

МЕХАНІЗМ ПІД'ЄДНАННЯ ЗОВНІШНІХ КОМПОНЕНТІВ В УНІВЕРСАЛЬНІЙ SCADA-СИСТЕМІ TRACE MODE 5.15

© Демида Б., 2009

Розглянуто метод під'єднання користувацьких зовнішніх модулів у SCADA-систему з метою розширення можливостей математичного апарату та реалізації нестандартних графічних елементів. Показані переваги застосування ActiveX контролера на прикладі прикладної задачі розрахунку, побудови та відображення векторних діаграм струмів та напруг в автоматизованій системі диспетчерського управління електричною підстанцією із застосуванням SCADA-системи TRACE MODE 5.15.

The methods of handling internal data of SCADA systems for purpose of improving their functionality and customizing them are analyzed in the paper. Advantages of using ActiveX technology are shown by the example of applied problem of vector diagrams construction. Algorithm of vector diagram construction and mechanisms of interaction between developed ActiveX controls and SCADA Trace Mode 5.15 are described.

Вступ

У сучасних АСУ ТП людина-диспетчер має можливість отримувати інформацію про стани віддалених керованих об'єктів та змінювати ці стани зі свого робочого місця. Така можливість забезпечується SCADA-системами (Supervisory Control And Data Acquisition) – спеціалізованими програмами, що організують інтерфейс між диспетчером та системами управління.

Сучасні SCADA-системи ґрунтуються на таких основних концептуальних положеннях [1–3]:

- використання стандартних протоколів взаємодії і відкритих програмних технологій;
- багаторівневість побудови системи;
- організація спільного інформаційного простору;
- взаємодія з існуючими системами.

Прийнято вважати, що система є відкритою, якщо для неї визначені й описані використовувані формати даних і процедурний інтерфейс, що дає змогу під'єднати до неї "зовнішні", незалежно розроблені компоненти. Опора на стандартизацію та відкриті програмні технології забезпечує використання *універсальних* SCADA-систем в різноманітних галузях промисловості, зменшуючи тим вартість проекту.

Розроблення власних програмних модулів в універсальній SCADA-системі

Перед розробниками систем автоматизації часто постає питання про створення власних (не передбачених у межах SCADA-системи) програмних модулів і введення їх до створюваної системи автоматизації.

Представлені на ринку SCADA-систем програмні продукти описуються рядом характеристик, котрі можна виділити у три групи показників:

- технічні характеристики;
- вартісні характеристики;
- експлуатаційні характеристики.

Хоча питання про відкритість системи є важливою характеристикою SCADA-систем, ця характеристика не входить до жодної з перерахованих вище груп. Фактично відкритість системи означає доступність специфікацій системних (у змісті SCADA) викликів, що реалізують той або інший системний сервіс. Це може бути й доступ до графічних функцій, функцій роботи з базами

даних, розширення алгоритмічних можливостей і т.д. Якщо SCADA-система є відкритою, тобто дає змогу визначати додаткові механізми обробки її внутрішніх даних, то можна доповнити її функціональність та адаптувати для розв'язання конкретних задач. Цей факт надає більшої свободи, більшої гнучкості вибору конкретної SCADA і у деяких випадках забезпечує економічну вигоду.

Універсальна SCADA підтримує взаємодію з «зовнішнім світом» за допомогою ([3, 4], рис. 1):

- обміну даними з контролерами і платами вводу/виводу за вбудованими і зовнішніми протоколами включаючи і технологію OPC;
- передача даних у деякий пакет програмування (наприклад, за механізмом ODBC), обробка даних у цьому пакеті та зворотна передача;
- обміну даними з зовнішніми ужитками з використанням механізмів DDE, OPC;
- реалізації нестандартних графічних елементів та під'єднання зовнішніх модулів обробки та обрахунку даних на клієнтській машині з використанням зовнішніх ActiveX-компонентів;
- підтримка зовнішніх програмних блоків мовою Сі у вбудованих в SCADA-системи мовах програмування.

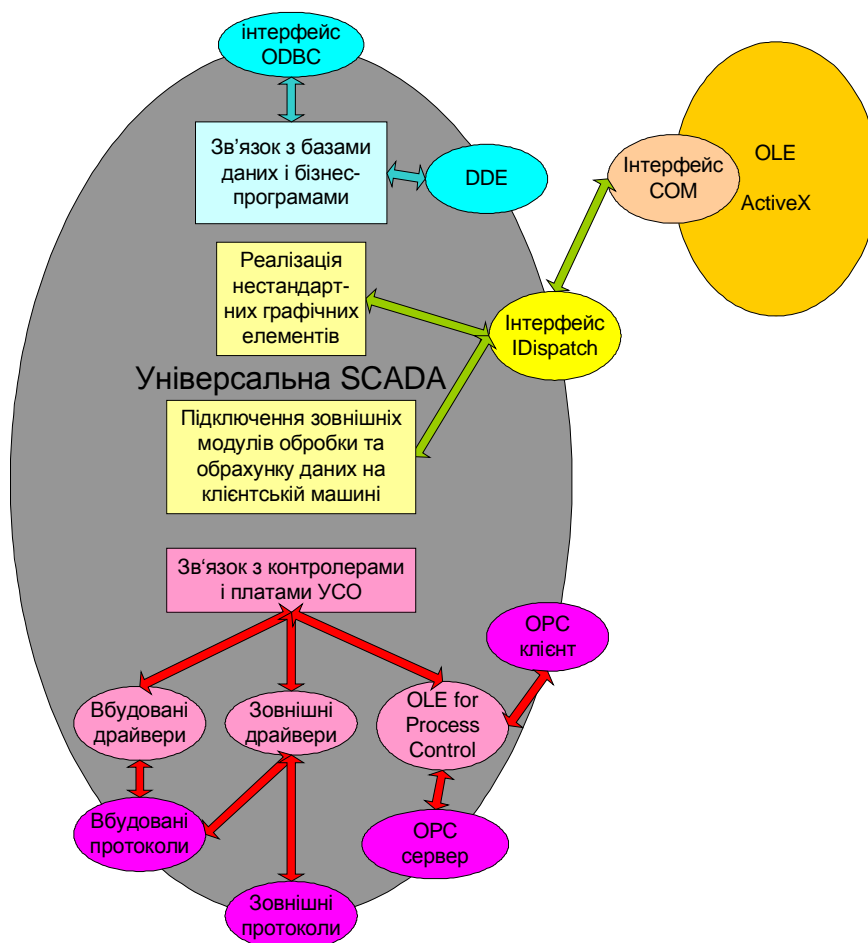


Рис. 1. Використання відкритих програмних технологій та взаємодія з існуючими системами в універсальній SCADA

Застосування пакетів програмування вимагає більше ресурсів і є більш трудомістким, оскільки написання програмного коду в цьому випадку ускладнюється описом обміну даними. Вбудовані мови програмування обмежуються організацією математичної обробки даних і не мають засобів для візуального представлення даних. Крім того, взагалі кажучи, різні SCADA-системи використовують різні мови програмування. Технологія ActiveX дає змогу впроваджувати у SCADA-систему об'єкти, що можуть зчитувати внутрішні дані системи, обробляти їх, реагувати на дії

користувача (введення даних за допомогою клавіатури чи миші), повертати результати обчислень самій системі та відображати дані у зручній наочній формі. Можливості об'єктів ActiveX в обчисленнях та візуальному представленні даних фактично необмежені.

Постановка задачі

Автоматизована система диспетчерського управління електричною підстанцією розроблена на основі програмного комплексу SCADA "Trace Mode 5.15" фірми AdAstraResearch та комплексу засобів обчислювальної, вимірювальної техніки і компонентів АСУТП провідних вітчизняних і закордонних виробників [2]. Для цієї АСУ необхідно розробити компоненту ActiveX, яка зчитує з каналів Трейс Моуд реальні вимірні значення струму, активної та реактивної потужності за кожною з трьох фаз. За зчитаними значеннями та відомими формулами обчислити значення напруги та зсуву фаз для кожної фази та побудувати векторну діаграму струмів та напруг у мнемосхемі АРМ диспетчера. Усі вхідні і результуючі дані також відобразити у табличному вигляді. Додатково (на вимогу диспетчера) повідомляти у графічному вигляді про вихід зсуву фаз за допустимі межі. Потрібно забезпечити можливість редагування меж.

Побудова векторних діаграм струмів та напруг

При побудові векторних діаграм для трьох фаз для наочності використовуються три осі; кут між додатними напрямками кожної пари осей становить 120° . Напруга кожної з фаз А, В, С відкладається на відповідній осі в додатному напрямку. Між вектором струму фази і відповідною віссю є кут φ , що знаходиться зі співвідношень [5, 6]:

$$\begin{aligned}P &= UI \cos \varphi; \\ Q &= UI \sin \varphi; \text{ або} \\ \cos \varphi &= P/S; \sin \varphi = Q/S,\end{aligned}$$

де S – повна потужність; P – активна потужність; Q – реактивна потужність; I – діюче значення струму.

Для фаз В і С відносно декартової системи координат вектор відкладається від додатної півосі ординат на кути $120^\circ + \varphi$ і $240^\circ + \varphi$ відповідно.

На рис. 2 зображений приклад векторної діаграми.

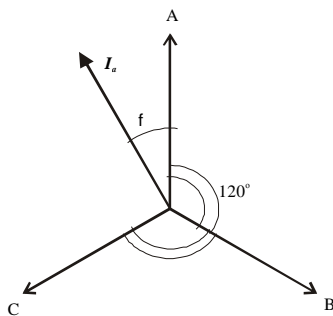


Рис. 2. Вигляд векторної діаграми струмів та напруг

Для відтворення значення, наприклад, вектора I (тобто, фізичного значення сили струму) на екрані, необхідно враховувати масштаб відображення діаграми струмів та напруг. Оскільки значення струмів заздалегідь невідомі, то масштабування слід здійснювати гнучко. В даній роботі масштабування проводиться так:

- вибирається число $\max I$ – максимальне зі зчитаних значень струмів трьох фаз;
- приймається, що величину $\max I$ деяких одиниць струму зображають на екрані n пікселів;
- якщо $\max I \neq 0$, то відношення $n/\max I$ показує, скільки пікселів на екрані зображується одиниця сили струму.

За описаним підходом екранна довжина жодного вектора струму не перевищує n пікселів, а отже, зображуваний вектор завжди розташований у видимій області екрана.

Аналогічно зображають значення напруги на екрані. При цьому програмний код допускає, що найбільший струм може стосуватися однієї фази, а найбільша напруга – іншої. Графік призначений для унаочнення зсувів фаз, а також відношень між струмами та відношень між напругами фаз.

У коді програми присутнє ділення на величини струму (при визначенні масштабу за струмом і визначенні напруги), напруги (при визначенні масштабу за напругою) і повної потужності (при визначенні косинуса та синуса зсуву фаз). При кожному діленні перевіряють на рівність дільника нулю. Це запобігає виникненню виняткової ситуації “ділення на нуль”. На рис. 3 зображено блок-схему програми обчислення та побудови векторної діаграми.

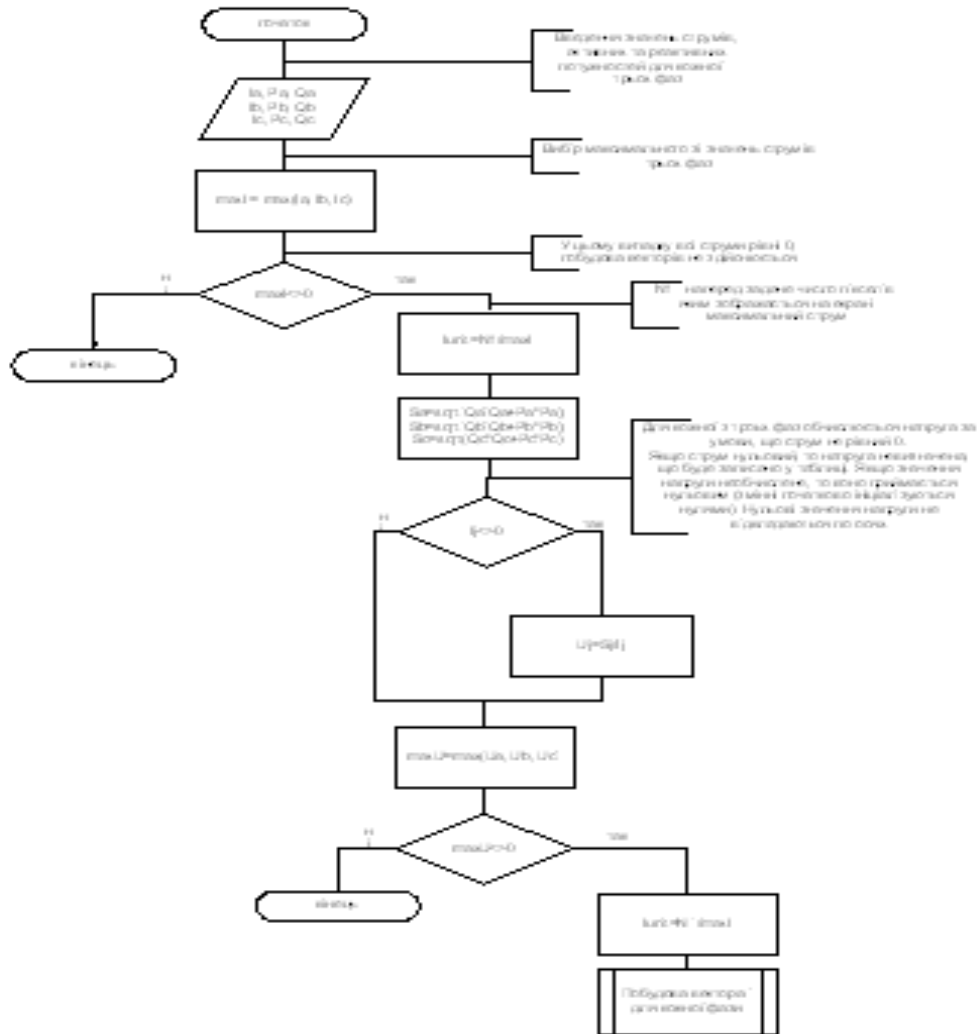


Рис. 3. Блок-схема програми обчислення та побудови векторної діаграми

Тестування роботи ActiveX компоненти

Тестуємо роботи компоненти за схемою:

1. Перевірку працездатності ActiveX компоненти для довільних вхідних даних:

$I \neq 0; P > 0; Q > 0;$

$I \neq 0; P > 0; Q < 0;$

$I \neq 0; P < 0; Q > 0;$

$I \neq 0; P < 0; Q < 0;$

$I \neq 0; P = 0$ при $Q > 0$ і при $Q < 0;$

$I \neq 0; Q = 0$ при $P > 0$ і при $P < 0;$

$I \neq 0; P = 0$ і $Q = 0;$ така комбінація не має фізичного змісту, проте вона може виникнути, якщо диспетчер з якихось причин не пов'яже змінні P і Q зі значеннями реальних каналів, проте асоціює

змінну I зі значенням відповідного каналу. У цьому випадку всі зчитані значення відображаються у таблиці, але не на графіку;

$I=0; P=0; Q=0;$

$I=0; P < 0; Q < 0$ – комбінація реально неможлива, але може бути результатом неправильного під'єднання каналів або значної нерівномірності розподілу сили струмів за фазами.

2. Перевірку коректності масштабування;
 3. Контроль правильності перевірки текстових полів;
 4. Контроль за виконанням уставок;
 5. Тестування правильності представлення чисел та заповнення таблиці.
- Результати тестування показано на рис. 4.

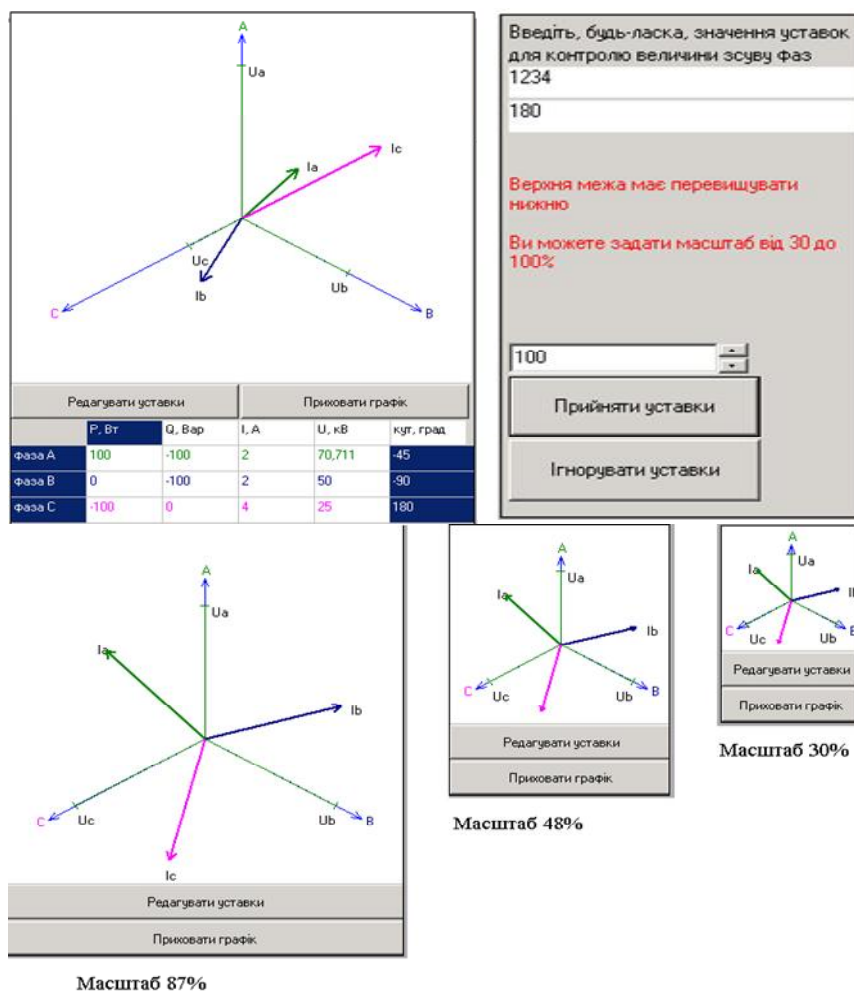


Рис. 4. Екранні форми результатів тестування компоненти

Взаємодія ActiveX компонентів із системою Трейс Моуд

Взаємодія ActiveX компонентів із системою Трейс Моуд здійснюється через інтерфейс *Idispatch*.

Інтерфейс *Idispatch* 4 методи:

- **Invoke** використовується для виклику усіх методів усіх дисп-інтерфейсів;
- **GetIDsOfNames** дає змогу клієнтові вказати ім'я методу дисп-інтерфейсу та отримати у відповідь відповідний *DISPID*;
- **GetTypeInfo** повертає вказівник інтерфейсу *ITypeInfo* об'єкта – інформації про тип, яка описує цей інтерфейс. Використовуючи методи *ITypeInfo*, клієнт може дізнатися про все, що потрібно для упаковки параметрів і виклику методів диспінтерфейсу;
- **GetTypeInfoCount** повертає інформацію про те, чи підтримує цей об'єкт видачу інформації типу в період виконання, тобто, чи буде виклик *GetTypeinfo* повертати корисну інформацію.

Керівну компоненту формують вбудованими засобами SCADA-системи. Можливі такі режими взаємодії Трейс Моуд та ActiveX-компонентів [8]:

- асоціювання з каналами;
- передача в ActiveX останнього повідомлення звіту тривоги, відправленого у графічну консоль;
- зміна значення каналу;
- отримання від МРЧ змінених значень каналів з часовими мітками.

У цьому режимі ActiveX може вказати: необхідне постійне поновлення чи тільки при його видимості.

ActiveX-компонент, який має інтерфейс *IDispatch*, при вставці та редагуванні властивостей компонента редактор представлення даних додає допоміжний бланк *Таблиця Властивостей* (див. рис. 5).

Для властивості, яка має числовий чи логічний метод зміни, потрібно в меню поля *Прив'язка* відповідного рядка таблиці вибрати **Вхід** чи **Вихід** компонента. У випадку вибору пункту **Вхід** значення передається у властивість з каналу, а у випадку вибору пункту **Вихід** – з властивості у канал. У зазначеному бланку виводиться таблиця властивостей компонента, причому лише ті властивості, які мають метод їх зміни. Метод може бути числовим, логічним чи текстовим. Після цього на екран виводиться діалог прив'язки до каналу, що має вигляд, як на рис. 6.

Властивості компоненти з доданими атрибутами **Вхід** чи **Вихід** можуть бути пов'язані з довільним числовим атрибутом каналів поточної бази у діалогі прив'язки до каналу (див. рис. 6).

За допомогою цього діалогу потрібно прив'язати властивість до каналу та його атрибута, а також встановити подію, за якою передають дані.

Трейс Моуд може застосовувати усі зареєстровані в системі компоненти ActiveX.

У Трейс Моуд включені форми відображення, оформлені як ActiveX-компоненти, що підтримують специфікацію TMX. До цих форм відображення відносяться таблиця перегляду значень каналу, перегляд звіту тривоги і менеджер рецептів.

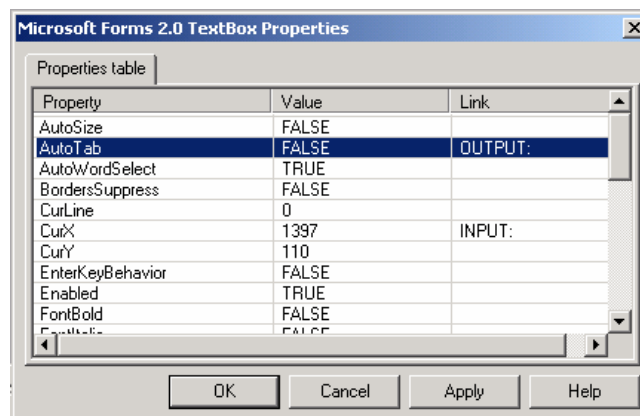


Рис. 5. Таблиця визначення властивостей компоненти ActiveX

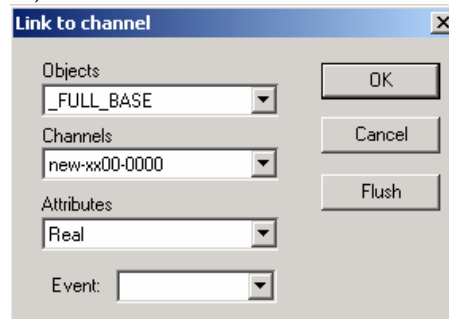


Рис. 6. Схема асоціювання з каналом

Приклад роботи розробленої компоненти, інтегрованої в АРМ диспетчера АСУ ПС „Промислова”, показаний на рис.7. На мнемосхемі диспетчерського управління підстанцією за допомогою

доданої кнопки «Діаграма» можна включити графічну форму та таблицю відображення значень напруги, струму та кута фаз між ними за кожним з фідерів. Як видно з рисунка, табличні значення підсвічені відповідним кольором залежно від перевищення відповідного порогу, котрий можна задати через меню «Налаштування», що загалом підвищує рівень сприйняття інформації диспетчером.

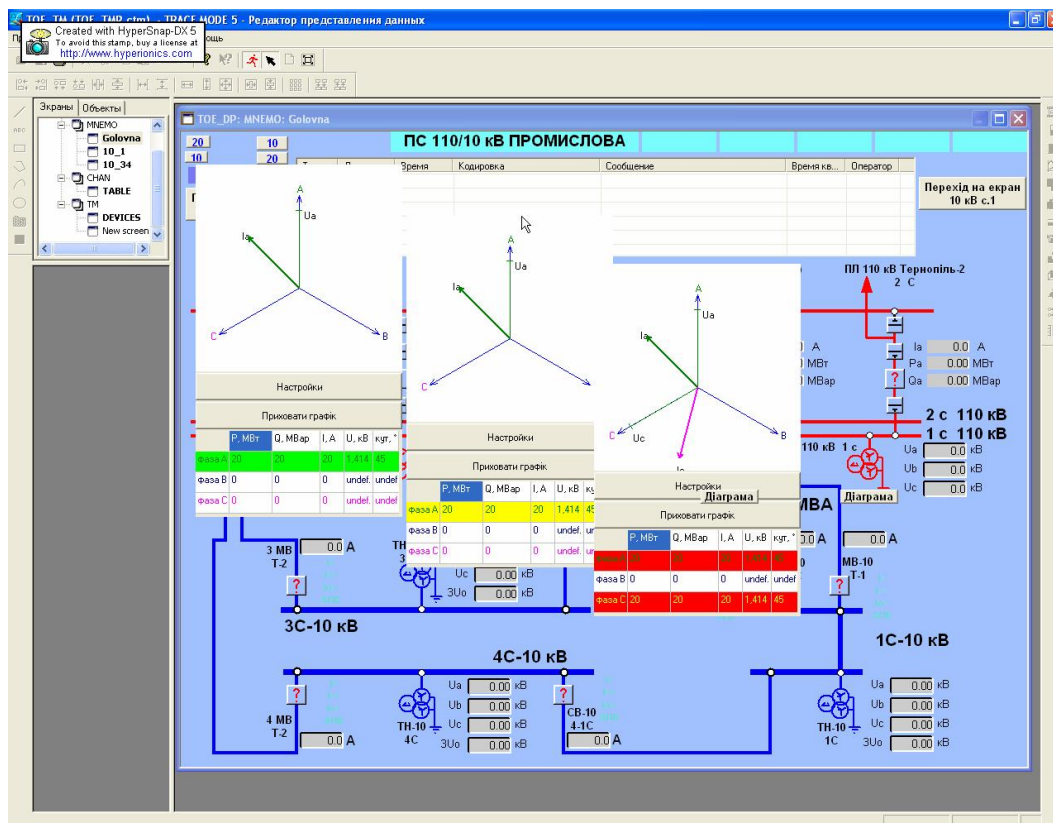


Рис. 7. Векторні діаграми струмів та напруг у мнемосхемі АРМ диспетчера АСУ ПС.

Висновки

У цій роботі на прикладі ActiveX-компонент, що будують векторні діаграми за реальними значеннями та відображають всю необхідну для диспетчера супутню інформацію, продемонстровані розширення функціональних можливостей універсальної SCADA-системи. Розроблена компонента універсальна, але була впроваджена і пройшла тестування у SCADA-системі Трейс Моуд 5.15.

Використання в універсальних SCADA-системах ActiveX-компонентів через стандартний COM інтерфейс дає змогу реалізувати нестандартні графічні елементи та під'єднати модулі користувача, які реалізують обробку даних та обчислення на клієнтській машині.

Отже, технологія ActiveX є оптимальним і перспективним інструментом розширення функціональності універсальних SCADA-систем.

1. Scada.ru – Публікації – SCADA-системи: погляд зсередини. 2. Деміда Б., Зербіно Д., Пелешко Д. Реалізація принципів паралельного збору інформації на прикладі SCADA-системи "TracEMode" // Вісник Нац. ун-ту "Львівська політехніка" "Комп'ютерні науки та інформаційні технології". – 2005. № 543. – С.3–10. 3. Кузнецов А. Genesis for Windows – графічна scada-система для розробки АСУ ТП // Сучасні технології автоматизації. – 1997. – №3. 4. ТРЕЙС МОУД – інтегрована SCADA- і softlogic-система для розробки АСУ ТП // URL: <http://adastra.ru/rw/tm/tm5/>. 5. Бронштейн И.Н., Семендяев К.А. Справочник по математике для инженеров и учащихся вузов. – М. Наука, 1965. – 608 с. 6. Зевеке Г.В., Ионкин П.А., Нетушил А.В. и Страхов С.В. Основы теории цепей. – М.: Энергия, 1965. 7. Чепелл Д. Технологии ActiveX и OLE / Пер. с англ. – М.: Издательский отдел "Русская редакция" ТОО "Channel Trading Ltd.", 1997. – 320 с. 8. Довідкова система програмного пакета Трейс Моуд 5.15.