

ИНТЕГРАЦИЯ DELPHI ПРИЛОЖЕНИЙ С СИСТЕМАМИ ПРОМЫШЛЕННОЙ АВТОМАТИЗАЦИИ ЧЕРЕЗ СТАНДАРТ OPC

Невиницын В.Ю., Грименицкий П.Н., Отлягузов Д.С., Ваняйкин И.К.

Невиницын Владимир Юрьевич, Грименицкий Павел Николаевич, Отлягузов Демид Сергеевич,
Ваняйкин Илья Константинович

Ивановский государственный химико-технологический университет,

г. Иваново, Россия. 153000, Ивановская область, г. Иваново, пр. Шереметевский, д. 7.

E-mail: nevinitsyn@isuct.ru, grim@isuct.ru, demid0139@gmail.com, vanyaykin_ik@isuct.ru

В статье рассматривается вопрос разработки OPC-клиента на языке программирования Embarcadero Delphi для взаимодействия с различными узлами АСУТП. Приведены основные сведения о стандарте OPC (Open Platform Communications) и актуальность его использования в сфере промышленной автоматизации. В качестве примера рассмотрена работа с OPC-сервером Modbus Universal MasterOPC и процедура создания в нем новой конфигурации. Описан процесс добавления тэгов в OPC-сервер и приведены возможные настройки их параметров. Далее в работе показано, каким образом на языке Delphi можно реализовать механизм чтения и записи значений параметров в соответствующие тэги OPC-сервера. Приведен внешний вид разработанного приложения, описаны используемые компоненты. В качестве библиотеки для работы с OPC-серверами использовалась dOPC и входящий в ее состав компонент dOPCDAlient. Приведен исходный код процедур обработки событий элементов формы программы. Продемонстрирован результат работы приложения и взаимодействие с OPC-сервером Modbus Universal MasterOPC.

Ключевые слова: автоматизация, программирование, Delphi, OPC, сервер, клиент, SCADA.

INTEGRATION OF DELPHI APPLICATIONS WITH INDUSTRIAL AUTOMATION SYSTEMS USING THE OPC STANDARD

Nevinitsyn V.Yu., Grimenitsky P.N., Otyaguzov D.S., Vanyaykin I.K.

Nevinitsyn Vladimir Yur'evich, Grimenitsky Pavel Nikolayevich, Otyaguzov Demid Sergeyevich,
Vanyaykin Ilya Konstantinovich

Ivanovo State University of Chemical Technology,

Ivanovo, Russia. 153000, Ivanovo region, Ivanovo, Sheremetevsky ave., 7.

E-mail: nevinitsyn@isuct.ru, grim@isuct.ru, demid0139@gmail.com, vanyaykin_ik@isuct.ru

The paper deals with the problem of creating an OPC-client in Embarcadero Delphi programming language for interaction with various modules of the control systems. The basic information about the OPC (Open Platform Communications) standard and the relevance of its use in the field of industrial automation are given. As an example, the work with OPC-server Modbus Universal MasterOPC and the procedure of creating a new configuration in it are considered. The process of adding tags to the OPC-server is described and possible settings of their parameters are given. Further in the work it is shown how in Delphi language it is possible to realize the mechanism of reading and writing the values of parameters to the corresponding tags of OPC-server. The appearance of the designed application is given, the components used are described. The dOPC library and its dOPCDAlient component were used for working with OPC servers. The source code of procedures for processing events of the program form elements is given. The result of the application operation and interaction with OPC-server Modbus Universal MasterOPC is demonstrated.

Keywords: automation, programming, Delphi, OPC, server, client, SCADA

АКТУАЛЬНОСТЬ ПРОБЛЕМЫ

В настоящее время наблюдается качественный скачок в развитии систем управления технологическим оборудованием и технологическими процессами. Современные системы управления стали децентрализованными и носят название распределенных систем управления (PCY) [1]. На нижнем уровне структуры PCY находятся контроллеры – устройства, осуществляющие непосредственное управление технологическими процессами в реальном времени в темпе протекания технологического процесса. На верхнем уровне PCY расположены удаленные компьютеры (рабочие станции), позволяющие выполнять такие задачи, как сбор данных о параметрах технологического процесса, их обработка, архивирование принятой информации, визуализация информации о ходе технологического процесса, сигнализация о предаварийных и аварийных ситуациях и ряд других. Рабочие станции верхнего уровня представляют собой автоматизированные рабочие места (АРМ) производственного персонала, обслуживающего технологический процесс: операторы, технологи, наладчики, диспетчеры и т.д. Верхний уровень PCY называют диспетчерским или операторским уровнем. Диспетчерский уровень представлен персональными компьютерами (ПК), рабочими станциями, промышленными компьютерами, на которых установлены системы диспетчерского контроля и сбора данных (SCADA-системы). Взаимодействие между SCADA-системой и программируемыми контроллерами осуществляется с помощью стандарта OPC [2].

Технология OPC (Open Platform Communications) – это стандарт для обмена данными между промышленными устройствами и программным обеспечением в системах автоматизации. OPC обеспечивает унифицированный интерфейс, позволяющий различного оборудованию и приложениям взаимодействовать независимо от производителя и протокола. OPC устраняет необходимость разработки индивидуальных драйверов для каждого устройства, упрощая интеграцию разнородных систем, а также поддерживает работу в распределенных системах, что важно для современных промышленных предприятий.

В настоящее время стандарт OPC остается ключевой технологией для интеграции и управления данными в автоматизированных системах, особенно в условиях цифровизации и развития Industry 4.0. Так, в работах [3-11] показано применение технологии OPC при решении различных задач: проектирование систем управления, систе-

мы диспетчеризации и сбора данных, отладка кода программы контроллера на имитационной модели объекта, разработка учебных тренажеров и симуляторов систем управления технологическими установками и аппаратами. Следует отметить, что в ряде работ в качестве OPC-клиента для имитации работы технологического процесса используются различные системы компьютерной математики, например, пакет MATLAB, который обладает избыточным функционалом. Также на практике в ряде случаев требуется разработать свою собственную узкоспециализированную систему сбора данных (АРМ), причем существующие на рынке SCADA-системы не соответствуют требованиям заказчика. В этом случае целесообразно написать свое собственное приложение, как это предлагают некоторые проектные организации, предоставляющие услуги разработки АРМ оператора на различных языках программирования.

В рамках данной работы показано, как можно использовать язык Delphi для взаимодействия с OPC-сервером, что актуально при разработке собственных узкоспециализированных АРМ оператора, SCADA-систем, различных HMI интерфейсов взаимодействия с ПЛК, а также цифровых двойников и учебных тренажеров, где предполагается имитация реального объекта и передача в контроллер данных с модели объекта вместо сигналов с датчиков.

Создание конфигурации в Modbus Universal Masteropc Server. На этапе отладки программы воспользуемся таким OPC-сервером, как Modbus Universal MasterOPC Server (рис. 1), который предназначен для организации связи по стандарту OPC различных устройств и узлов АСУТП. Данный сервер также поддерживает наиболее распространенный в промышленности протокол Modbus ASCII/RTU/TCP для работы с контроллерами ОВЕН ПЛК, СПК, МКОН и другими. Добавим в конфигурацию OPC-сервера узел Node1, новое устройство Device1 и в него два тэга – Tag1 и Tag2 (рис. 2). На этапе отладки программы при добавлении тэгов выберем регион данных SERVER_ONLY, т.е. это переменные, которые будут находиться в сервере и не будут привязаны к каналам ввода-вывода реальных устройств (рис. 3). Среди доступных типов данных можно также выбрать bool, int32, uint32, float, double или string. Доступно три варианта типа доступа: ReadWrite (чтение и запись тэга), ReadOnly (только чтение), WriteOnly (только запись тэга).

В нашем случае тип данных тэга в сервере укажем float (вещественный тип), а тип доступа – ReadWrite.

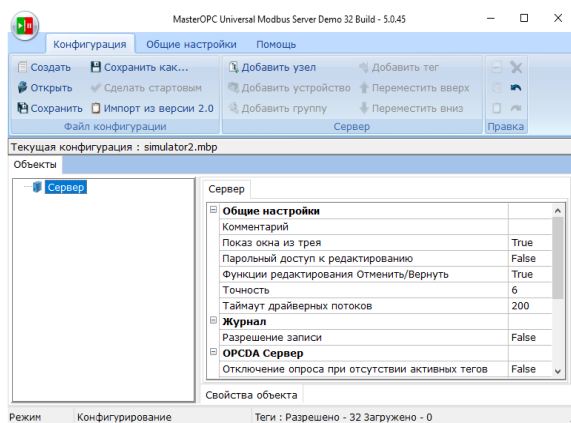


Рис. 1. Внешний вид окна OPC-сервера
Fig. 1. OPC-server window

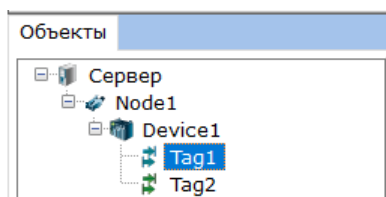


Рис. 2. Добавление новых элементов в структуру сервера
Fig. 2. Adding new items to the server structure

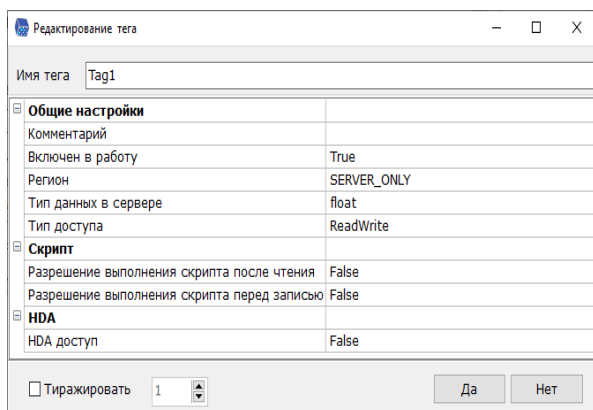


Рис. 3. Настройки тега
Fig. 3. Tag settings

Далее запустим конфигурацию, нажав кнопку **Старт** на панели инструментов (рис. 4). После данного действия вся конфигурация OPC-сервера (все добавленные устройства и тэги) будет доступна всем OPC-приложениям (клиентам), установленным на данной рабочей станции.

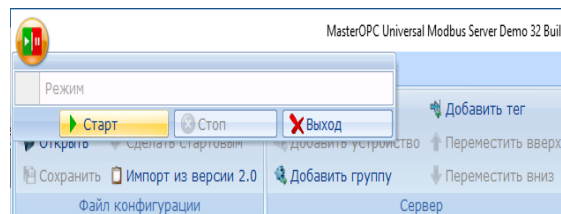


Рис. 4. Запуск конфигурации
Fig. 4. Starting the configuration

РАЗРАБОТКА DELPHI-ПРИЛОЖЕНИЯ

На следующем этапе создадим приложение в среде разработки Embarcadero Delphi 12, которое будет представлять собой OPC-клиент для чтения и записи значений в соответствующие тэги OPC-сервера.

В настоящее время на рынке имеется довольно мало компонентов для Delphi, которые позволяют обеспечить взаимодействие с OPC-серверами. К числу таких решений можно отнести библиотеку QuickOPC (разработчик OPC Labs) или решение от фирмы KASSL с компонентами dOPC. В данной работе мы использовали второй вариант (библиотека компонентов dOPC), как наиболее удобную и функциональную. Даная библиотека включает в себя три компонента (dOPCDAClient, dOPCServerBrowser и dOPCGUI), которые обеспечивают широкий функционал для взаимодействия по технологии OPC (рис. 5).

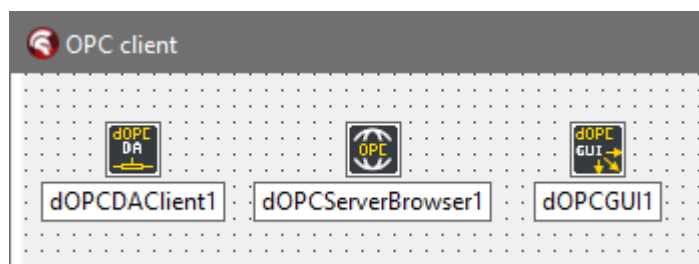
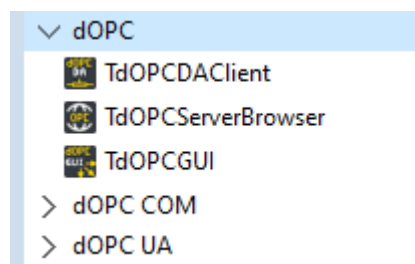


Рис. 5. Компоненты категории dOPC
Fig. 5. dOPC components



Основным компонентом библиотеки является блок dOPCDAClient. Этот компонент предназначен для работы с OPC DA (Data Access) серверами. OPC DA – это стандарт, который позволяет клиентским приложениям читать и записывать

данные с устройств автоматизации (например, ПЛК, датчиков и т.д.). Его можно применять для подключения к OPC серверу для мониторинга или управления промышленным оборудованием. Компонент dOPCDAClient обеспечивает следующие

функции: подключение к OPC DA серверу; чтение и запись данных (тэгов) с OPC сервера; подписка на изменения данных (асинхронное обновление); управление группами тэгов (OPC Groups) и элементами (OPC Items); обработка ошибок и событий, связанных с подключением и обменом данными. Далее покажем на примере, как можно взаимодействовать с OPC-сервером с помощью

компонента dOPCDAClient. Для начала организуем пользовательский интерфейс приложения, для чего добавим на форму необходимые элементы (рис. 6). Приложение должно обладать следующим функционалом: подключение к указанному OPC-серверу, чтение и запись в указанные тэги сервера, непрерывный мониторинг тэгов с указанным периодом опроса.

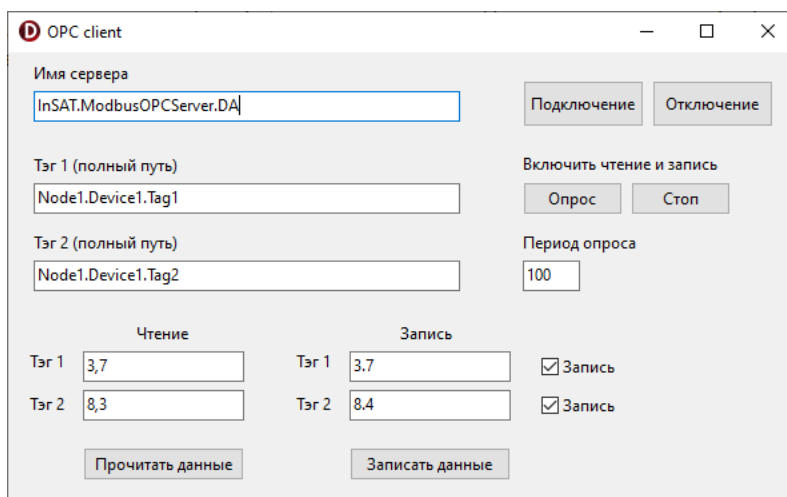


Рис. 6. Внешний вид разработанного приложения
Fig. 6. Appearance of the designed application

Для подключения к OPC-серверу используется следующий код (выполняется при нажатии на кнопку «Подключение»):

```
procedure TForm1.Button5Click(Sender: TObject);
begin
  dOPCDAClient1.Connect(EditSrvName.Text);
  dOPCDAClient1.OPCGroups.Add('NewGroup');
  dOPCDAClient1.OPCGroups[0].OPCItems.AddItem(EditTag1.Text);
  dOPCDAClient1.OPCGroups[0].OPCItems.AddItem(EditTag2.Text);
end;
```

В вышеприведенном фрагменте кода происходит подключение OPC-серверу с указанным в текстовом поле EditSrvName именем. Далее создается новая группа, в которую добавляются необходимые для работы тэги. Имена желаемых тэгов читаются из полей EditTag1 и EditTag2.

Отключение клиента осуществляется выполнением кода (кнопка «Отключение»):

```
procedure TForm1.Button6Click(Sender: TObject);
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  Timer1.Interval:=StrToInt(EditTmr.Text); // период опроса
  // чтение
  if dOPCDAClient1.OPCGroups[0].OPCItems[0].IsReadable
  then edit1.Text:=dOPCDAClient1.OPCGroups[0].OPCItems[0].ValueStr;
  if dOPCDAClient1.OPCGroups[0].OPCItems[1].IsReadable
  then edit2.Text:=dOPCDAClient1.OPCGroups[0].OPCItems[1].ValueStr;
```

```
begin
  dOPCDAClient1.OPCGroups[0].Destroy;
  dOPCDAClient1.Disconnect;
end;
```

Для чтения значения тэга используется свойство ValueStr для соответствующего элемента и группы компонента dOPCDAClient. Для записи параметра в указанный тэг можно использовать процедуры WriteAsync (асинхронная запись) или WriteSync (синхронная запись). Для непрерывного чтения и записи значений тэгов с заданным периодом опроса (задается в поле EditTmr) на форму добавлен компонент Timer, который выполняет код с заданным интервалом. Чтение и запись параметров происходит с соответствующей проверкой доступности тэга в OPC-сервере (свойства IsReadable, IsWriteable).

Для записи тэгов также необходимо установить соответствующие флажки на форме. Ниже приведен соответствующий код программы.

// запись

```
if (CheckBox1.Checked and dOPCDAClient1.OPCGroups[0].OPCItems[0].IsWriteAble) then
  dOPCDAClient1.OPCGroups[0].OPCItems[0].WriteAsync(edit3.Text,-1,-1);
```

```
if (CheckBox2.Checked and dOPCDAClient1.OPCGroups[0].OPCItems[1].IsWriteAble) then
  dOPCDAClient1.OPCGroups[0].OPCItems[1].WriteAsync(edit4.Text,-1,-1);
```

end;

Таймер включается путем выполнения команд timer1.Enabled:=True или timer1.Enabled:=False, которые указаны в обработчиках кнопок «Опрос» и «Стоп».

Также в программе предусмотрено однократное чтение и запись значений тэгов (кнопки «Прочитать данные» и «Записать данные»).

РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Запустим созданное приложение, укажем имя сервера «InSAT.ModbusOPCServer.DA» и полный путь к желаемым тэгам: «Node1.Device1.

Tag1» и «Node1.Device1.Tag2». Далее подключимся к OPC-серверу и запустим опрос с помощью соответствующих кнопок формы.

Далее перейдем в окно OPC-сервера Modbus Universal MasterOPC Server и проверим значения тэгов (рис. 7). Можно убедиться, что в тэги записались указанные в программе значения. Также на вкладке сообщения можно видеть все операции над тэгами. В данном случае это процедура записи значения с указанием времени.

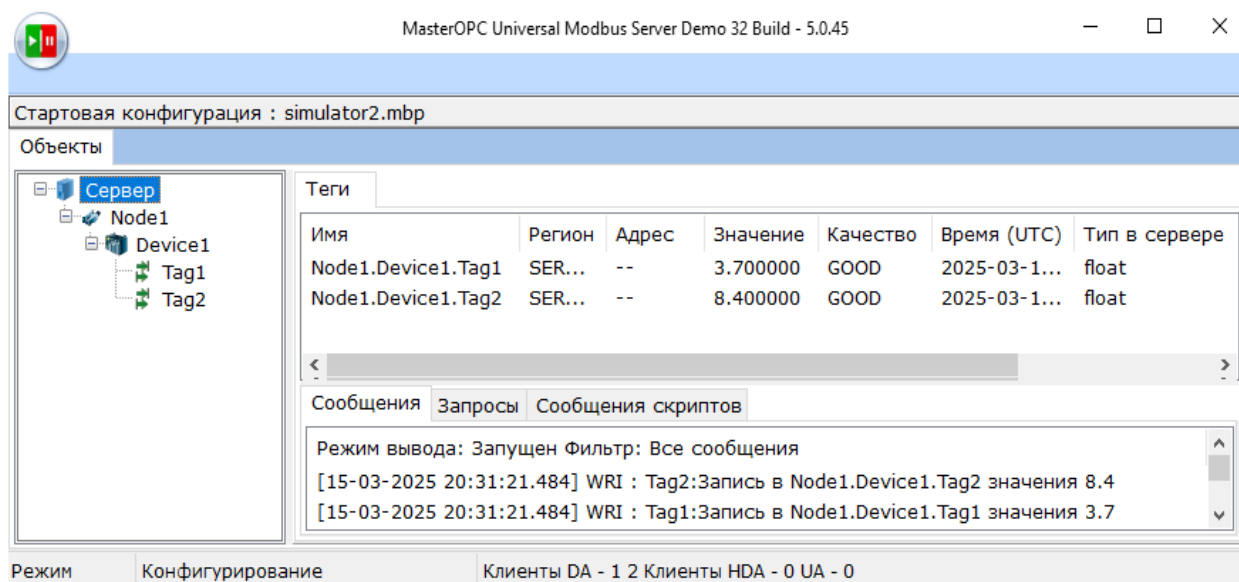


Рис. 7. Окно мониторинга OPC-сервера

Fig. 7. OPC-server monitoring window

Таким образом в работе показано, как можно применять технологию OPC и компоненты dOPC в среде программирования Delphi для разработки программного обеспечения АСУТП и написания собственных АРМ оператора, SCADA и НМИ систем. Данную методику можно также использовать при внедрении цифровых двойников в предприятие для объединения физической производственной системы и ее имитационной модели посредством технологии OPC. Также следует отметить сферу разработки программных

симуляторов систем управления и полунатурных тренажеров АСУТП для обучения персонала, в которых реальный объект заменяется его адекватной имитационной моделью и интегрируется с существующей АСУТП через стандарт OPC.

Авторы заявляют об отсутствии конфликта интересов, требующего раскрытия в данной статье

The authors declare the absence a conflict of interest warranting disclosure in this article.

ЛИТЕРАТУРА

1. Харазов В.Г. Интегрированные системы управления технологическими процессами. 3-е изд. СПб.: Профессия. 2013. 550 с.

REFERENCES

1. Harazov V.G. Integrated process control systems. 3rd edition. SPb.: Professiya. 2013. 550 p. (in Russian).

2. **Кангин В.В., Кангин М.В., Ямолдин Д.Н.** Разработка SCADA-систем: учебное пособие. Москва; Вологда: Инфра-Инженерия, 2019. 564 с.
3. **El Zerk A., Ouassaid M., Zidani Y.** Development of a real-time framework between MATLAB and PLC through OPC-UA: A case study of a microgrid energy management system. *Scientific African*. 2023. V. 21. P. e01846. DOI 10.1016/j.sciaf.2023.e01846.
4. **Мочалов Л.В., Дроздов В.Г.** Моделирование автоматической системы управления мяльно-трепальным агрегатом. *Технологии и качество*. 2024. № 1(63). С. 40-44. DOI 10.34216/2587-6147-2024-1-63-40-44.
5. **Qiu G.** Particle swarm algorithm and OLE for process control based PID design for dual capacity water tank. *Journal of Physics: Conference Series*. 2021. V 2044. N 1. P. 012175. DOI 10.1088/1742-6596/2044/1/012175.
6. **Нажимова Н.А., Наумова Е.Г., Кулигина Н.О.** Взаимодействие системы Matlab и промышленного программного обеспечения системы управления процессом синтеза стирол-акриловой дисперсии. *Приборы и системы. Управление, контроль, диагностика*. 2017. № 10. С. 8-11.
7. **Lyu Z.** Handbook of Digital Twins. 1st ed. CRC Press, 2024. 922 p.
8. **Rani S., Bhambri P., Kumar S., Pareek P.K., Elngar A.A.** AI-Driven Digital Twin and Industry 4.0: A Conceptual Framework with Applications. 1st ed. CRC Press, 2024. 338 p.
9. **Невиницын В.Ю., Грименицкий П.Н., Отлягузов Д.С., Смирнов К.О.** Разработка цифрового двойника на языке программирования Delphi. *Современные наукоемкие технологии. Региональное приложение*. 2024. № 4(80). С. 131-137. DOI 10.6060/snt.20248004.00018.
10. **Невиницын В.Ю., Грименицкий П.Н., Лихач Д.С., Субботин П.А.** Разработка автоматизированной системы сбора данных и оперативного диспетчерского управления ректификационной установкой. *Современные наукоемкие технологии. Региональное приложение*. 2024. № 2(78). С. 85-92. DOI 10.6060/snt.20247802.00012.
11. **Невиницын В.Ю., Грименицкий П.Н., Ваныякин И.К., Отлягузов Д.С.** Разработка цифровых двойников типовых технологических процессов. *Современные наукоемкие технологии. Региональное приложение*. 2025. № 1(81). С. 111-117. DOI 10.6060/snt.20258101.00013.
2. **Kangin V.V., Kangin M.V., Yamoldinov D.N.** Development of SCADA-systems: textbook. Moscow: Infra-Engineering, 2019. 564 p. (in Russian).
3. **El Zerk A., Ouassaid M., Zidani Y.** Development of a real-time framework between MATLAB and PLC through OPC-UA: A case study of a microgrid energy management system. *Scientific African*. 2023. V. 21. P. e01846. DOI 10.1016/j.sciaf.2023.e01846.
4. **Mochalov L.V., Drozdov V.G.** Simulation of the automatic control system of the chopping and cracking unit. *Tekhnologii i kachestvo*. 2024. N 1(63). P. 40-44. (in Russian). DOI 10.34216/2587-6147-2024-1-63-40-44.
5. **Qiu G.** Particle swarm algorithm and OLE for process control based PID design for dual capacity water tank. *Journal of Physics: Conference Series*. 2021. V 2044. N 1. P. 012175. DOI 10.1088/1742-6596/2044/1/012175.
6. **Nazhimova N.A., Naumova E.G., Kuligina N.O.** Interaction of Matlab system and industrial software of control system of styrene-acrylic dispersion synthesis synthesis process. *Pribory i sistemy. Upravlenie, kontrol', diagnostika*. 2017. N 10. P. 8-11. (in Russian).
7. **Lyu Z.** Handbook of Digital Twins. 1st ed. CRC Press, 2024. 922 p.
8. **Rani S., Bhambri P., Kumar S., Pareek P.K., Elngar A.A.** AI-Driven Digital Twin and Industry 4.0: A Conceptual Framework with Applications. 1st ed. CRC Press, 2024. 338 p.
9. **Nevinitsyn V.Yu., Grimenitsky P.N., Otyaguzov D.S., Smirnov K.O.** Digital twin design using Delphi programming language. *Modern high technologies. Regional application*. 2024. N 4(80). P. 131-137. (in Russian). DOI 10.6060/snt.20248004.00018.
10. **Nevinitsyn V.Yu., Grimenitsky P.N., Likhach D.S., Subbotin P.A.** Design of an automated system for data acquisition and supervisory control of a rectification unit. *Modern high technologies. Regional application*. 2024. N 2(78). P. 85-92. (in Russian). DOI 10.6060/snt.20247802.00012.
11. **Nevinitsyn V.Yu., Grimenitsky P.N., Vanyaykin I.K., Otyaguzov D.S.** Digital twins design of typical technological processes. *Modern high technologies. Regional application*. 2025. N 1(81). P. 111-117. (in Russian). DOI 10.6060/snt.20258101.00013.

Поступила в редакцию(Received) 11.03.2025
Принята к опубликованию (Accepted) 14.04.2025