

## СПЕЦИФИКА РЕАЛИЗАЦИИ ИСТОРИЧЕСКИХ ДАННЫХ В OPC UA СЕРВЕРЕ ДЛЯ СИСТЕМЫ ЧПУ

Г.М. Мартинов, А.С. Захаров (МГТУ «СТАНКИН»)

Рассмотрена структура механизма сбора и хранения исторических данных о ходе процесса обработки на станке с системой ЧПУ с применением протокола OPC UA. В качестве примера приведена программная реализация OPC UA сервера для системы ЧПУ «АксиОМА Контрол», использующего библиотеку с открытым исходным кодом open62541. Проведено тестирование работы с историческими данными с применением универсального клиента UaExpert<sup>1</sup>.

Ключевые слова: система ЧПУ, OPC UA сервер, OPC UA клиент, исторические данные.

## Введение

С развитием Industry 4.0 и внедрением в современную промышленность Internet-технологий значимость сбора и архивирования данных, получаемых от систем ЧПУ, обрело новый смысл. Для получения данных теперь не нужно обходить множество станков с системами ЧПУ и скачивать данные по отдельности с каждой системы управления, все данные в режиме реального времени отправляются в централизованное хранилище для последующей обработки, анализа и оптимизации рабочего процесса [1, 2]. OPC UA – стандарт для передачи данных в промышленных сетях, он обеспечивает надежное и защищенное соединение между системами управления технологическими устройствами, при этом реализуется взаимодействие с устройствами, функционирующими под управлением разных операционных систем.

В систему ЧПУ встраивается OPC UA сервер для получения данных от ядра системы управления в режиме реального времени и последующей их передачи OPC UA клиентам [3].

## Механизм реализации OPC сервера с историческими данными

В процессе исследования были проанализированы следующие библиотеки с открытым исходным

кодом, реализующие стандарт OPC UA: freeopcua, UA-.NETStandard, UA Expert SDK и open62541. Библиотеки UA-.NETStandard и UA Expert SDK имеют C#<sup>2</sup> реализацию, что является ограничением для их применения в системах управления, функционирующих в жестком реальном времени. В результате выбрана библиотека open62541, имеющая C-реализацию и работающая как под управлением ОС Windows и Linux.

Структура протокола OPC UA состоит из 11 частей (рис. 1). С первой по седьмую частям спецификации определяются основные механизмы взаимодействий, такие как безопасность, профили, службы, модель адресного пространства. Части спецификации с восьмой по 11 определяют типы доступа к данным, аварийные сигналы и условия [4].

Модель адресного пространства, описанная в третьей части спецификации, представляет собой иерархию узлов, которая структурирует передаваемые данные, каждый узел содержит набор атрибутов. Часть атрибутов, такие как «Тип данных», является обязательной, остальные атрибуты необязательные, например, атрибут «Описание» [5].

## Формирование исторических данных

Формирование исторических данных определено в 11 части спецификации, клиентам предоставляется



Рис. 1. Спецификация OPC UA

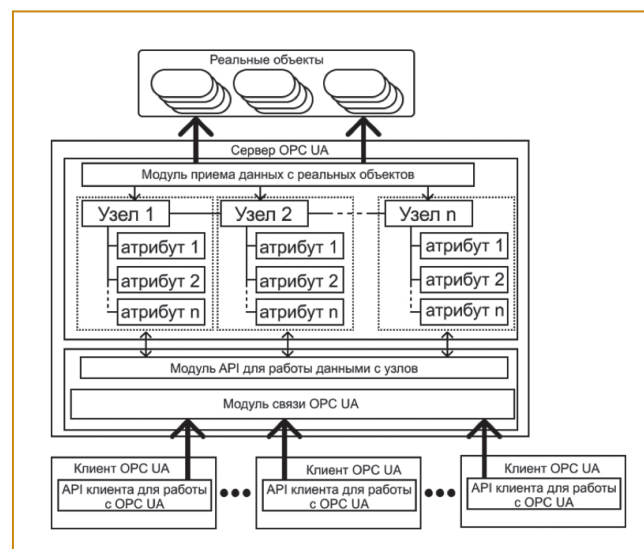


Рис. 2. Структура клиент – серверного взаимодействия согласно стандарту OPC UA

<sup>1</sup> Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-07-00305/20.

<sup>2</sup> C# — объектно-ориентированный язык программирования с автоматическим управлением памятью, предназначенный для программирования на платформе .NET.

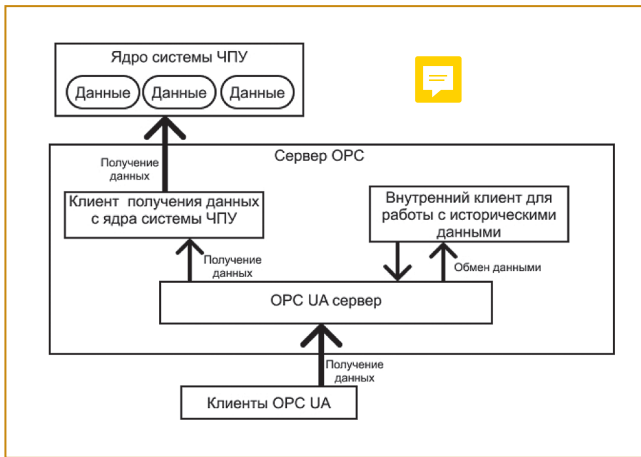


Рис. 3. Реализация сбора и передачи исторических данных по протоколу OPC UA

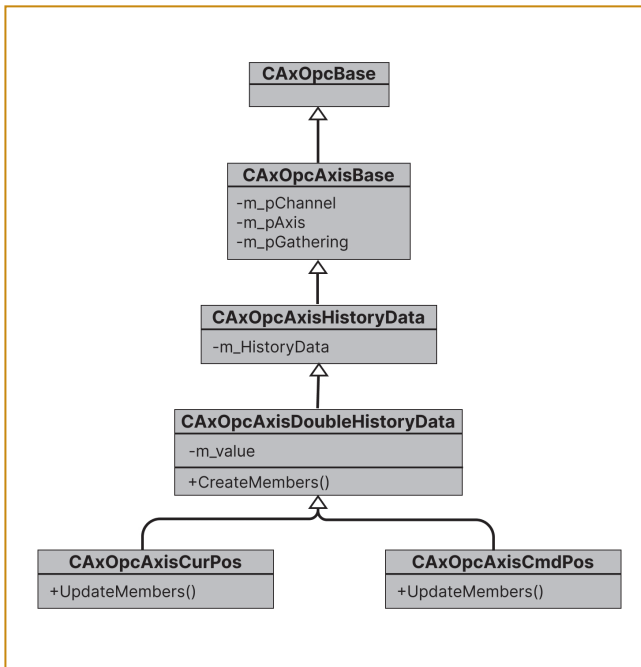


Рис. 4. Диаграмма классов сбора исторических данных для осей в нотации UML

доступ к данным, полученным за определенный интервал времени, данные формируются по модели адресного пространства узлов, каждый узел сохраняет информацию о себе. Все данные хранятся в OPC UA сервере, и по запросу OPC UA клиента сервер выдает информацию об узле за определенный интервал времени (рис. 2) [6].

### Реализация сбора и передачи исторических данных по протоколу OPC UA в системе ЧПУ

Сервер для сбора данных реализован на языке программирования C++, сервер состоит из трех модулей: клиента ядра системы управления, отвечающего за получение данных от ядра системы ЧПУ, клиента для внутренней работы с историческими данными и модуль на базе библиотеки open62541, реализующий работу по протоколу OPC UA (рис. 3).

Формирование исторических данных происходит при получении исходных данных от ядра системы ЧПУ, на основе полученных данных формирует узлы согласно стандарту OPC UA, далее клиент запрашивает данные от сервера для получения сформированных исторических данных.

В реализации OPC UA сервера специальные классы формируют и хранят исторических данных за определенный интервал времени. В качестве иллюстрации приведена диаграмма классов (рис. 4) получения исторических данных с осей системы ЧПУ. Базовый класс CAxOpcBase реализует механизм получение данных от ядра системы управления в реальном времени и их буферизацию, далее эти данные могут быть запрошены в формате обычных либо исторических данных.

### Тестирование разработанного OPC UA сервера

В качестве инструмента тестирования использован свободно распространяемый универсальный OPC UA клиент UaExpert, позволяющий вывести исторические данные отслеживаемых узлов в графической или текстовой форме (рис. 4) [7].

Подключенные в текущем моменте времени OPC UA серверы отображаются в окне «Информация о сервере». Окно «Дерево узлов» отображает доступные (реализованные)

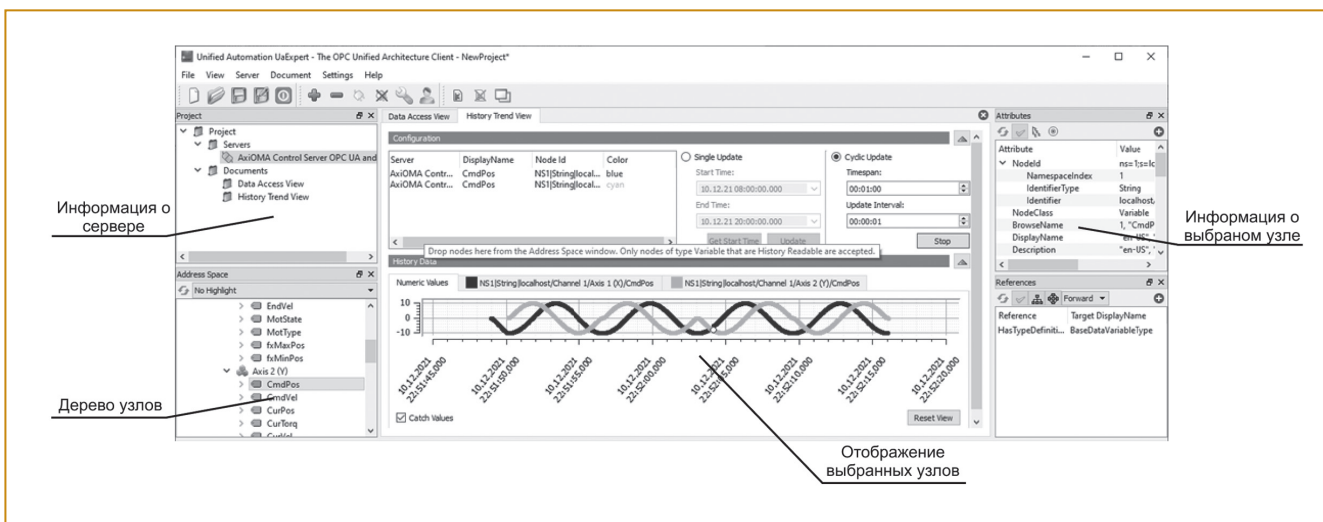


Рис. 5. Отображение данных в программе UaExpert

узлы данных, далее пользователь выбирает отслеживаемые данные с системы ЧПУ и переносит их в окне «Отображение выбранных узлов» (рис. 5).

#### Заключение

Использование библиотеки `open62541` предоставляет готовое решение по модели безопасности протокола OPC UA и позволяет сфокусировать процесс разработки OPC UA сервера на информационной модели данных системы ЧПУ, в том числе и с использованием исторических данных.

Разработка OPC UA сервера на языке C++ обеспечивает портируемость решения для ОС Windows и Linux.

Тестирование обычных узлов и узлов с историческими данными, получаемыми от системы ЧПУ «АксиОМА Контроль», осуществлялось с использованием свободно распространяемого инструментария `UaExpert`, являющегося по сути универсальным OPC UA клиентом.

#### Список литературы

1. *Мартинов Г.М.* Цифровые производственные технологии согласно концепции Industry 4.0 // Автоматизация в промышленности. 2019. №5. с.3-5.
2. *Мартинов Г.М., Козак Н.В., Нежметдинов Р.А., Григорьев А.С., Обухов А.И., Мартинова Л.И.* Метод деком-

позиции и синтеза современных систем с ЧПУ // Автоматизация в промышленности. 2013. № 5. С. 9-15.

3. *Martinov G., Issa A. and Martinova L.* (2019). Controlling CAN Servo Step Drives and Their Remote Monitoring by Using Protocol OPC UA. In: 2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon). Vladivostok: IEEE, pp.1-5.
4. *Mahnke W., Leitner S.H., Damm M.* OPC Unified Architectureringer Science. 2009.
5. *Martinov G., Nikishechkin P., Al Khoury, A and Issa A.* 2020. Control and remote monitoring of the vertical machining center by using the OPC UA protocol. In: IOP Conf. Series: Materials Science and Engineering 919, 032030. pp.1-8.
6. *Kovalev I., Issa A., Nikishechkin P., Chervonnova N. and Petrov A.* Development of a data collection system for a CNC system using cloud FRED technology and OPC UA specification. ICMTMTE 2020, MATEC Web of Conferences 329, 03043 (2020).
7. *Никишечкин П.А., Аль Хури А., Исса А., Червоннова Н.Ю.* Использование протокола OPC UA для мониторинга работы технологического оборудования, управляемого с помощью программно-реализованного контроллера // Автоматизация в промышленности. 2020. №5. с.37-40

**Мартинов Георгий Мартинович** — д-р техн. наук, заведующий кафедрой компьютерных систем управления,  
**Захаров Александр Сергеевич** - аспирант кафедры компьютерных систем управления  
 ФГБОУ ВО «МГТУ «СТАНКИН».  
 Контактный телефон (499) 972-94-40.  
 E-mail: e-mail@ncsystems.ru