

Лісовець С.М.

Таврійський національний університет імені В.І. Вернадського

ДОСТУП ДО OPC-СЕРВЕРІВ З СЕРЕДОВИЩА .NET ЗА ДОПОМОГОЮ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ADVOSOL

Сучасне промислове виробництво передбачає використання сучасних автоматизованих систем керування (систем автоматизації), які повинні не тільки забезпечувати оптимальне керування таким виробництвом, але й простий і ефективний контроль за ним зі сторони обслуговуючого персоналу. Такий контроль звичайно здійснюється за допомогою SCADA-систем – пакетів програм, які призначені для збору, обробки, відображення і архівування даних про об'єкт керування, а також виконання над такими даними певних додаткових дій в залежності від функціонального наповнення таких пакетів. Одним з «недоліків» представлених на ринку SCADA-систем є їхня надлишковість. Більша частина функціональних можливостей таких систем може ніколи не бути використана, при цьому обслуговування таких систем вимагає персоналу з високою кваліфікацією. Одним з існуючих підходів до контролю за виробництвом є власна розробка SCADA-систем з одночасним використанням одного або кількох готових OPC-серверів. Основною перешкодою такій розробці є значна складність безпосереднього програмування обміну даними з OPC-серверами. В такому випадку на допомогу «приходять» такі продукти, як, наприклад, програмні бібліотеки Advosol Inc., які дозволяють легко організувати обмін даними з OPC-серверами з середовища .NET. В проведеному дослідженні, результати якого наведено в матеріалах статті, на прикладі панелі оператора ИП320 з використанням мови програмування С# в середовищі програмування Visual Studio показана можливість обміну даними між такою панеллю і програмним забезпеченням на С# через OPC Server 320 з використанням бібліотек Advosol Inc. В даному випадку обмін здійснювався через інтерфейс RS-485 по протоколу Modbus RTU, причому сама панель по черзі працювала в режимах Master і Slave, а для її налаштування використовувалася програма-конфігуратор.

Ключові слова: OPC-сервер, SCADA-система, людино-машинний інтерфейс, програмна бібліотека, середовище програмування.

Постановка проблеми. Промислове виробництво у всьому світі, що особливо проявляється в розвинених країнах, знаходиться в постійному розвитку і вдосконаленні. Певна частина такого розвитку припадає на різні системи керування, і, зокрема, на автоматизовані системи. Такі системи вирішують дві основні задачі: керування виробництвом (виробничим процесом, виробничою операцією) і реалізація людино-машинного інтерфейсу. Людино-машинний інтерфейс, по суті, надає можливість людині в будь-який заданий момент часу розуміти, що відбувається на певному виробництві і за необхідності вносити в роботу такого виробництва потрібні коригування. Сучасні людино-машинні інтерфейси реалізуються за допомогою SCADA-систем – пакетів програм, які за допомогою різних текстових і графічних елементів в реальному часі відображають елементи виробництва так, щоб ця інформація могла бути швидко сприйнята і опрацьована. Зважаючи на те, що на сучасному виробництві одночасно можуть використовуватися кілька сотень

і навіть тисяч одиниць обладнання від різних виробників, які об'єднані в ті або інші промислові комп'ютерні мережі і які реалізують різні підходи до налаштування своєї роботи, реалізація для такого виробництва SCADA-систем представляє собою достатньо складну задачу [1].

Одним з напрямів реалізації SCADA-систем є використання OPC-серверів як проміжної ланки між обладнанням і людино-машинним інтерфейсом. Одна SCADA-система одночасно може використовувати кілька OPC-серверів, кожен з яких може бути оптимізованим для роботи з заданим обладнанням по певним комунікаційним інтерфейсам з використанням певних протоколів обміну даними. Наприклад, обладнання ТОВ «РАУТ-АВТОМАТИК» може використовувати власний протокол UNIVERS [2], а обладнання ТОВ «АКУТЕК» – власний протокол ОБЕН [3]. Крім того, представлені на ринку SCADA-системи часто мають дуже широкий функціонал, більша частина якого ніколи не буде використовуватися, і при цьому вони є достатньо складними

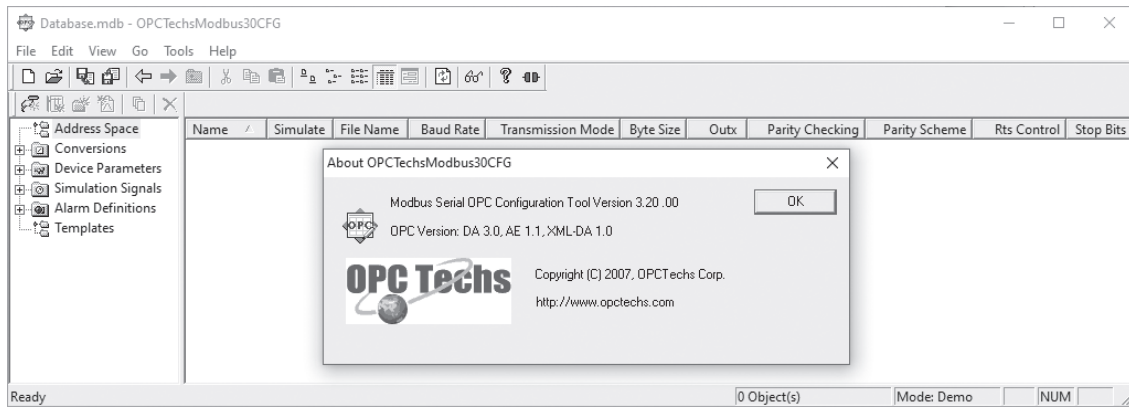


Рис. 1. Зовнішній вигляд конфігуратора OPC Server 320

і дорогими, а також вимогливими до апаратно-програмного забезпечення. Таким чином, якщо необхідно керувати нескладним виробничим процесом або операцією, доцільним може бути використання кількох OPC-серверів потрібних виробників обладнання (які звичайно розповсюджуються вільно і безкоштовно), а проектування і реалізацію SCADA-системи виконати самостійно.

Аналіз останніх досліджень і публікацій. Найбільш затребуваним на ринку OPC-стандартом, незважаючи на деякі недоліки і певну «застарілість», є OPC Data Access. Типовим OPC-сервером, який підтримує цей стандарт, є OPC Server 320 (див. рис. 1) [4]. Зокрема, він підтримує версії 1.0, 2.04, 2.05 і 3.0 цього стандарту.

Отримати доступ до даних таких серверів можна програмним шляхом. Основним недоліком такого доступу є те, що деякі з OPC-серверів (в тому числі і OPC Data Access) базуються на таких технологіях Microsoft Corp., як OLE, ActiveX і/або COM/DCOM та не є платформонезалежними.

Такі технології є достатньо складними і «заплутаними» у використанні, вимагають докладання великої кількості зусиль і можуть приводити до появи помилок, які складно виявити (внаслідок чого може суттєво зменшитися надійність обміну даними з OPC-серверами, що для промисловості може бути критичним). Крім того, вони фактично не розвиваються і замінюються іншими сучасними технологіями – зокрема, .NET. Відповідно, на теперішній момент часу на ринку існують різні розробники як платного, так і безкоштовного програмного забезпечення для підтримки роботи OPC-серверів з середовища .NET. Така підтримка полягає у використанні створених такими розробниками вже готових і багатократно перевірених

програмних бібліотек, які звичайно можна підключати до власного програмного забезпечення. Одним з таких розробників є компанія Advosol Inc., яка розпочала свою діяльність в 1996 р. як постачальник програмних компонентів і інструментів для роботи з OPC-технологіями (в тому числі і з OPC Server 320) [5]. Одним з засобів роботи з OPC-технологіями, які рекомендує ця компанія, є мова програмування C# в середовищі програмування Visual Studio.

Постановка завдання. Завданням виконаного дослідження була розробка на C# в Visual Studio програмного забезпечення (SCADA-системи з мінімальними можливостями) з використанням програмних бібліотек Advosol Inc. з метою отримання повного доступу до OPC-сервера OPC Server 320, а також експериментальна перевірка роботи такого програмного забезпечення з використанням панелі оператора ИП320 (див. рис. 2) [6, 7, 8].



Рис. 2. Зовнішній вигляд панелі оператора ИП320

Така панель представляє собою графічний рідинно-кристалічний дисплей з вісьмома керувальними та дванадцятьма цифровими і функціональними кнопками, яка може працювати через інтерфейс RS-485 по протоколу Modbus RTU в режимах Master або Slave.

Виклад основного матеріалу. Для приймання даних в напрямі від ИП320 до OPC Server 320 використовувався елемент «Функціональная кнопка» (див. рис. 3).



Рис. 3. Елемент «Функціональная кнопка»

Для приймання даних в напрямі від OPC Server 320 до ИП320 використовувався елемент «Регистр» (див. рис. 4).



Рис. 4. Елемент «Регистр»

Для програмування ИП320 використовувався додаток «Конфігуратор ИП320», який дозволяв на одному дисплеї ИП320 формувати кілька логічних екранів і здійснювати до кожного з них незалежний доступ.

Для роботи з програмними бібліотеками Advosol Inc. в проєкт на C# в Visual Studio необхідно було додати посилання на відповідні dll-файли (наприклад, базові класи знаходилися в файлах OpсNetBase.Net4.dll і OpсNetBase.Net6.dll). Згідно з технологією .NET, вся логіка роботи з OPC-серверами реалізується за допомогою кількох класів. Основними такими класами є класи OpсServerBrowser (доступ до зареєстрованих OPC-серверів), BrowseTree (перегляд «адресного простору» OPC-сервера), ShowBrowseTree (перегляд «адресного простору» в елементі керування Windows TreeView), ShowBrowseTreeList (перегляд «адресного простору» в TreeView і ListView), OpсDataBind (автоматичне відображення і оновлення даних в елементах керування Windows), SyncIOGroup (синхронні читання і запис даних), RefreshGroup (асинхронні читання і запис даних), ItemListLoader (читання

конфігураційних файлів XML) і XmlIO (серіалізація і десеріалізація масивів).

Для безпосереднього обміну даними з ИП320 використовувати всі класи одночасно потреби не було – відповідно, було використано тільки кілька з них. Зокрема, конструктор класу OpсServerBrowser дозволяв створювати відповідний об'єкт для локального комп'ютера, для мережі (потрібне мережеве ім'я) або як Host із заданими параметрами доступу (потрібні ім'я домену, мережеве ім'я, пароль і логін). Для отримання списку OPC-серверів використовувався метод GetServerList(...) класу OpсServerBrowser, причому окремо можна було отримати OPC-сервери версій 2.04 і 2.05 та 3.0.

Обмін даними з ИП320 відбувався в синхронному режимі, тому основним класом, який використовувався, був SyncIOGroup. Метод Item(...) такого класу дозволяв по унікальному найменуванню елемента даних (повному «шляху» до такого елемента даних) отримати відповідну унікальну «структуру» ItemDef, яка утримувала властивості такого елемента у вигляді, в свою чергу, трьох окремих «структур» OpсIDef, OpсIInfo і OpсIRslt. Для доступу до елементів даних використовувалися чотири методи класу SyncIOGroup.

Зокрема, метод Add(...) додавав елемент даних в групу (він звичайно викликався автоматично при виклику методів Read(...) і Write(...)). Відповідно, для видалення елемента даних з групи використовувався метод Remove(...). Найбільш «корисними» для обміну даними були методи Read(...) і Write(...). Перший з них читав елемент даних, використовуючи унікальне найменування елемента даних або відповідну «структуру» ItemDef, а другий здійснював запис такого елемента.

Обмін даними з ИП320 здійснювався приблизно кожні 0,5 с, що при швидкості обміну даними 115200 кбіт/с забезпечувало практично «миттєву» реакцію такої панелі.

Висновки. В статті показано, що доступ до OPC-серверів (зокрема, згідно з OPC-стандартом OPC Data Access) можна здійснити, використовуючи не достатньо складні технології Microsoft Corp., такі як COM/DCOM і деякі інші, а середовище .NET і достатньо прості програмні компоненти і інструменти для роботи з OPC-технологіями сторонніх розробників (зокрема, Advosol Inc.).

Список літератури:

1. Пупена О.М. Розроблення людино-машинних інтерфейсів та систем збирання даних з використанням програмних засобів SCADA/HMI: Навч. посіб. Київ: Видавництво Ліра-К, 2020. 594 с. ISBN 978-617-7910-07-6.

2. OPC-сервер ЮНІВЕРС: офіц. веб-сайт. URL: www.raut-automatic.kiev.ua. Дата останнього перегляду: 04.03.2024.
3. Опис протоколу обміну між ПЕОМ і приладами ОБЕН: офіц. веб-сайт. URL: www.aqteck.com.ua. Дата останнього перегляду: 04.03.2024.
4. OPC Server 320: офіц. веб-сайт. URL: www.opctechs.com. Дата останнього перегляду: 04.03.2024.
5. OPC Classic for .NET. OPC DA .NET Client Development Toolkit for C# and VB.NET: офіц. веб-сайт. URL: www.advosol.com. Дата останнього перегляду: 04.03.2024.
6. Лісовець С.М. Особливості програмування панелі оператора ИП320 для використання як людиномашинного інтерфейсу / С.М. Лісовець, Н.В. Омецинська, О.Г. Гуйда, К.І. Скрипка, І.Л. Ківа. *Таврійський науковий вісник. Серія: Технічні науки*. 2021. Випуск 6. С. 43–49. DOI: 10.32851/tnv-tech.2021.6.6.
7. Лісовець С.М. Використання панелі оператора ИП320 в системах автоматизованого керування / С.М. Лісовець, Р.А. Кучма. *IV міжнародна науково-практична конференція «Мехатронні системи: інновації та інжиніринг»*. К.: КНУТД. 22 жовтня 2020. С. 178–179.
8. ОБЕН ИП320. Панель оператора. Настанова щодо експлуатування АРАВ.421449.002 РЭ.

Lisovets S.M. ACCESS TO OPC SERVERS FROM THE .NET ENVIRONMENT USING ADVOSOL SOFTWARE

Modern industrial production involves the use of modern automated control systems (automation systems), which should not only provide optimal control of such production, but also simple and effective control over it by service personnel. Such control is usually carried out with the help of SCADA systems – software packages that are designed to collect, process, display and archive data about the control object, as well as perform certain additional actions on such data, depending on the functional content of such packages. One of the «disadvantages» of SCADA systems on the market is their redundancy. Much of the functionality of such systems may never be used, and maintenance of such systems requires highly skilled personnel. One of the existing approaches to production control is the in-house development of SCADA systems with the simultaneous use of one or more ready-made OPC servers. The main obstacle to such development is the significant complexity of direct programming of data exchange with OPC servers. In this case, such products as, for example, software libraries of Advosol Inc. «come to the rescue», which allow you to easily organize data exchange with OPC servers from the .NET environment. In the research conducted, the results of which are presented in the materials of the article, using the example of the ИП320 operator panel using the C# programming language in the Visual Studio programming environment, the possibility of data exchange between such a panel and C# software via OPC Server 320 using Advosol Inc. libraries is shown. In this case, the exchange was carried out via the RS-485 interface using the Modbus RTU protocol, and the panel itself worked alternately in Master and Slave modes, and the configurator program was used to configure it.

Key words: OPC server; SCADA system; human-machine interface; software library; programming environment.