). Таке виконання дає можливість уніфікувати кабелі відгалуження. Підключення коробки TSX CPP110 до магістрального кабелю можна проводити з використанням аксесуарів сторонніх виробників (конектори, клеми, коробки), або через спеціальні 4-х портові коробки відгалуження SX CAN TDM4 (рис.9.5) з вбудованим термінатором лінії.

**Д9.1.3. Методика конфігурування карти.** Для створення конфігурації мережі для ПЛК TSX Micro/Premium, окрім програм PL7 PRO або Unity PRO, необхідне додаткове спеціалізоване ПЗ – SyCon  $V \geq 2.8$ . Порядок створення конфігурації наступний:

1. Створюється проект конфігурації мережі на SyCon та записується на диск у вигляді файлу з розширенням \*.CO;
2. Створюється конфігурація ПЛК TSX Premium, в якій для 1-го каналу процесорного модуля (або модуля SCY 21601) замовляється комунікаційна карта TSX CPP110;
3. Для комунікаційної карти вказується файл проекту мережі, створений за допомогою SyCon (рис. 9.6), налаштовуються діапазон змінних для відображення вхідних R-PDO та вихідних T-PDO комунікаційної карти.

Після конфігурування мережі та комунікаційної карти, конфігурація завантажується в ПЛК разом з проектом. Кількість конфігураційних даних для мережі CANOpen може бути дуже великою, що приведе до необхідності виділення значної пам'яті процесора. Тому передбачені два варіанти завантаження конфігураційних даних карти для TSX CPP110: "UNITY PRO" та "SyCon". Необхідний режим завантаження вибирається у вікні "Configuration load mode" (див.

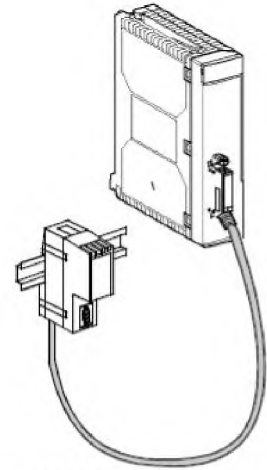


Рис.9.4. Зовнішній вигляд карти TSX CPP110 з відгалуженням



Рис.9.5. Коробка відгалуження TSX CAN TDM4 на 4 порти типу Sub-D 9-pin

рис 9.6). У випадку вибору режиму "SyCon", завантаження конфігураційних даних карти TSX CPP110 необхідно проводити засобами SyCon (див. додаток 9.4). У випадку режиму "UNITY PRO" конфігураційні дані TSX CPP110 будуть зберігатися та змінюватися разом з даними проекту (конфігурація ПЛК, програма, тощо). Режим SyCon є сенс використовувати тільки для зменшення пам'яті, що виділяється для конфігураційних даних. **В лабораторній роботі завжди вказуйте режим SyCon.**

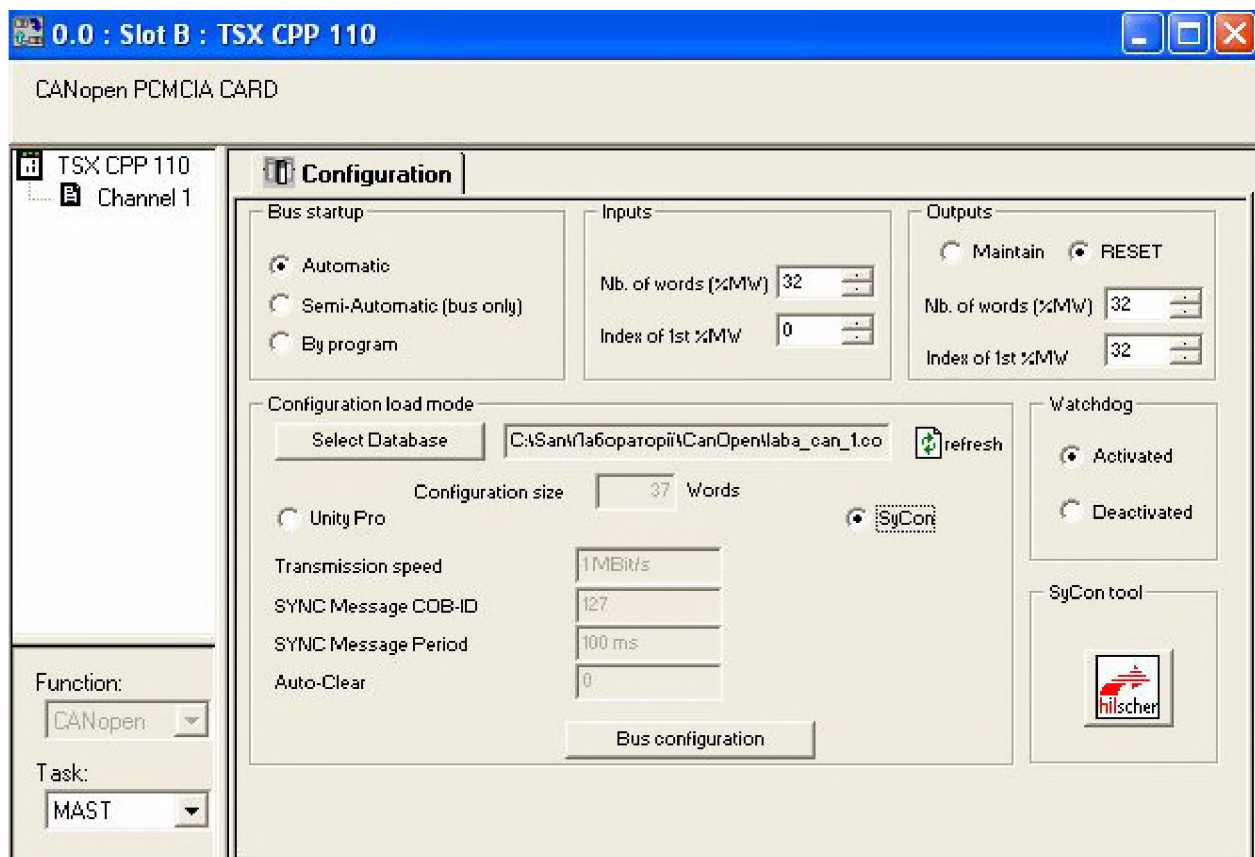


Рис.9.6. Конфігурація мережі в UNITY PRO.

## Додаток 9.2. Використання та конфігурування Advantys STB NCO 2212.

**Д9.2.1. Загальні положення.** Advantys STB – це сімейство наборної системи з модулів розподіленого вводу/виводу, живлення, інших типів модулів та пристроїв, які підключаються до промислової мережі, як один модульний пристрій. Таку систему також називають островом. Крім Advantys STB, Шнейдер Електрик пропонує острови Advantys OTB та Advantys FTB.

Підключення острову Advantys STB до промислової мережі проводиться через відповідний базовий мережний інтерфейсний модуль (NIM). Крім функцій обміну даними процесу через мережний порт, NIM також надає можливість конфігурування острову через додатковий порт CFG з використанням спеціального ПО – Advantys Configuration Software. Крім того, CFG порт дає можливість підключення до острову засобів SCADA/HMI



Рис.9.7. Комунікаційний модуль STB NCO 2212 для мережі CANOpen

по мережі Modbus RTU/ASCII. Використання порту CFG та Advantys Configuration Software в лабораторній роботі не розглядається.

Для мережі CANOpen використовується NIM модуль STB NCO 2212 (рис.9.7). Даний модуль характеризується такими властивостями:

- профіль по замовченню DSP-401;
- максимальна кількість Об'єктів Словнику: 32 T-PDO та 32 R-PDO; 512 device-specific Об'єктів та 512 manufacturer-specific Об'єктів;
- розподіл Об'єктів в Словнику та PDO-Відображення по замовченню відповідно до DSP-401: R-PDO-1 для 64 Дискретних виходів, R-PDO-2 (додатково R-PDO-3 та R-PDO-4) для 12 аналогових виходів, T-PDO-1 для 64 дискретних входів, T-PDO-2 (додатково T-PDO-3 та T-PDO-4) для 12 аналогових входів;
- бітова швидкість до 1 Мбіт/с;

**Д9.2.2. Підключення.** Підключення модуля STB NCO 2212 до магістрального кабелю виконується через 9-піновий SUB-D конектор, що стандартизований CANOpen. В якості коробки відгалуження можна використати TSX CAN TDM4.

**Д9.2.3. Налаштування швидкості та адреси.** Налаштування швидкості та адреси для NCO 2212 проводиться за допомогою вбудованих перемикачів (рис.9.8). Для налаштування бітової швидкості, перед увімкненням живлення засобу, нижній перемикач переключають в позицію BAUD RATE а верхній – в позицію потрібної швидкості: 0 (10 кБіт/с), 1 (20 кбіт/с), 2 (50 кбіт/с), 3 (125 кбіт/с), 4 (250 кбіт/с), 5 (500 кбіт/с), 6 (800 кбіт/с), 7 (1 Мбіт/с). Після включення живлення, Advantys запам'ятає вказану швидкість.

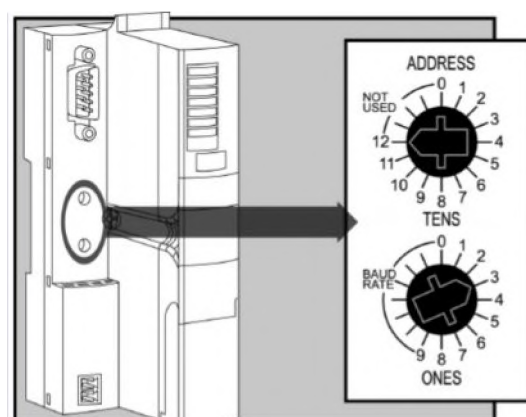


Рис.9.8. Налаштування адреси та бітової швидкості для Advantys STB.

Адреса пристрою (1-127) задається комбінацією позицій обох перемикачів: верхній – десятки, нижній – одиниці.

**Д9.2.4. Об'єкти словнику та PDO-Відображення.** Розподіл Об'єктів Словнику в області Device-Profile Area проводиться згідно профілю DSP-401:

- Індекс  $6000_{16}$ : дискретні входи згруповані 8-бітними блоками; Під-Індекси 1-32 ( $20_{16}$ ) відповідають за номер 8-бітного блоку; дискретні входи рахуються зліва-направо по їх розміщенню в острові;
- Індекс  $6200_{16}$ : дискретні виходи згруповані 8-бітними блоками; Під-Індекси 1-32 ( $20_{16}$ ) відповідають за номер 8-бітного блоку; дискретні виходи рахуються зліва-направо по їх розміщенню в острові;
- Індекс  $6401_{16}$ : 16-бітні аналогові входи; Під-Індекси 1-32 ( $20_{16}$ ) відповідають за номер каналу аналогового входу; аналогові входи рахуються зліва-направо по їх розміщенню в острові;
- Індекс  $6411_{16}$ : 16-бітні аналогові виходи; Під-Індекси 1-32 ( $20_{16}$ ) відповідають за номер каналу аналогового виходу; аналогові виходи рахуються зліва-направо по їх розміщенню в острові;

PDO-Відображення по замовченню, відповідає мінімальним вимогам профілю DSP-401 для 2-х R-PDO та 2-х T-PDO, а також додатково включає по 2 R-PDO та 2

T-PDO. Таким чином Відображення та комунікаційні параметри PDO по замовченню, має наступний вигляд:

PDO	Відображення та комунікаційні параметри PDO
R-PDO-1	Об'єкти Індекс 6200, Під-індекси 1...8 (64 дискретних виходів), асинхронно
R-PDO-2	Об'єкти Індекс 6411, Під-індекси 1...4 (4 16-бітових аналогових виходів), асинхронно
R-PDO-3	Об'єкти Індекс 6411, Під-індекси 5...8 (4 16-бітових аналогових виходів), асинхронно
R-PDO-4	Об'єкти Індекс 6411, Під-індекси 9...12 (4 16-бітових аналогових виходів), асинхронно
T-PDO-1	Об'єкти Індекс 6000, Під-індекси 1...8 (64 дискретних входів), передача асинхронно по зміні значення
T-PDO-2	Об'єкти Індекс 6401, Під-індекси 1...4 (4 16-бітових аналогових входів), асинхронно по зміні значення
T-PDO-3	Об'єкти Індекс 6401, Під-індекси 5...8 (4 16-бітових аналогових входів), асинхронно по зміні значення
T-PDO-4	Об'єкти Індекс 6401, Під-індекси 9...12 (4 16-бітових аналогових входів), асинхронно по зміні значення

### Додаток 9.3. Використання та конфігурування CANOpen для Altivar ATV 71.

**Д9.3.1. Загальні положення.** Частотні перетворювачі (надалі ЧПП) Altivar 71 мають вбудований порт Modbus/CANOpen з типом роз'єму RJ-45. Даний порт призначений для конфігурування ATV 71, управління та контролю його роботи як по мережі Modbus RTU так і по CANOpen. Режим роботи порту визначається конфігураційними настройками ЧПП, які проводяться через спеціальну панель, що прикріплена на його фронтальній стороні. В режимі CANOpen ЧПП підтримує профіль DSP-402 (профіль для пристроїв PDS), який крім розподілу Об'єктів в словнику та PDO-Відображення по замовченню, визначає послідовність управління самим PDS. Детальніше управління пристроями PDS з використанням профілю DSP-402 розглядається в додатку 9.7.

**Д9.3.2. Підключення.** Один із варіантів підключення до магістральної шини проводиться з використанням спеціальної коробки відгалуження VW3 CAN TAP 2 (рис.9.9). Ця коробка має 2 порти RJ-45 для підключення частотних перетворювача (ATV1 та ATV2) за допомогою кабелю VW3 CA RR 1 та порт для підключення ПК (PowerSuite). Останній дозволяє в режимі роботи системи по Modbus RTU налаштовувати частотні перетворювачі за допомогою спеціального конфігураційного забезпечення PowerSuite (в лабораторній роботі дане ПЗ не використовується). Крім того, дана коробка має вбудований термінатор лінії.

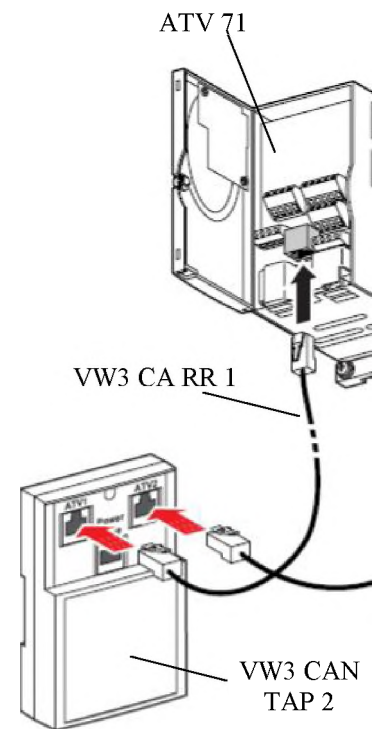


Рис.9.9. Підключення ATV71 до коробки відгалуження VW3 CAN TAP 2

### Д9.3.3. Налаштування частотного перетворювача для роботи з CANOpen.

Конфігурування частотного перетворювача може проводитись з використанням панелі настройки, параметри якої відображені в таблиці:

1.DriveMenu -> 1.9.Communication -> CANopen	[CANopen address]	адреса вузла
	[CANopen bit rate]	бітова швидкість
1.DriveMenu -> 1.6.COMMAND -> CANopen	[Profile]	профіль управління PDS
	[Ref. 1 channel]	джерело завдання швидкості/частоти (для завдання частоти з мережі CANOpen, повинен бути вибраний рівним "CANOpen"), додатково див. Д9.7
	[Cmd channel 1]	настройка 1-го каналу управління на джерело команди управління (для управління PDS з CANOpen повинен бути вибраний рівним "CANOpen"), додатково див. Д9.7
	[Cmd switching]	перемикач каналів управління (для настройки каналу 1 виставляється в режим "chn1 active"), додатково див. Д9.7

### Додаток 9.4. Робота з SyCon.

**Д9.4.1. Загальні положення.** Програмний пакет SyCon (від компанії Hilscher) призначений для конфігурування промислових мереж PROFIBUS, InterBus, CANOpen, DeviceNet, ControlNet, SDS, AS-Interface і т.д. пристроїв Hilscher. За допомогою цієї програми також конфігуруються комунікаційні модулі та карти PROFIBUS, InterBus та CANOpen для TSX Micro/Premium від Шнейдер Електрик.

В лабораторній роботі SyCon використовується для конфігурування CANOpen карти TSX CPP110. Створення проекту та його завантаження в CPP110 має наступну послідовність:

- 1) В SyCon створюється проект (File->New) з вибором мережі CANOpen.
- 2) Добавляються потрібні \*.EDS файли для пристроїв (File->Copy EDS), якщо вони відсутні в бібліотеці SyCon (в лабораторній роботі всі необхідні пристрої вже є в бібліотеці);
- 3) В конфігурації шини вставляється вузол NMT Ведучого (Insert->Master-> вибрати TSX CPP 110, нажати Add), виставляють необхідну адресу (Node ID) та назву (Description).
- 4) В конфігурації шини вставляються необхідні вузли NMT-Ведених (Insert->Node-> вибрати місце вставки-> лівий клік -> вибрати необхідний пристрій, нажати Add), виставляють необхідну адресу (Node ID) та назву (Description).
- 5) Для NMT-Ведених вузлів конфігуруються PDO (див. Д9.4.2).
- 6) Для NMT-Ведених вузлів конфігурується Node BOOT-UP, тобто поведінка NMT-Ведучого з вибраним вузлом при його включенні (див. Д9.4.3).
- 7) Для NMT-Ведених вузлів конфігуруються параметри контролю за помилками (Error Control); (див. Д9.4.4).
- 8) Для NMT-Ведених вузлів, при необхідності, конфігуруються значення Об'єктів Словнику, які необхідно змінити при старті. (див. Д9.4.5).
- 9) Конфігурується NMT-Ведучий та загальні параметри роботи шини. (див. Д9.4.6).
- 10) Створена конфігурація зберігається на диску (для підключення до проекту UNITY PRO) та завантажується в карту TSX CPP 110 через ПЛК (див. Д9.4.7).

**Д9.4.2. Конфігурування PDO.** Виклик конфігураційного вікна (рис.9.10) для конфігурування NMT-Ведених вузлів проводиться по подвійному кліку по його зображенню.

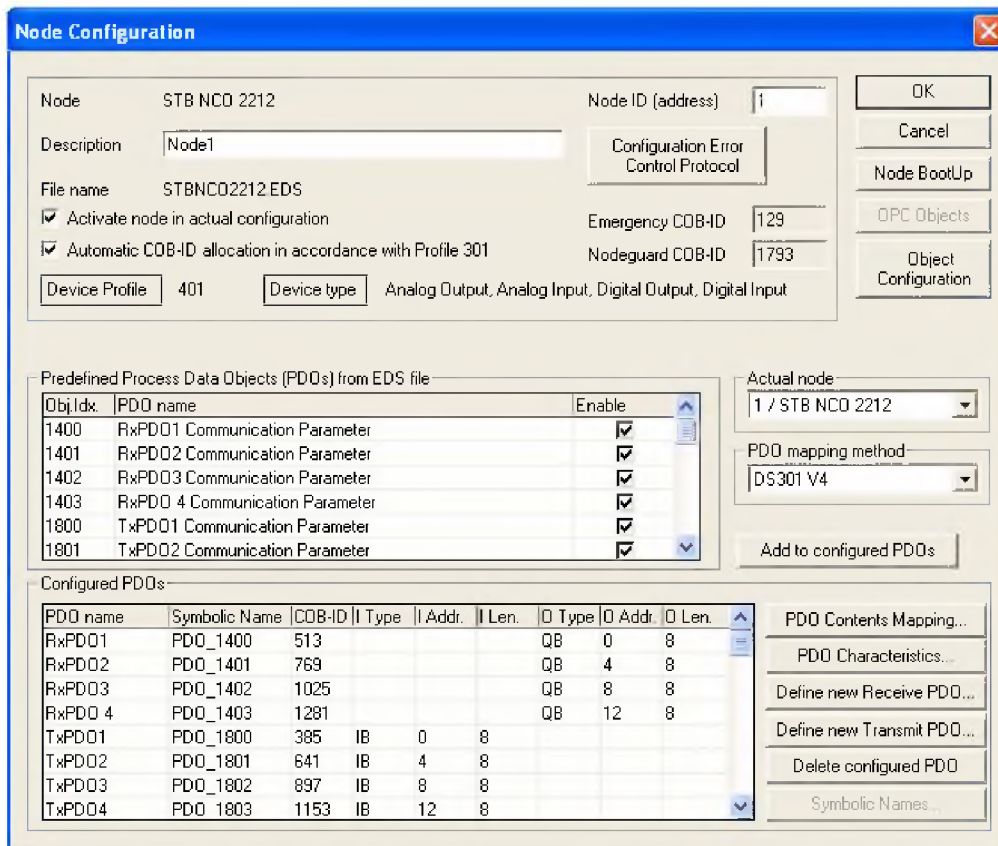


Рис.9.10. Вигляд вікна конфігурації NMT-Веденого вузла в SyCon.

Зміна профілю пристрою проводиться через кнопку "Device Profile" а уточнення типу через "Device Type".

По замовченню NMT-Ведучий обмінюється з NMT-Веденим. Однак можлива конфігурація, коли NMT-Ведений не приймає участь в обміні PDO з даним NMT-Веденим, а тільки конфігурує та управляє цим вузлом. Участь даного вузла в обміні з NMT-Ведучим конфігурується опцією "Activate node in actual configuration".

По замовченню всі PDO NMT-Ведених пов'язані з PDO NMT-Ведучими по наперед визначеній схемі

ідентифікації, тобто всі PDO отримують наперед визначені COB-ID, на які налаштований NMT-Ведучий. Переключення між ручним та автоматичним завданням COB-ID для PDO виставляється в "Automatic COB-ID alloaction in accordance with Profile 301" (рис.9.10).

Список наперед визначених PDO відображається у вікні для NMT-Веденого доступний у вікні "Predefined PDOs from EDS file", а їх активація/деактивація проводиться опцією "Enable".

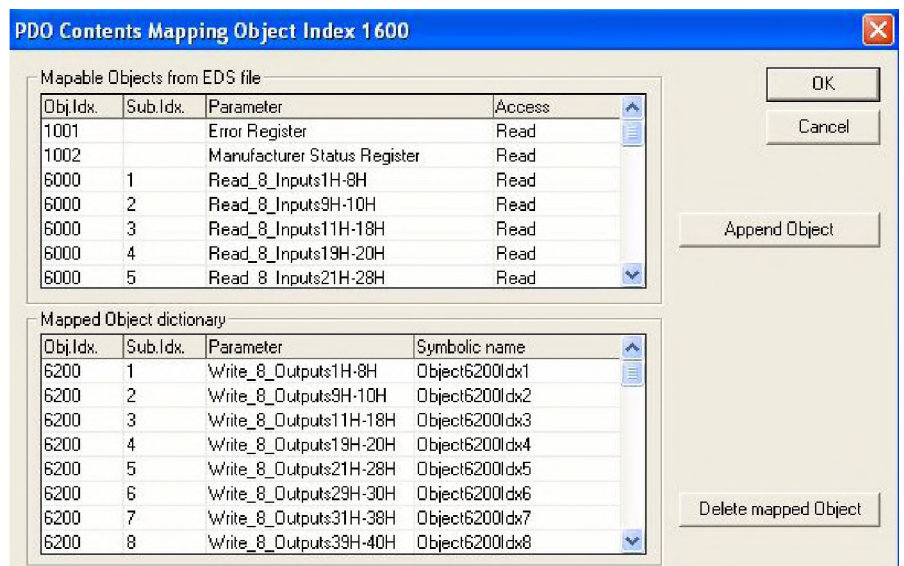


Рис.9.11. Вигляд вікна настройки PDO-Відображення.

Активні для обміну PDO відображаються в списку "Configured PDOs". Подвійний клік по вибраним активним PDO (або кнопка PDO Contents Mapping...) приводить до виклику вікна настройки PDO-Відображення (рис.9.11.). Доступні для PDO-Відображення Об'єкти Словнику показуються у списку "Mapable Objects from EDS file", подвійний клік по якому (або команда "Appened Object") додає його в список Відображених об'єктів даного PDO. Перелік Відображених на PDO Об'єктів вказаний у списку "Mapped Object dictionary".

Вікно налаштування комунікаційних параметрів обміну PDO викликається кнопкою "PDO Characteristics" (рис.9.10). Для R-PDO (рис.9.12) та T-PDO (рис.9.13) вікно налаштування комунікаційних параметрів відрізняється.

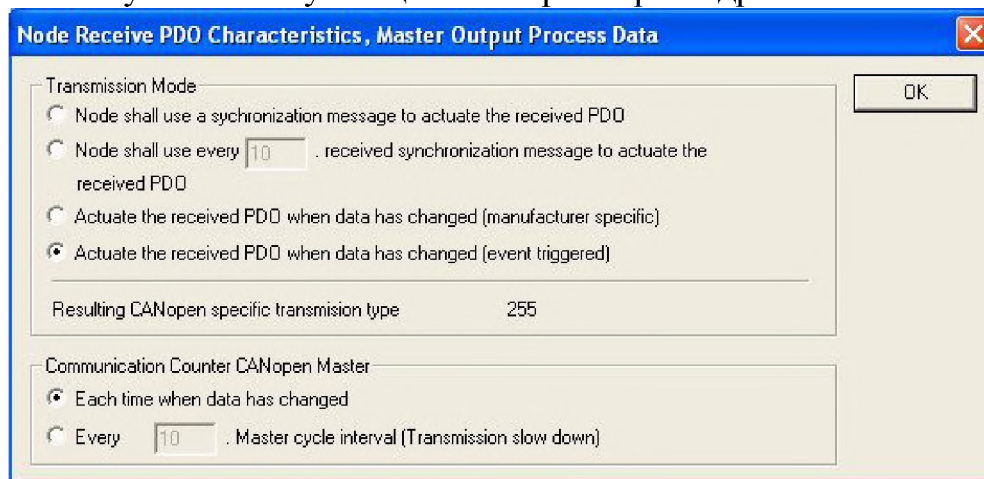


Рис.9.12. Вигляд вікна настройки характеристики R-PDO.

Налаштування комунікаційних параметрів обміну R-PDO для NMT-Веденого це по суті налаштування відповідного йому T-PDO для NMT-Ведучого. В наступній таблиці наведені комунікаційні параметри для R-PDO.

#### Комунікаційні параметри для R-PDO.

	Transmission Type	Тип ініціації передачі PDO NMT-Ведучим
Node shall use a synchronization message to actuate the received PDO	0	PDO буде передаватись ациклічно при події визначеній в "Communication counter CANOpen Master", але тільки в момент отримання Об'єкту SYNC
Node shall use every X received synchronization message to actuate the received PDO	1-240 (X)	PDO передаються синхронно з періодом, який визначається кількістю Об'єктів SYNC між передачами, вказану в значенні X (тип PDO).
Actuate the received PDO when data has changed (manufacturer specific)	254	момент відправки в мережу PDO не залежить від появи Об'єкту SYNC, і ініціюється подією визначеною в "Communication counter CANOpen Master"
Actuate the received PDO when data has changed (event triggered)	255	момент відправки в мережу PDO не залежить від появи Об'єкту SYNC і визначається профілем пристрою

Внизу вікна настройки (рис.9.12) в "Communication counter CANOpen Master" вказується подія ініціації обміну для ациклічних операцій:

- Each time when data has changed – кожен раз при зміні значення;
- Every Y Master cycle interval – кожні Y циклів NMT-Ведучого (час, що потрібний для внутрішньої обробки та оновлення значень всіх PDO)

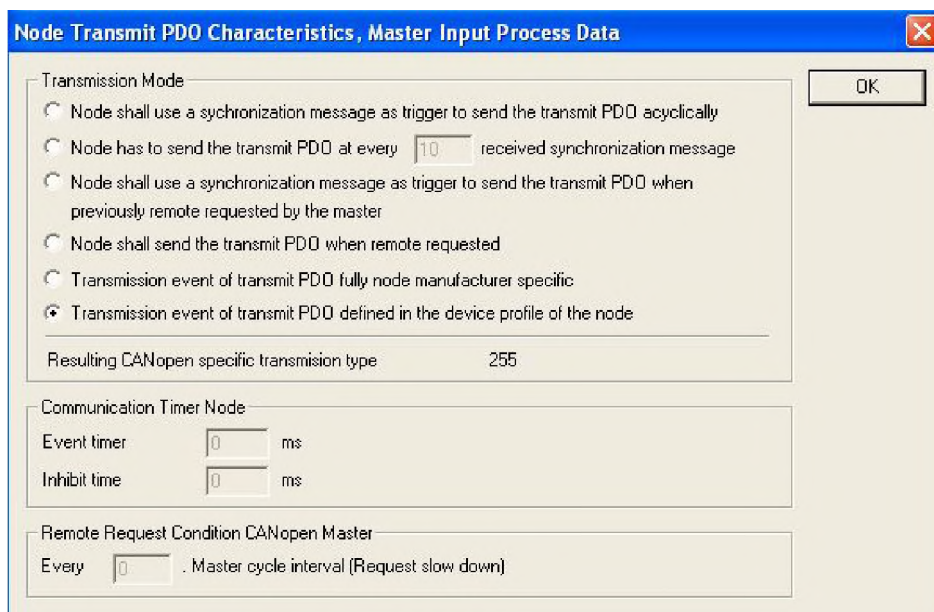


Рис.9.13. Вигляд вікна настройки характеристики T-PDO.

Налаштування комунікаційних параметрів обміну T-PDO проводиться згідно наступної таблиці.

#### Комунікаційні параметри для T-PDO

	Transmission Type	Тип ініціації передачі T-PDO NMT-Веденим
Node shall use a synchronization message as trigger to send the transmit PDO acyclically	0	PDO буде передаватись ациклічно при зміні значення, але тільки в момент отримання Об'єкту SYNC
Node has to send the transmit PDO at every (X) received synchronization message	1-240 (X)	PDO передаються синхронно з періодом, який визначається кількістю Об'єктів SYNC між передачами, вказану в значенні X (тип PDO).
Node shall use a synchronization message as trigger to send the transmit PDO when previously remote requested by the master	252	в момент отримання Об'єкту SYNC вхідні дані T-PDO будуть оцифровані і збережені в T-PDO, а відправлений T-PDO відбудеться в момент отримання запиту Дистанційного Кадру (RTR)
Node shall send the transmit PDO when remote requested	253	T-PDO відновляється і відправляється при отриманні Дистанційного Кадру (RTR)
Transmission event of transmit PDO fully node manufacturer specific	254	момент відправки в мережу PDO не залежить від появи Об'єкту SYNC і визначається виробником пристрою
Transmission event of transmit PDO defined in the device profile of the node	255	момент відправки в мережу PDO не залежить від появи Об'єкту SYNC і визначається профілем пристрою

Для T-PDO типу 254 та 255 можна визначити інтервал часу між відправками T-PDO. Цей параметр доступний не для всіх типів пристроїв і настраюється в області "Communication Timer Node" в полі "Event timer" (рис.9.13). Враховуючи, що Відображені Об'єкти на PDO типів 254 та 255 можуть оновлюватися дуже часто, передача PDO з високим пріоритетом може витіснити інші PDO. Для запобігання цього ефекту в області "Communication Timer Node" визначене поле "Inhibit Time" (Час Заборони), яке вказує на мінімальний час між двома передачами даного PDO.

Інтервал відправки Дистанційного Кадру (RTR) NMT-Ведучим для T-PDO типів 252 та 253 настраюється в полі "Remote Request CANopen Master".

Перелік всіх PDO на шині доступний через меню "View->ID Table" або "View->Address Table".



### Д9.4.3. Конфігурування поведінки NMT-Веденого.

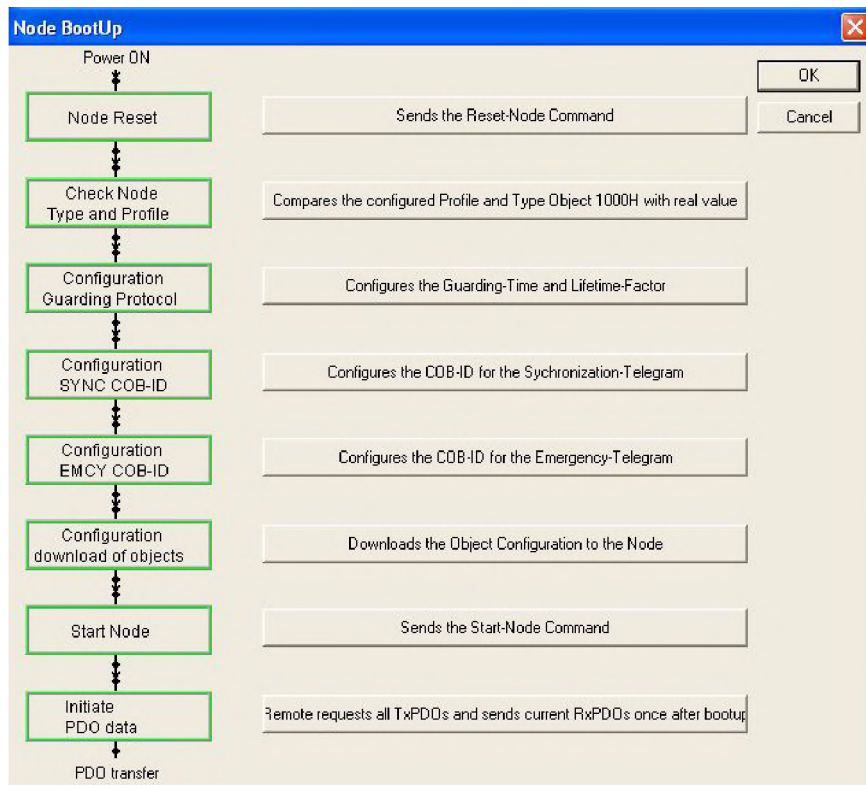


Рис.9.14. Вікно налаштування управління NMT-Ведучим NMT-Веденим при старті .

Node Reset	при активній опції, при старті NMT-Ведучого, він відправляє даному вузлу команду Node Reset
Check Node Type and Profile	при активній опції, NMT-Ведучий порівнює дійсний профіль пристрою з вказаним в конфігурації, якщо вони не співпадуть буде згенерована помилка параметризації
Configuration Guarding Protocol	при активній опції, NMT-Ведучий записує в пристрій параметри NodeGuard та Life-Time
Configuration SYNC COB-ID	при активній опції, NMT-Ведучий записує в пристрій значення COB-ID об'єкту SYNC
Configuration EMCY COB-ID	при активній опції, NMT-Ведучий записує в пристрій значення COB-ID об'єкту EMCY
Configuration download of objects	при активній опції, NMT-Ведучий записує в пристрій значення конфігураційних Об'єктів для PDO-обміну
Start Node	при активній опції NMT-Ведучий відправляє команду Start для переводу його в операційний режим
Initiate PDO data	при активній опції NMT-Ведучий проводить обмін даними PDO з даним вузлом

#### Д9.4.4. Конфігурування параметрів контролю за помилками NMT-Веденого.

Параметри контролю за помилками NMT-Веденого конфігуруються у відповідному вікні, через кнопку "Configuration Error Control Protocol" (рис.9.10). В залежності від можливостей вузла вибирається один з двох протоколів Node Guard або Heartbeat (рис.9.15).

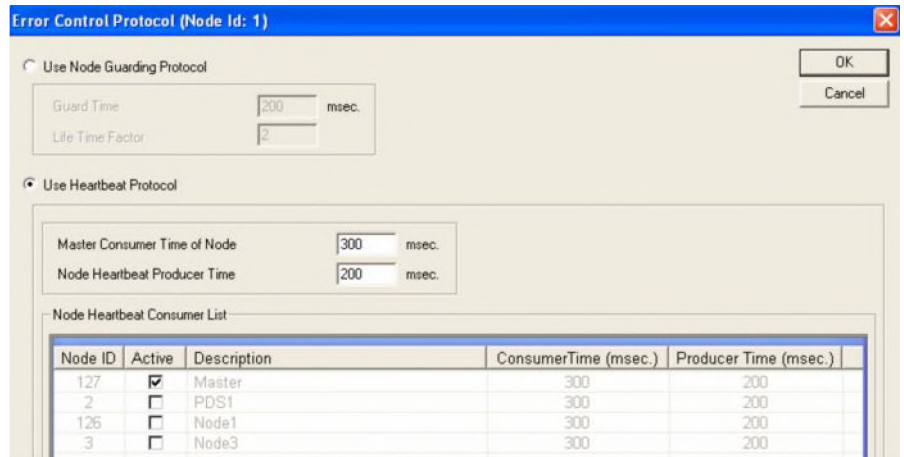


Рис.9.15. Вікно налаштування параметрів контролю за помилками.

Д9.4.5. Конфігурація початкових значень Об'єктів Словнику через сервіс SDO. Об'єкти Словнику, які необхідно змінити при старті вузла, можна продивитись та змінити шляхом виклику вікна "Object Configuration" відповідною кнопкою у вікні конфігурації вузла (рис.9.10). У вікні Object Configuration (рис.9.16) в списку "Predefined supported Objects in the EDS file" знаходяться доступні Об'єкти Словнику відповідно до підключеного EDS файлу. Перелік Об'єктів, що будуть змінюватися при BootUp процедурі запуску вузла сервісом SDO, та їх значень вказуються у списку "Configured Objects automatically written while Node startup sequence". Там розміщуються ті Об'єкти, які призначені для конфігурування PDO (відмічені PDO Dialog), а також туди можна додати будь який інший Об'єкт, значення якого може бути змінений (доступ Read/Write).

Перелік всіх Об'єктів на шині які змінюються при старті за допомогою SDO доступний через меню "View->SDO Table".

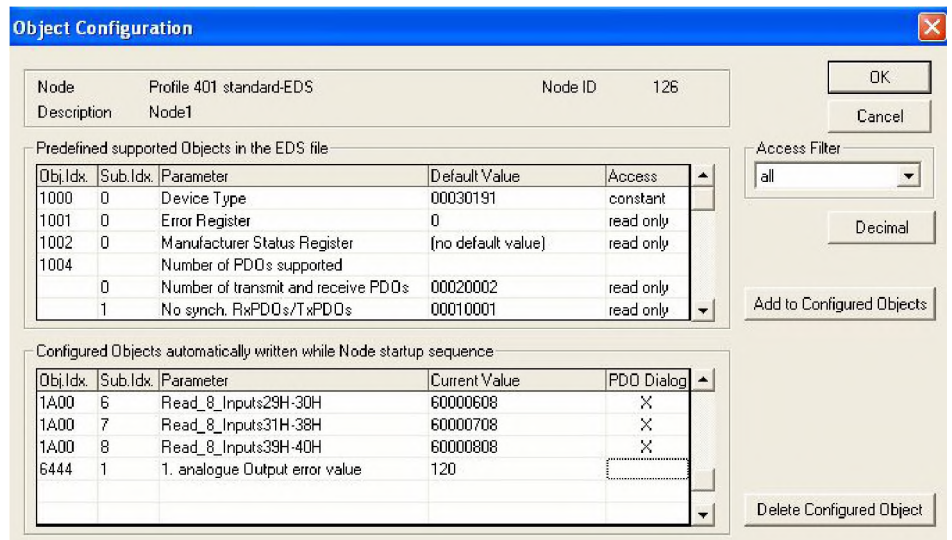


Рис.9.16. Вікно налаштування початкових значень Об'єктів Словнику через SDO.

Д9.4.6. Конфігурація загальних параметрів роботи шини. Вікно конфігурації загальних параметрів роботи шини викликається в меню "Settings->Bus Parameter" при активному курсорі на зображенні NMT-Ведучого. В цьому вікні (рис.9.17) настроюється бітова швидкість, поведінка NMT-Ведучого при виявленні помилки NMT-Веденого (зупинка комунікаційного обміну при виявленні помилки на будь

якому вузлі), COB-ID ідентифікатор та частота появи об'єкту SYNC, активація Heartbeat та його настройки.

#### Д9.4.7. Завантаження конфігурації в карту TSX CPP 110 через ПЛК.

Завантаження конфігурації SyCon в комунікаційну карту проводиться через ПЛК. В лабораторній роботі зв'язок з ПЛК проводиться через драйвер UNITELWAY. Для вибору та налаштування цієї комунікації необхідно в меню Settings->Device Assigment вибрати XWAY, далі вказати драйвер UNTLW01 і нажати Ок для підтвердження.

Для відправки даних в карту вибирається Online->Download.

**Контролер при цьому повинен знаходитись в стані STOP.**

Рис.9.17. Вікно налаштування загальний параметрів роботи шини.

### Додаток 9.5. Таблиці для оформлення частини 1 лабораторної роботи.

Таблиця T9.1.SDO Table

Варіант 1 (робоче місце 2).					
No de ID	Obj. Idx	Sub. Idx	Parameter	CurValue (HEX)	коментар
	1005	0	COB-ID Sync	80	
	1006	0	Communication Cycle Period	64	
1	1800	1	COB-ID	181	
		2	Transmission type	14	
		3	Inhibit time	0	
	1A00	0	Number of mapped objects	1	
		1	8bit Input Block No. 1	60000108	
2	1400	1	COB-ID	202	
		2	Transmission type	FF	
	1600	0	Number of mapped objects	1	
		1	Controlword	60400010	
	1800	1	COB-ID	182	
		2	Transmission type	FD	
		3	Inhibit time	0	

		5	Event timer	0	
	1801	1	COB-ID	282	
		2	Transmission type	FF	
		3	Inhibit time	12C	
		5	Event timer	1F4	
	1A00	0	Number of mapped objects	4	
		1	Statusword	60410010	
		2	Control Effort	60440010	
		3	Frequency ref.	20020410	
		4	Motor current	20020510	
	1A01	0	Number of mapped objects	3	
		1	Motor torque	20020610	
		2	Mains voltage	20020810	
		3	Motor voltage	20020910	
	2000	13	VMAX	10	
<b>Варіант 2 (робоче місце 3).</b>					
	1005	0	COB-ID Sync	80	
	1006	0	Communication Cycle Period	C8	
2	1400	1	COB-ID	202	
		2	Transmission type	FF	
	1600	0	Number of mapped objects	1	
		1	8bit Output Block No. 1	62000108	
	1800	1	COB-ID	182	
		2	Transmission type	FD	
		3	Inhibit time	0	
	1A00	0	Number of mapped objects	1	
		1	8bit Input Block No. 1	60000108	
	1017	0	Producer Heartbeat Time	C8	
	1016	1	Consumer Heartbeat Time	7F012C	
	3000	0	1 byte special output object No. 1	FF	
3	1016	1	Consumer Heartbeat Time	7F012C	
	1017	0	Producer Heartbeat Time	C8	
	1400	1	COB-ID	203	
		2	Transmission type	0	
	1600	0	Number of mapped objects	1	
		1	Target Velocity	60420010	
	1800	1	COB-ID	183	
		2	Transmission type	FF	
		3	Inhibit time	12C	
		5	Event timer	7D0	
	1801	1	COB-ID	283	
		2	Transmission type	14	
		3	Inhibit time	0	
		5	Event timer	0	
	1A00	0	Number of mapped objects	2	
		1	Statusword	60410010	
		2	Control Effort	60440010	
	1A01	0	Number of mapped objects	3	
		1	Frequency ref.	20020410	
		2	Motor current	20020510	
		3	Motor torque	20020610	
	2000	14	FMAX	5	

<b>Варіант 3 (робоче місце 4).</b>				
	1005	0	COB-ID Sync	80
	1006	0	Communication Cycle Period	FA
3	1400	1	COB-ID	203
		2	Transmission type	0
	1600	0	Number of mapped objects	1
		1	8bit Output Block No. 1	62000108
	1800	1	COB-ID	183
		2	Transmission type	0
		3	Inhibit time	0
	1A00	0	Number of mapped objects	1
		1	8bit Input Block No. 1	60000108
	3001	0	1 byte special output object No. 2	FF
4	1400	1	COB-ID	204
		2	Transmission type	14
	1600	0	Number of mapped objects	2
		1	Controlword	60400010
		2	Target Velocity	60420010
	1800	1	COB-ID	184
		2	Transmission type	FD
		3	Inhibit time	0
		5	Event timer	0
	1801	1	COB-ID	284
		2	Transmission type	FF
		3	Inhibit time	12C
		5	Event timer	3E8
	1A00	0	Number of mapped objects	4
		1	Frequency ref.	20020410
		2	Statusword	60410010
		3	Motor current	20020510
		4	Motor torque	20020610
	1A01	0	Number of mapped objects	2
		1	Main Voltage	20020810
		2	Motor Voltage	20020910
<b>Варіант 4 (робоче місце 5).</b>				
	1005	0	COB-ID Sync	80
	1006	0	Communication Cycle Period	1F4
4	1016	1	Consumer Heartbeat Time	7F012C
	1400	1	COB-ID	204
		2	Transmission type	14
	1600	0	Number of mapped objects	1
		1	8bit Output Block No. 1	62000108
	1800	1	COB-ID	184
		2	Transmission type	FD
		3	Inhibit time	0
	1A00	0	Number of mapped objects	1
		1	8bit Input Block No. 1	60000108
	1017	0	Producer Heartbeat Time	C8
5	1016	1	Consumer Heartbeat Time	7F012C
	1017	0	Producer Heartbeat Time	C8
	1400	1	COB-ID	205
		2	Transmission type	0

	1600	0	Number of mapped objects	2	
		1	Target Velocity	60420010	
		2	Controlword	60400010	
	1800	1	COB-ID	185	
		2	Transmission type	FF	
		3	Inhibit time	12C	
		5	Event timer	1F4	
	1A00	0	Number of mapped objects	4	
		1	Statusword	60410010	
		2	Control Effort	60440010	
		3	Frequency ref.	20020410	
		4	Motor current	20020510	

Таблиця Т9.2. Параметри настройки області пам'яті для для обміну PDO в TSX Premium.

Параметр	Значення для Inputs	Значення для Outputs
Index of 1st %MW		
Nb. of words (%MW)		

## Додаток 9.6. Використання в прикладній програмі ПЛК Premium сервісів SDO.

**Д9.6.1. Загальні положення.** Сервіси SDO в TSX Premium доступні через використання функцій читання/запису READ\_VAR та WRITE\_VAR. Функції описані в додатках до лабораторної роботи №3, тут розглянемо тільки особливості їх використання для SDO.

**Д9.6.2. Використання ADDR.** Для формування адреси в функціях READ\_VAR та WRITE\_VAR використовується функція ADDR, де формат поля адреси має наступний вигляд:

**'0.m.1.SYS'**

, де m – номер процесорного модуля в шасі (0 або 1).

**Д9.6.3. Використання READ\_VAR.** Для зчитування Об'єктів Словнику через сервіс SDO в TSX Premium використовується функція READ\_VAR з наступним синтаксисом в форматі ST:

READ_VAR(Address, 'SDO', index:subindex, NodeID, Management_Param, Receiving Array);		
Address	ARRAY [0.. 5] OF <a href="#">INT</a>	Адреса NMT-ведучого на PCMCIA карті TSX CPP110, яка задається функцією ADDR (див.Д.9.6.1)
index:subindex	<a href="#">DINT</a>	адреса Об'єкту в Словнику, в молодшому слові – Індекс, в старшому – Під індекс, наприклад 16#00010002 вказує на Об'єкт з Індексом 0002 <sub>16</sub> та Підіндексом 0001 <sub>16</sub>
NodeID	<a href="#">INT</a>	адреса вузла на CANOpen, до якого йде звернення
Management_Param	ARRAY [0.. 3] OF <a href="#">INT</a>	Таблиця управління обміном (див.Д.4.1.6)
Receiving_Array	ARRAY [n.. m] OF <a href="#">INT</a>	Масив, який вміщує значення прочитаних змінних

**Д9.6.4. Використання WRITE\_VAR.** Для запису Об'єктів Словнику через сервіс SDO в TSX Premium використовується функція WRITE\_VAR з наступним синтаксисом в форматі ST:

WRITE_VAR (Address, 'SDO', index:subindex, NodeID, Data_to_Write, Management_Param);		
Address	ARRAY [0.. 5] OF <a href="#">INT</a>	Адреса NMT-ведучого на PCMCIA карті TSX CPP110, яка задається функцією ADDR (див.Д.9.6.1)
index:subindex	<a href="#">DINT</a>	адреса Об'єкту в Словнику, в молодшому слові – Індекс, в старшому – Під індекс, наприклад 16#00010002 вказує на Об'єкт з Індексом 0002 <sub>16</sub> та Підіндексом 0001 <sub>16</sub>
NodeID	<a href="#">INT</a>	адреса вузла на CANOpen, до якого йде звернення
Data_to_Write	ARRAY [n.. m] OF <a href="#">INT</a>	Масив, який вміщує значення даних, які треба записати
Management_Param	ARRAY [0.. 3] OF <a href="#">INT</a>	Таблиця управління обміном (див.Д.4.1.6)

Перед викликом функції WRITE\_VAR в останнє слово таблиці Management\_Param необхідно записати кількість байт, які треба передати через SDO.

**Д9.6.5. Використання WRITE\_VAR. Варіант програми для читання та запису значень об'єктів через сервіс SDO.** Приклад програми для читання та запису через SDO наведений на рис.9.18. Дискретні змінні READ\_SDO та WRITE\_SDO використовуються для ініціювання відповідно зчитування та запису, після чого значення цих змінних обнуляються. Перед викликом функції WRITE\_VAR в третє (останнє) слово таблиці що управляє обміном (PARA1[3]) записується кількість байт що передаються – два байта (ціле значення).

В лабораторній роботі сервіс SDO використовується для зчитування та запису конфігураційної настройки частотного перетворювача, яка відповідає за мінімальну частоту обертання (Index=2001<sub>16</sub>, SubIndex=6<sub>16</sub>). Дане значення зберігається в змінній ПЛК PDS\_LowSpeed.

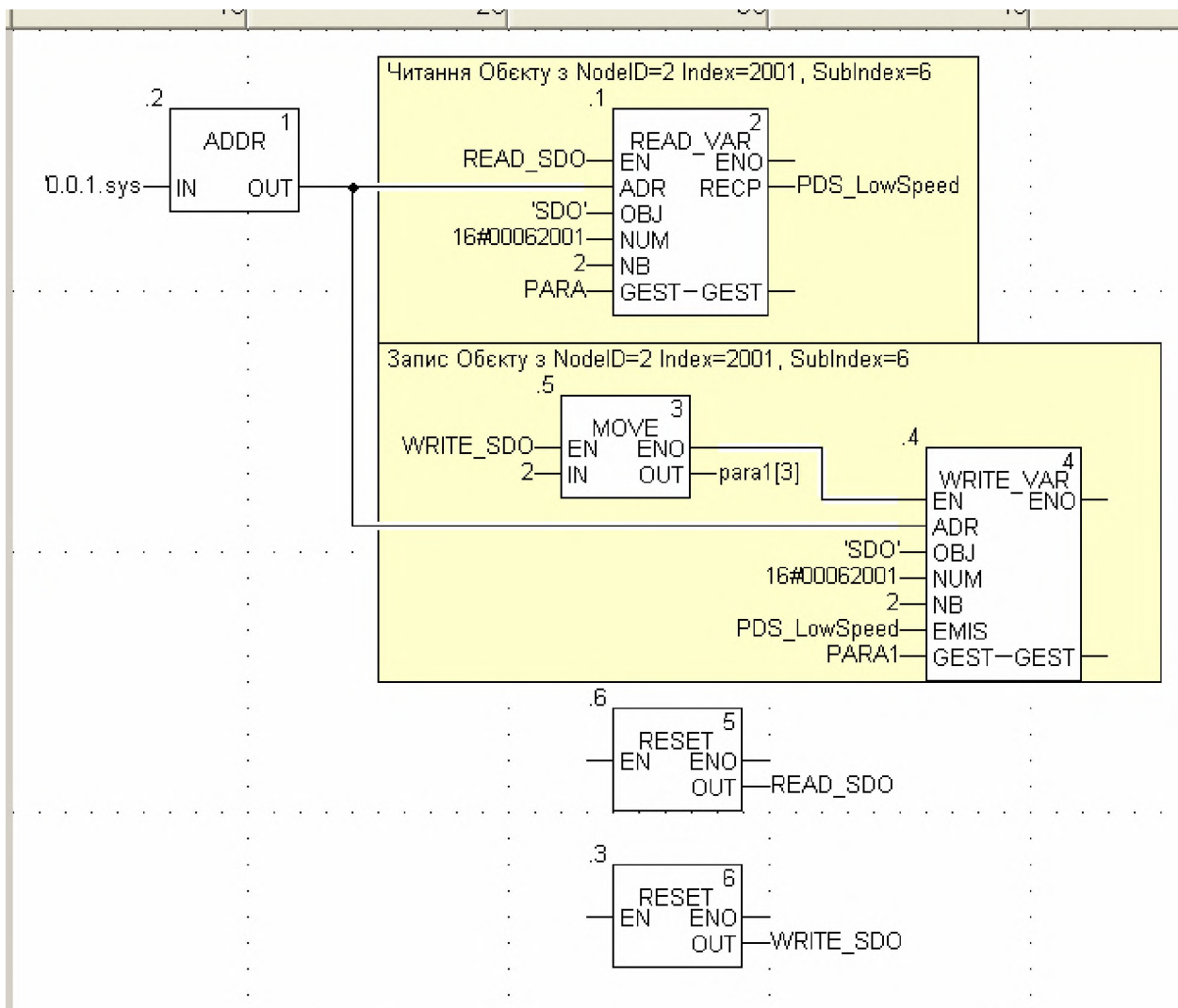


Рис.9.18. Прикладна програма для зчитування та запису SDO.

### Додаток 9.7. Управління частотними перетворювачами з використанням профілю CiA-402.

Поведінка приводу PDS згідно профілю CiA-402 залежить від функціонального профілю пристрою. В лабораторній роботі використовується профіль управління швидкістю. Згідно цього профілю автомат станів частотного перетворювача має вигляд, як на рис.9.19.



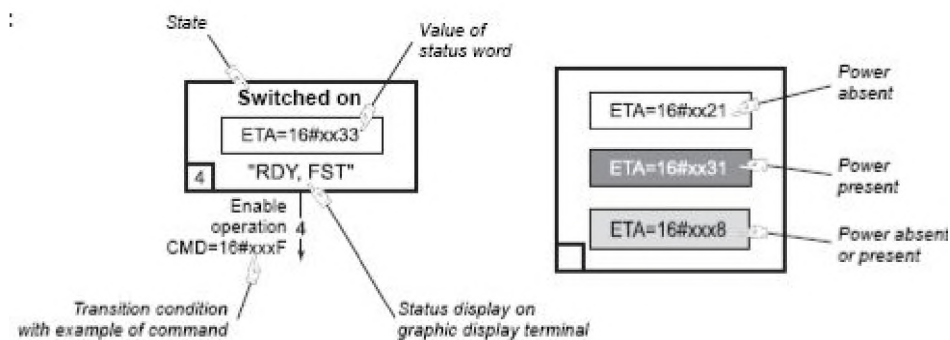
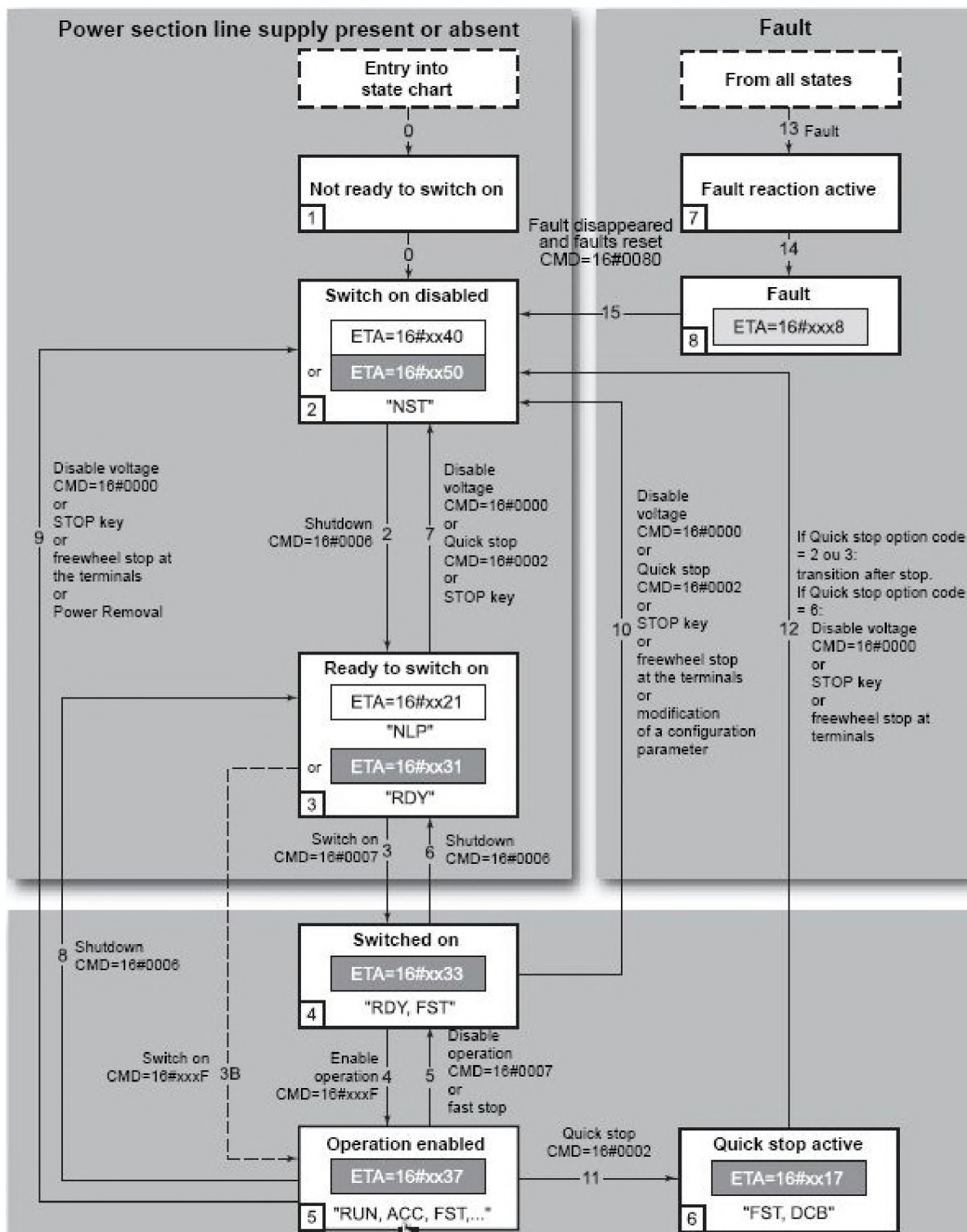


Рис.9.19. Автомат станів управління ЧПР по профілю управління швидкістю згідно СіА 402.

### **Список літератури.**

1. Пупена О.М., Ельперін І.В., Луцька Н.М., Ладанюк А.П. Промислові мережі та інтеграційні технології в автоматизованих системах: Навчальний посібник. – К.: Вид-во "Ліра-К", 2011. – 552 с.
2. Ельперін І.В. Промислові контролери: Навч.посіб. – К.: НУХТ, 2003. – 320 с.
3. [www.schneider-electric.com.ua](http://www.schneider-electric.com.ua)

Лабораторні роботи №5 "Використання шини UniTelway" та №8 "Використання Ethernet UNI-TE TCP/IP" призначені для факультативного вивчення за бажанням студентів. Протоколи робіт в електронному вигляді доступні на сервері кафедри інтегрованих автоматизованих систем управління в аудиторії А-532.