



Лекция № 5

«ПРИМЕНЕНИЕ ОРС ТЕХНОЛОГИЙ»

Ведущий преподаватель: канд. техн. наук, доцент кафедры ИУТС Альчаков Василий Викторович

Основные понятия

OPC (аббр. от англ. OLE for Process Control) — семейство программных технологий, предоставляющих единый интерфейс для управления объектами автоматизации и технологическими процессами.

Стандарт OPC разрабатывался с целью сократить затраты на создание и сопровождение приложений промышленной автоматизации.

Основная цель — предоставить разработчикам промышленных программ универсальный фиксированный интерфейс (то есть набор функций) обмена данными с любыми устройствами. В то же время разработчики устройств предоставляют программу, реализующую этот интерфейс (набор функций).

Структура системы на базе OPC

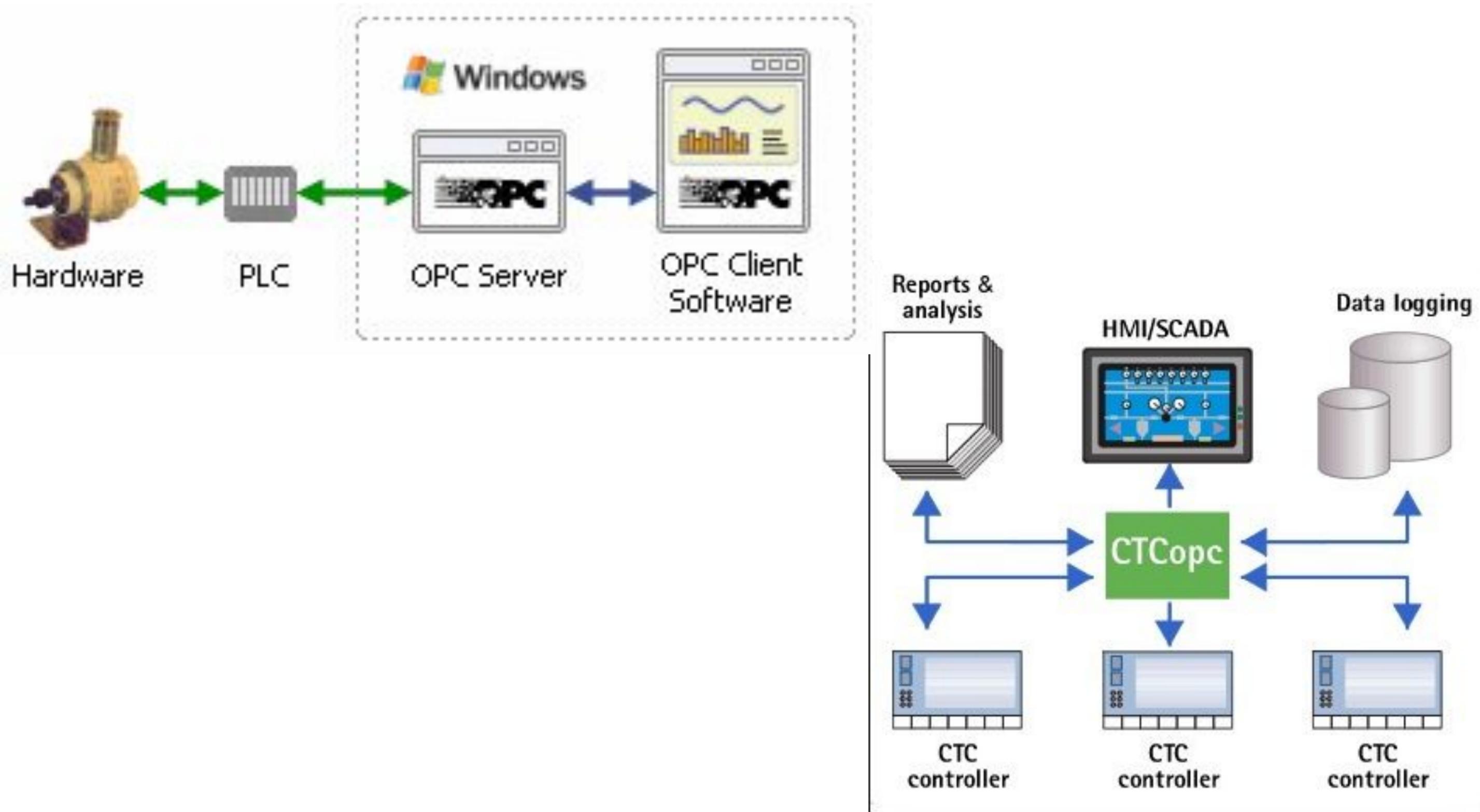


4

Что такое OPC? Где используются

OPC/

Структура системы на базе OPC

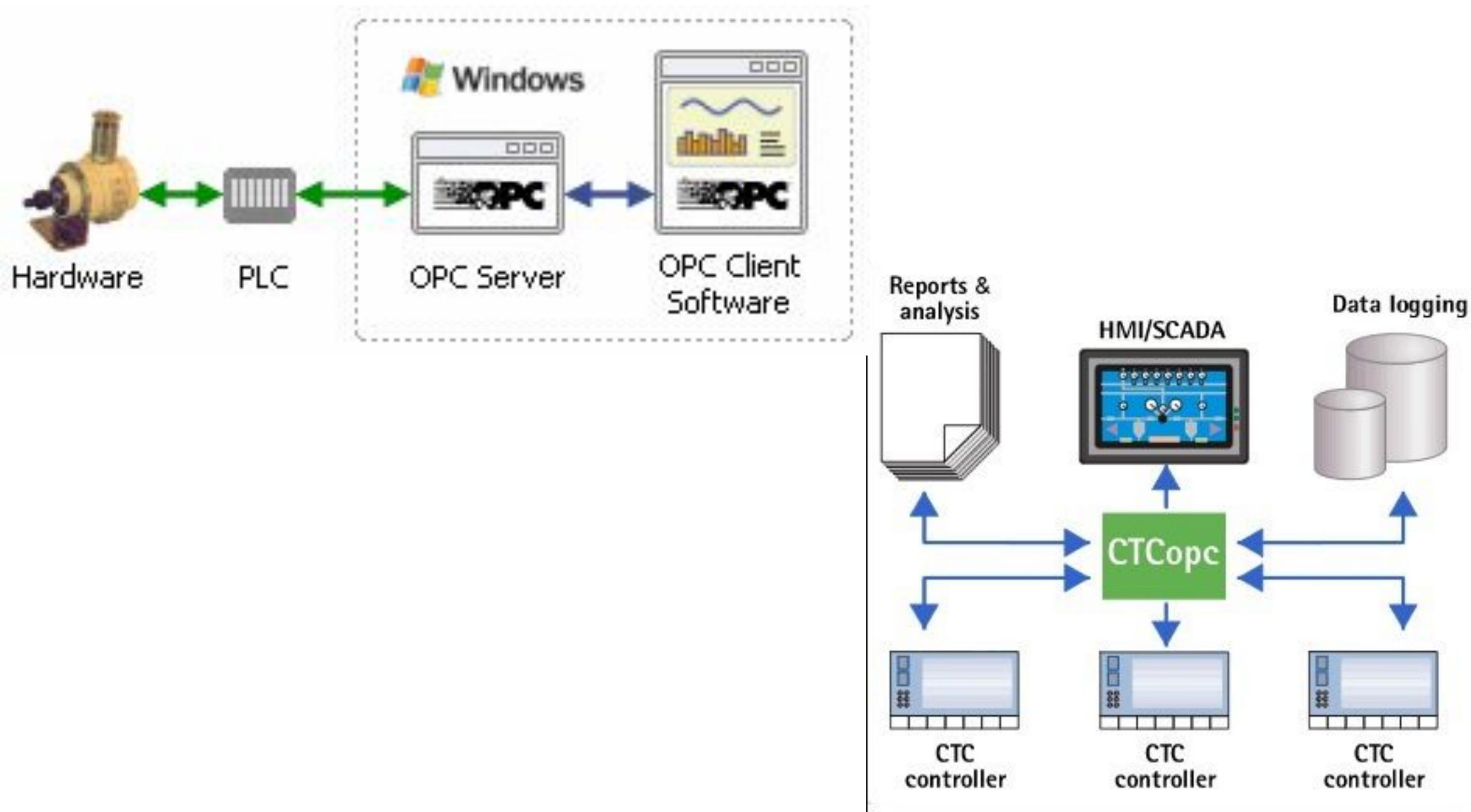


5

Что такое OPC? Где используются

OPC/

Структура системы на базе OPC



6

Что такое ОРС? Где используются

ОРС/

Распределенные объекты управления



7 Что такое ОРС? Где используются

ОРС?

Типовая структура ОРС (Hardware)

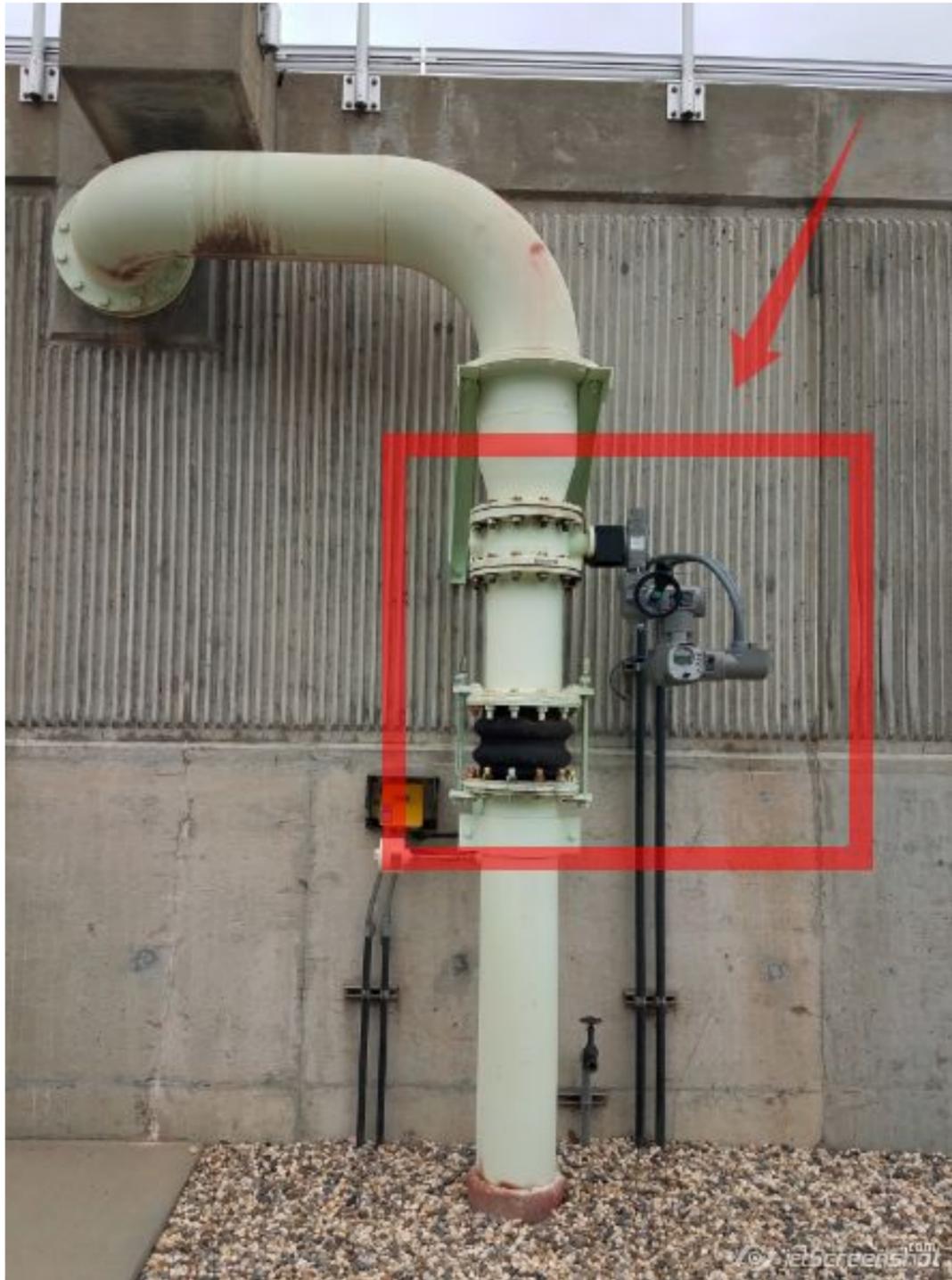


Caption: Components of a typical Safety Instrumented System (SIS) including sensor (1), safety PLC (2) and actor (3), consisting of actuator and valve.

8 Что такое ОРС? Где используются

ОРС/

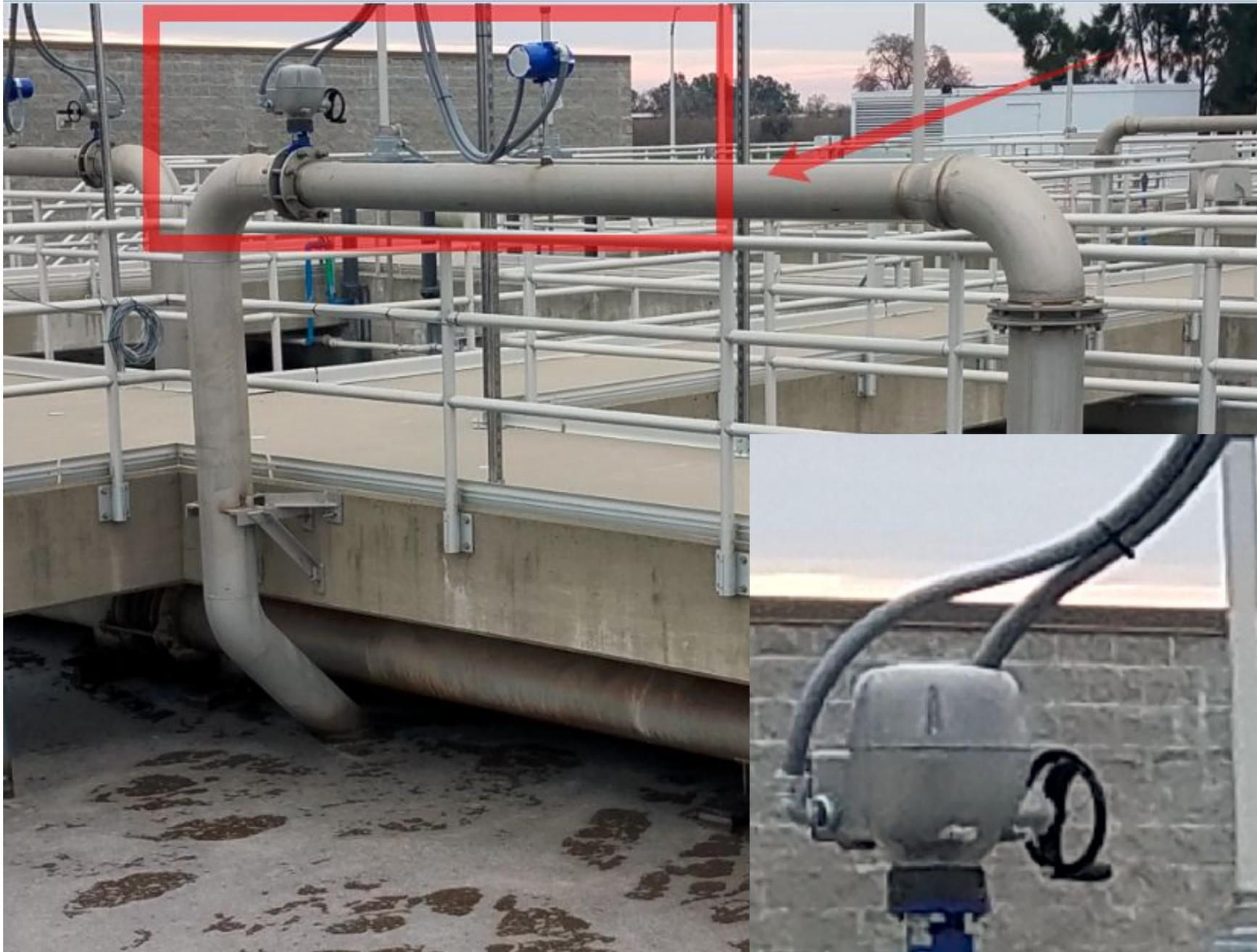
Пример использования



9 Что такое ОРС? Где используются

ОРС/

Пример использования



10 OPC Клиент/Сервер

Определения

OPC-сервер – программа, получающая данные во внутреннем формате устройства или системы и преобразующая эти данные в формат OPC. OPC-сервер является источником данных для OPC-клиентов. По своей сути OPC-сервер – это некий универсальный драйвер физического оборудования, обеспечивающий взаимодействие с любым OPC-клиентом.

OPC-клиент – программа, принимающая от OPC-серверов данные в формате OPC. Технология OPC определяет интерфейс между OPC-клиентом и OPC-серверами.

11 OPC Клиент/Сервер

Преимущество использования для пользователя

- Стандартный интерфейс позволяет при смене физического оборудования заменять только OPC-сервер, OPC-клиент продолжает работать.
- При смене OPC-клиента на новый все оборудование, работающее через OPC-сервер, можно будет использовать далее, не опасаясь отсутствия драйвера в новом приложении.

Таким образом, OPC-технология обеспечивает независимость потребителей от наличия или отсутствия драйверов или протоколов, что позволяет выбирать оборудование и программное обеспечение, наиболее полно отвечающее реальным потребностям бизнеса.

**Преимущество использования для
производителя оборудования**

- Универсальный механизм интеграции производимого им оборудования в любую систему, поддерживающую технологию OPC.

До создания OPC-технологии производителю промышленного оборудования приходилось создавать и поддерживать множество драйверов для наиболее распространенных систем автоматизации (или договариваться с производителями этих систем). Применение OPC-технологии позволяет отказаться от создания драйверов и заменяет их одним универсальным OPC-сервером, многократно сокращая затраты на разработку и дальнейшее сопровождение. При этом обеспечивается возможность подключения любой системы автоматизации, наиболее подходящей клиенту, а не только одной из нескольких наиболее распространенных.

13 OPC Сервер

Характеристика OPC сервера

Технология OPC была разработана для унификации механизмов взаимодействия программного обеспечения систем управления с аппаратурой этих систем. В рамках этой технологии OPC-серверы собирают данные от [контроллеров](#) и предоставляют их OPC-клиентам (например, SCADA-системам). Любой OPC-клиент может обмениваться данными с любым OPC-сервером вне зависимости от специфики устройства, для которого разрабатывался конкретный OPC-сервер.

Виды серверов (по функционалу)

13 OPC Сервер

Виды серверов (по функционалу)

OPC DA (OPC Data Access) - спецификация для обмена данными между клиентом (например SCADA) и аппаратурой (контроллерами, модулями ввода-вывода и др.) в реальном времени;

OPC Alarms & Events (A&E) - спецификация для уведомления клиента о событиях и сигналах тревоги, которые посылаются клиенту по мере их возникновения. Этот сервер пересылает аварийные сигналы, действия оператора, информационные сообщения, результаты контроля состояния системы;

OPC HDA (Historical Data Access) - спецификация для доступа к предыстории процесса (к сохраненным в архиве данным). Сервер обеспечивает унифицированный способ доступа с помощью DCOM технологии.

Обеспечивает чтение, запись и изменение данных;

OPC Unified Architecture - принципиально новый набор спецификаций, который уже не базируется на DCOM технологии.

15 OPC Сервер

OPC DA Сервер

Сервер OPC DA является наиболее широко используемым в промышленной автоматизации. Он обеспечивает обмен данными (запись и чтение) между клиентской программой и физическими устройствами.

Данные состоят из трех полей: значение, качество и временная метка.

Параметр качества данных позволяет передать от устройства клиентской программе информацию о выходе измеряемой величины за границы динамического диапазона, об отсутствии данных, ошибке связи и другие.

Режимы чтения данных

Существует четыре стандартных режима чтения данных из OPC сервера:

- Синхронный режим: клиент посылает запрос серверу и ждет от него ответ;
- Асинхронный режим: клиент отправляет запрос и сразу же переходит к выполнению других задач. Сервер после выполнения функции запроса посылает клиенту уведомление и тот забирает предоставленные данные;
- Режим подписки: клиент сообщает серверу список тегов, значения которых сервер должен отправлять клиенту только в случае их изменения. Для того, чтобы шум данных не был принят за их изменение, вводится понятие "мертвой зоны", которая слегка превышает максимально возможный размах помехи;
- Режим обновления данных: клиент вызывает одновременное чтение всех активных тегов. Активными называются все теги, кроме обозначенных как "пассивные". Такое деление тегов уменьшает загрузку

17 OPC Сервер

Kepware Server



[Products](#) ▾ [How to Buy](#) ▾ [Industries](#) ▾ [Partners](#) ▾

Introducing KEPServerEX Version 6

With new enhancements to core server functionality and user experience, our industry-leading industrial connectivity platform is enterprise-ready, IT-friendly, and secure.

[Learn More](#)

[Featured Solutions](#)

<https://www.kepware.com/en-us/>

18 OPC Сервер

Kepware Server

The image shows two overlapping windows from the Kepware software suite. The top window is 'KEPServerEX 6 Configuration [Connected to Runtime]', which displays a tree view of the project structure on the left and a table of tag configurations in the center. The bottom window is 'OPC Quick Client - Untitled *', which displays a tree view of the data source on the left and a table of real-time data values in the center. A red box highlights the 'Simulation Examples.Functions.Ramp3' and 'Simulation Examples.Functions.Random1' rows in the OPC Quick Client table.

KEPServerEX 6 Configuration [Connected to Runtime]

Tag Name	Address	Data Type	Scan Rate	Scaling	Description
Ramp1	RAMP (120, 35, 100...	Long	100	None	Value increments b.
Ramp2	RAMP (300, 150.75...	Float	100	None	Value decrements ..
Ramp3	RAMP (250, 0, 1000...	Long	100	None	Value increments b.
Ramp4	RAMP (250, 0, 1000...	Long	100	None	Value decrements ..
Random1	RANDOM (30, -20, ...	Long	100	None	Random values fro.
Random2	RANDOM (100, 0, 1...	Long	100	None	Random values fro.
Random3	RANDOM (100, -10...	Long	100	None	Random values fro.
Random4	RANDOM (1000, -9...	Long	100	None	Random values fro.

OPC Quick Client - Untitled *

Item ID	Data Type	Value	Timestamp	Quality
Simulation Examples.Functions.Ramp1	Long	59	10:55:27.870	Good
Simulation Examples.Functions.Ramp2	Float	155	10:55:27.870	Good
Simulation Examples.Functions.Ramp3	Long	460	10:55:27.870	Good
Simulation Examples.Functions.Ramp4	Long	715	10:55:27.870	Good
Simulation Examples.Functions.Random1	Long	50	10:55:27.870	Good
Simulation Examples.Functions.Random2	Long	491	10:55:27.870	Good
Simulation Examples.Functions.Random3	Long	-544	10:55:27.870	Good
Simulation Examples.Functions.Random4	Long	-124	10:55:26.852	Good

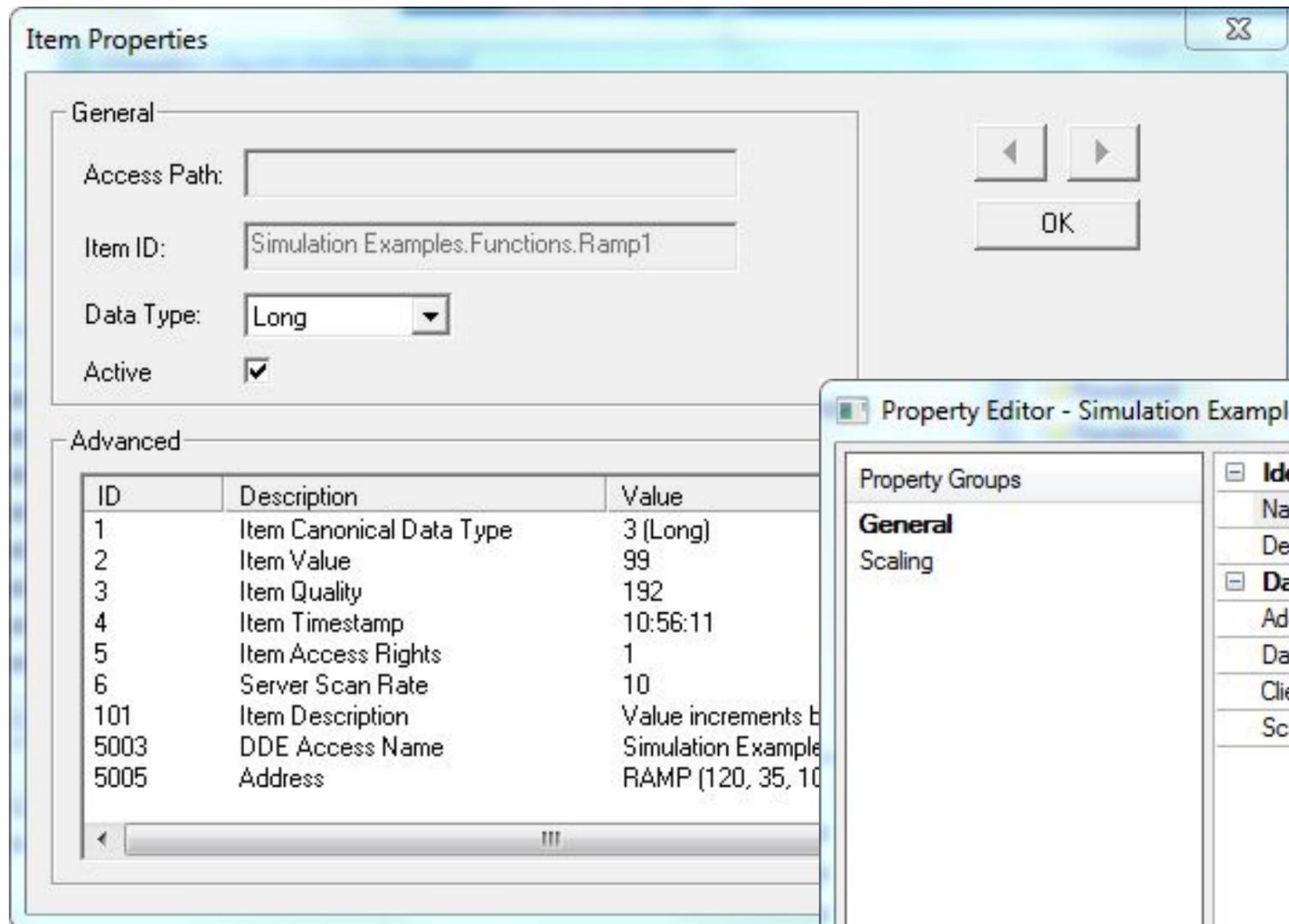
Date: 2/21/2017 10:53:28 AM
Event: Added 12 items to ...
2/21/2017 10:53:28 AM
Event: Added 4 items to ...

Ready

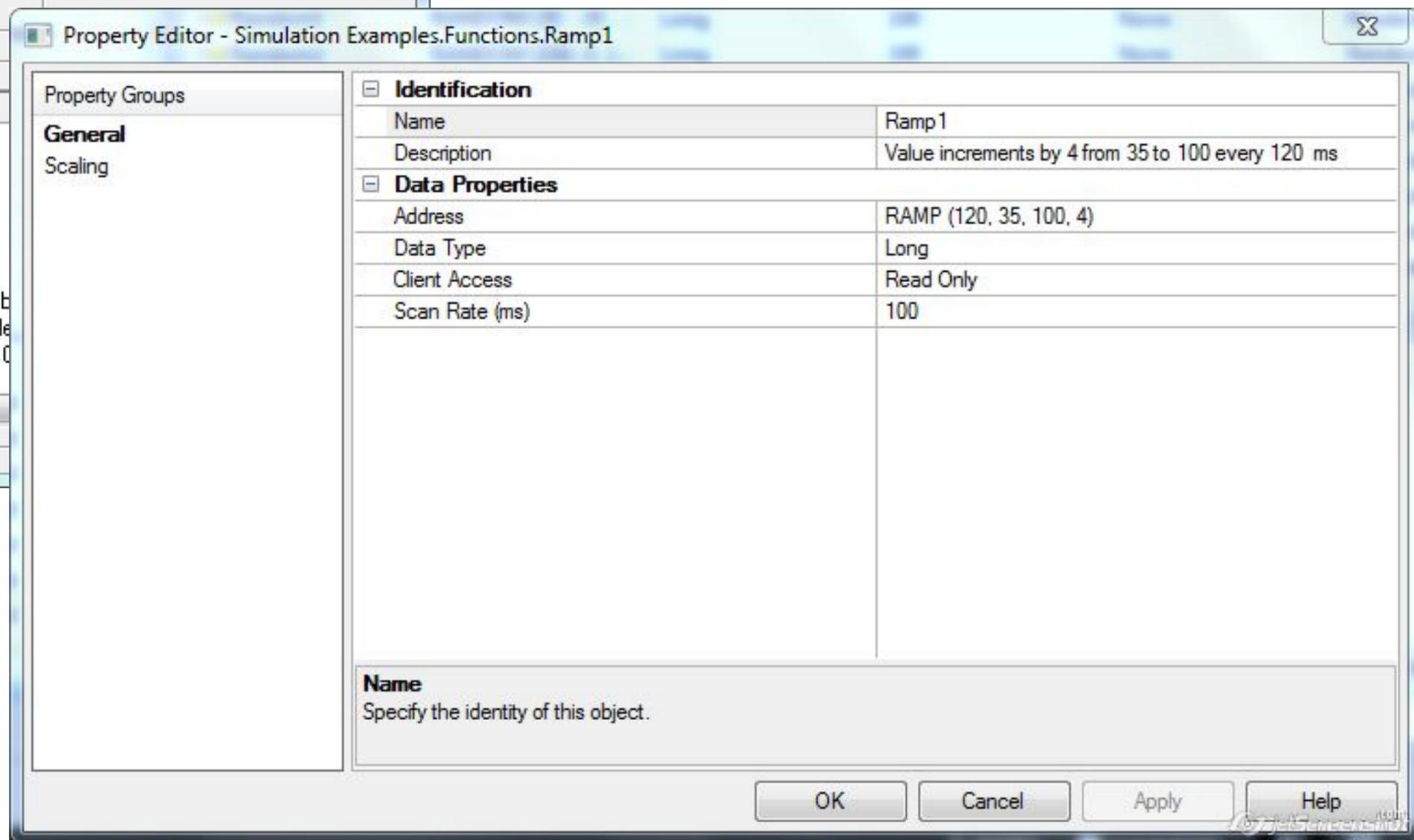
Item Count: 298

19 OPC Сервер

Keeware Server



Свойства Tag Item



ClientAce

Streamlines OPC connectivity through the
Microsoft .NET API

21 OPC Клиент

Kepware ClientACE + Visual Studio C# WinForm application

Объявить глобальные переменные

```
namespace ClientACEWinForm
{
    public partial class FormMain : Form
    {
        DaServerMgt DAServer = new DaServerMgt();
        ConnectInfo connectInfo = new ConnectInfo();
        public Boolean isOPCConnectionFiled = false;
    }
}
```

Установка

opcda://localhost/Kepware.KEPServerEX.V6/

```
private void buttonConnect_Click(object sender, EventArgs e)
{
    if (DAServer.ServerState == ServerState.DISCONNECTED)
    {
        connectInfo.LocalId = "en";
        connectInfo.KeepAliveTime = 1000;
        connectInfo.RetryAfterConnectionError = true;
        connectInfo.RetryInitialConnection = false;
        connectInfo.ClientName = "ClientAce";

        try
        {
            DAServer.Connect("opcda://localhost/Kepware.KEPServerEX.V6/", clientHandle, ref connectInfo, out isOPCConnectionFiled);
        }
        catch (Exception E)
        {
            MessageBox.Show(E.ToString());
        }
    }
}
```

22 OPC Клиент

Kepware ClientACE + Visual Studio C# WinForm application

Добавить метод Connect

The screenshot displays the Visual Studio IDE with the following components:

- Code Editor:** Shows the source code for `FormMain.cs`. The code includes several `using` statements and a partial class definition for `FormMain`. Lines 10 and 11 are highlighted with a red box:

```
10 using System.Windows.Forms;
11 using Kepware.ClientAce.OpcDaClient;
```
- Solution Explorer:** Shows the project structure. Under the `References` folder, `Kepware.ClientAce.OpcClient` and `Microsoft.CSharp` are highlighted with a red box. Other references include `System`, `System.Core`, `System.Data`, `System.Data.DataSetExtensions`, `System.Deployment`, `System.Drawing`, `System.Windows.Forms`, and `System.Xml`.
- Properties Window:** Shows the `FormMain.cs` File Properties.

23 OPC Клиент

Kepware ClientACE + Visual Studio C# WinForm application

Подписка на события

```
private void buttonSubscribe_Click(object sender, EventArgs e)
{
    if (DAServer.ServerState == ServerState.CONNECTED)
    {
        try
        {
            SubscribeToOPCDAServerEvents();
            SubscribeToData();
            ModifySubscription(true);
        }
        catch (Exception E)
        {
            MessageBox.Show(E.ToString());
        }
    }
}
```

24 OPC Клиент

Keeware ClientACE + Visual Studio C# WinForm application

Подписка на события

```
private void SubscribeToOPCDAServerEvents()
{
    DAServer.DataChanged += new
Keeware.ClientAce.OpcDaClient.DaServerMgt.DataChangedEventHandler
(DAServer_DataChanged);
}

public void DAServer_DataChanged(int clientSubscription, bool
allQualitiesGood, bool noErrors, ItemValueCallback[] ItemValues)
{
    object[] DCevHndlrArray = new object[4];
    DCevHndlrArray[0] = clientSubscription;
    DCevHndlrArray[1] = allQualitiesGood;
    DCevHndlrArray[2] = noErrors;
    DCevHndlrArray[3] = ItemValues;
    BeginInvoke(new DaServerMgt.DataChangedEventHandler(DataChanged),
DCevHndlrArray);
}
```

Подписка на данные

```
private void SubscribeToData()
{
    bool active = false;
    int updateRate = 1000;
    Single deadBand = 0;
    int revisedUpdateRate;

    itemIdentifiers = new ItemIdentifier[2];

    itemIdentifiers[0] = new ItemIdentifier();
    itemIdentifiers[0].ClientHandle = 1;
    itemIdentifiers[0].DataType = Type.GetType("System.Double");
    itemIdentifiers[0].ItemName = "Simulation Examples.Functions.Ramp1";

    itemIdentifiers[1] = new ItemIdentifier();
    itemIdentifiers[1].ClientHandle = 2;
    itemIdentifiers[1].DataType = Type.GetType("System.Double");
    itemIdentifiers[1].ItemName = "Simulation Examples.Functions.Ramp2";

    try
    {
        DAServer.Subscribe(clientSubscription, active, updateRate, out revisedUpdateRate, deadBand,
ref itemIdentifiers, out serverSubscription);
    }
    catch (Exception E)
    {
        MessageBox.Show(E.ToString());
    }
}
```

26 OPC Клиент

Kepware ClientACE + Visual Studio C# WinForm application

Обновление состояния подписки

```
private void ModifySubscription(bool action)
{
    DAServer.SubscriptionModify(serverSubscription, action);
}
```

27 OPC Клиент

Kepware ClientACE + Visual Studio C# WinForm application

Обработка события OnDataChanges

```
public void DataChanged(int clientSubscription, bool allQualitiesGood, bool noErrors,
ItemValueCallback[] ItemValues)
{
    foreach (ItemValueCallback item in ItemValues)
    {
        switch ((int)item.ClientHandle)
        {
            case 1:
                textBoxRamp1.Text = item.Value.ToString();
                textBoxRamp1TimeStamp.Text = item.TimeStamp.ToString();
                textBoxRamp1Quality.Text = item.Quality.ToString();
                break;
            case 2:
                textBoxRamp2.Text = item.Value.ToString();
                textBoxRamp2TimeStamp.Text = item.TimeStamp.ToString();
                textBoxRamp2Quality.Text = item.Quality.ToString();
                break;
        }
    }
}
```

28 OPC Клиент

Keeware ClientACE + Visual Studio C# WinForm application

WinForm in action

Identifier() (+ 2 overload(s))
act with default values.

Ramp1	Ramp2
95	152.5
2/21/2017 9:36:14 AM	2/21/2017 9:36:14 AM
OPC_QUALITY_GOOD	OPC_QUALITY_GOOD

CONNECTED

Connect Subscribe Disconnect

Demo Mode

ClientAce® is running in DEMO mode.

License information:

Release-Mode:
To get a fully functional release version of your application you have to digitally sign your application with the ClientAce® Sign Tool. To sign your application you need a valid license for ClientAce® on your development computer.

Demo-Mode:
You can continue to use ClientAce® without a valid license. It's fully functional, but pops up this message during demo time. After the demo time has expired you need to restart the application to reset the demo time.

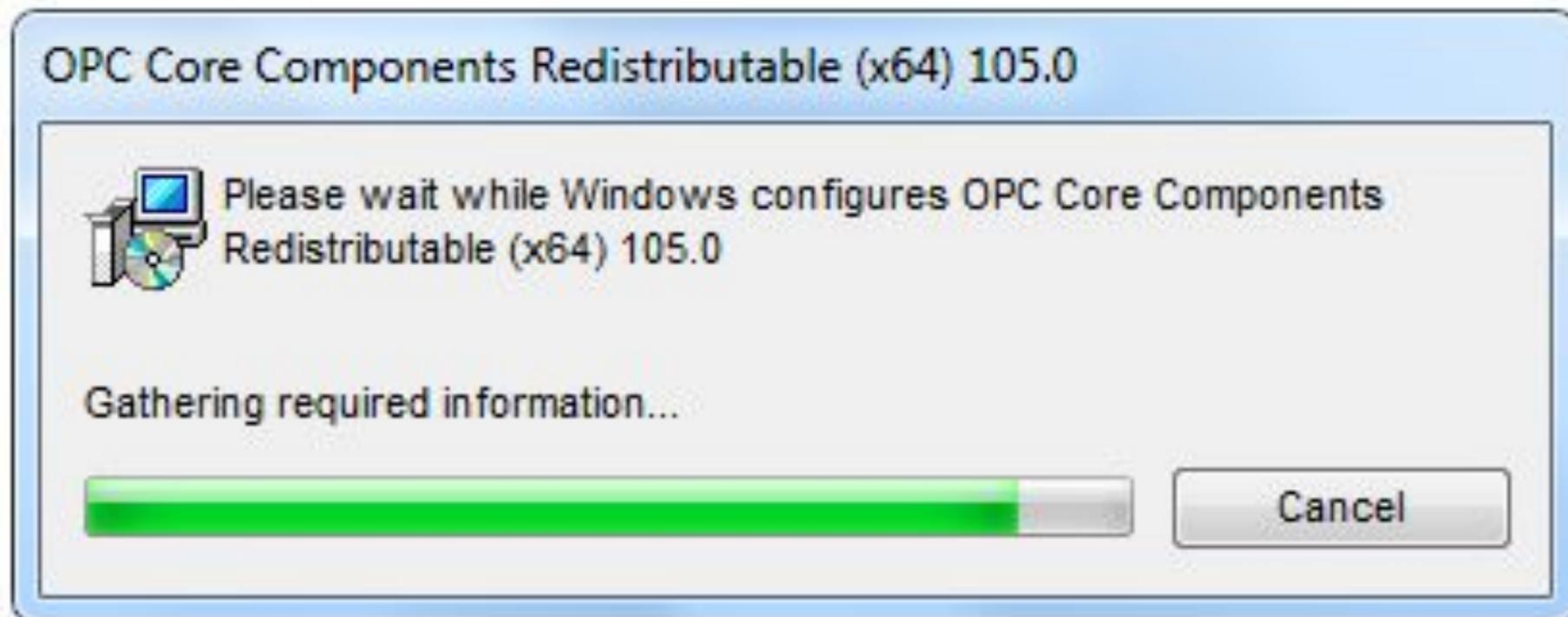
Continue

29 OPC Клиент

Matlab + OPC

Установка пакета

```
Command Window  
fx >> opcregister('-silent')
```



OPC Client



Проверка установки

```
>> sInfo = opcserverinfo('localhost')

sInfo =

    Host: 'localhost'
  ServerID: {'EEI.OPCSimulator' 'Kepware.KEPServerEX.V6' 'OPCSystems.NET.1'}
ServerDescription: {'OPCSimulator' 'KEPServerEX 6.0' 'OPCSystems.NET DA Server V1.0'}
  OPCSification: {'DA2' 'DA2' 'DA2'}
ObjectConstructor: {'opcda('localhost', 'EEI.OPCSimulator')' [1x44 char] [1x38 char]}
```

31 OPC Клиент

Matlab + OPC

Установка соединения

```
da = opcda('localhost', 'Kepware.KEPServerEX.V6')  
connect(da)
```

```
>> da
```

```
da =
```

```
Summary of OPC Data Access Client Object: localhost/Kepware.KEPServerEX.V6
```

Server Parameters

```
Host      : localhost  
ServerID  : Kepware.KEPServerEX.V6  
Status    : connected  
Timeout   : 10 seconds
```

Object Parameters

```
Group     : 0-by-1 dagroup object  
Event Log : 0 of 1000 events
```

Добавление группы тегов для мониторинга

```
grp = addgroup(da, 'MonitoringGroup')

itmIDs = {'Simulation Examples.Functions.Ramp1', ...
          'Simulation Examples.Functions.Sine1', ...
          'Simulation Examples.Functions.Random1'};

itm = additem(grp, itmIDs)
```

Настройка параметров мониторинга

```
logDuration = 60;  
logRate = 1;  
numRecords = ceil(logDuration./logRate)  
grp.UpdateRate = logRate;  
grp.RecordsToAcquire = numRecords;
```

33 OPC Клиент

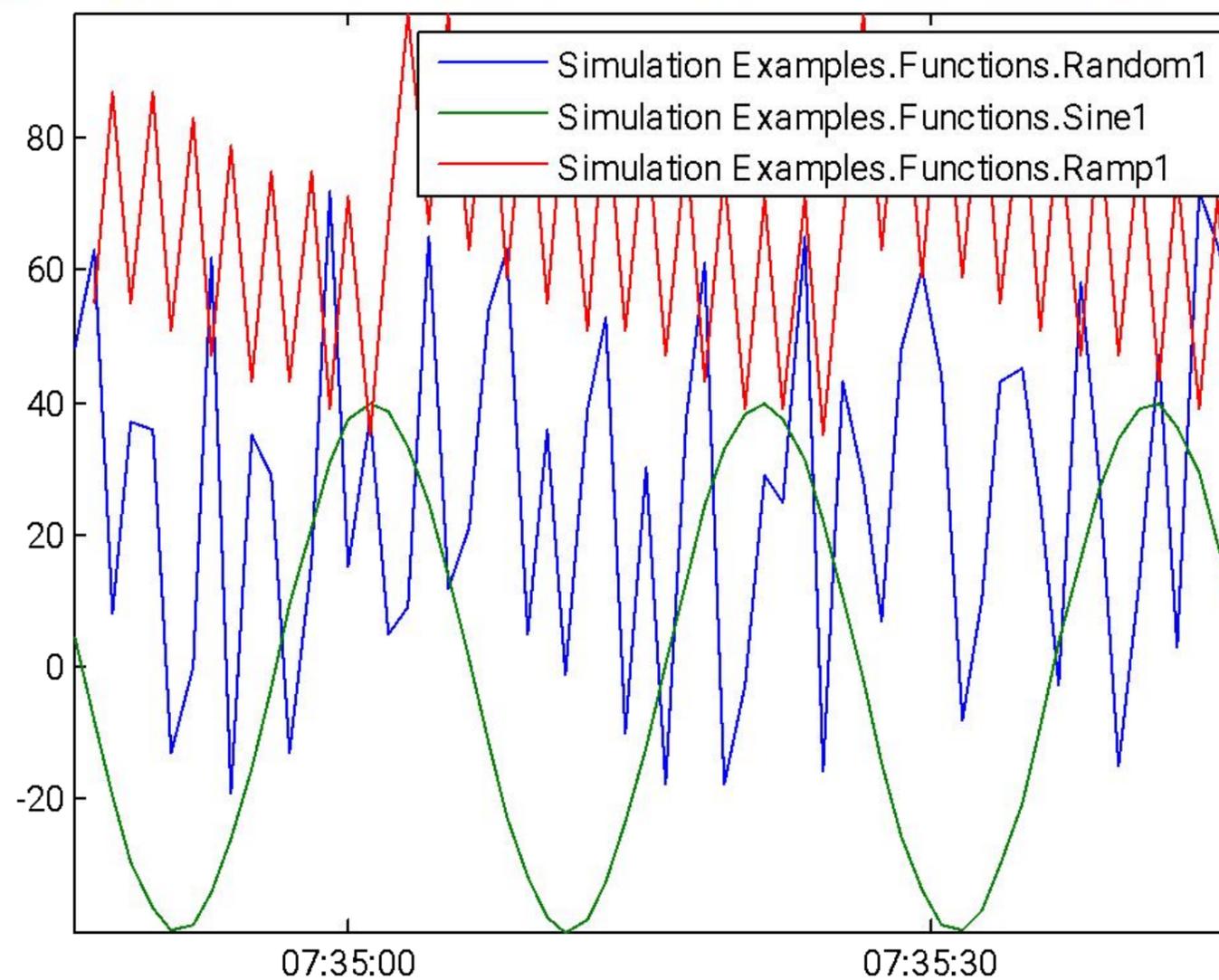
Matlab + OPC

Запуск мониторинга

```
start (grp)
wait (grp)
[logIDs, logVal, logQual, logTime, logEvtTime] = getdata (grp, 'double');
```

Результаты мониторинга

```
start (grp)  
wait (grp)  
[logIDs, logVal, logQual, logTime, logEvtTime] = getdata (grp, 'double');
```



Заккрытие соединения

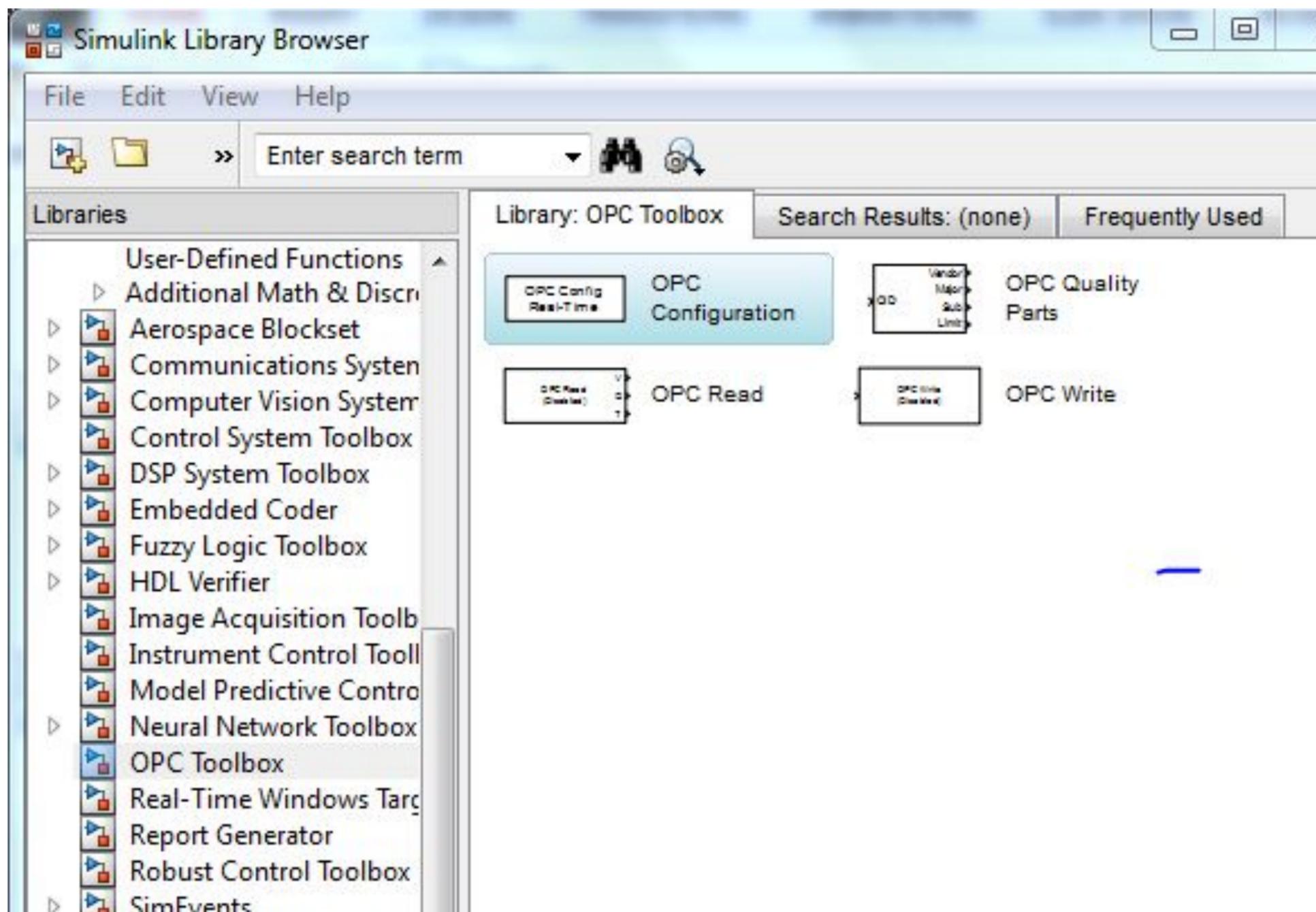
```
disconnect(da)  
delete(da)
```

Недостаток метода – offline режим сбора данных

37 OPC Клиент

Matlab + OPC

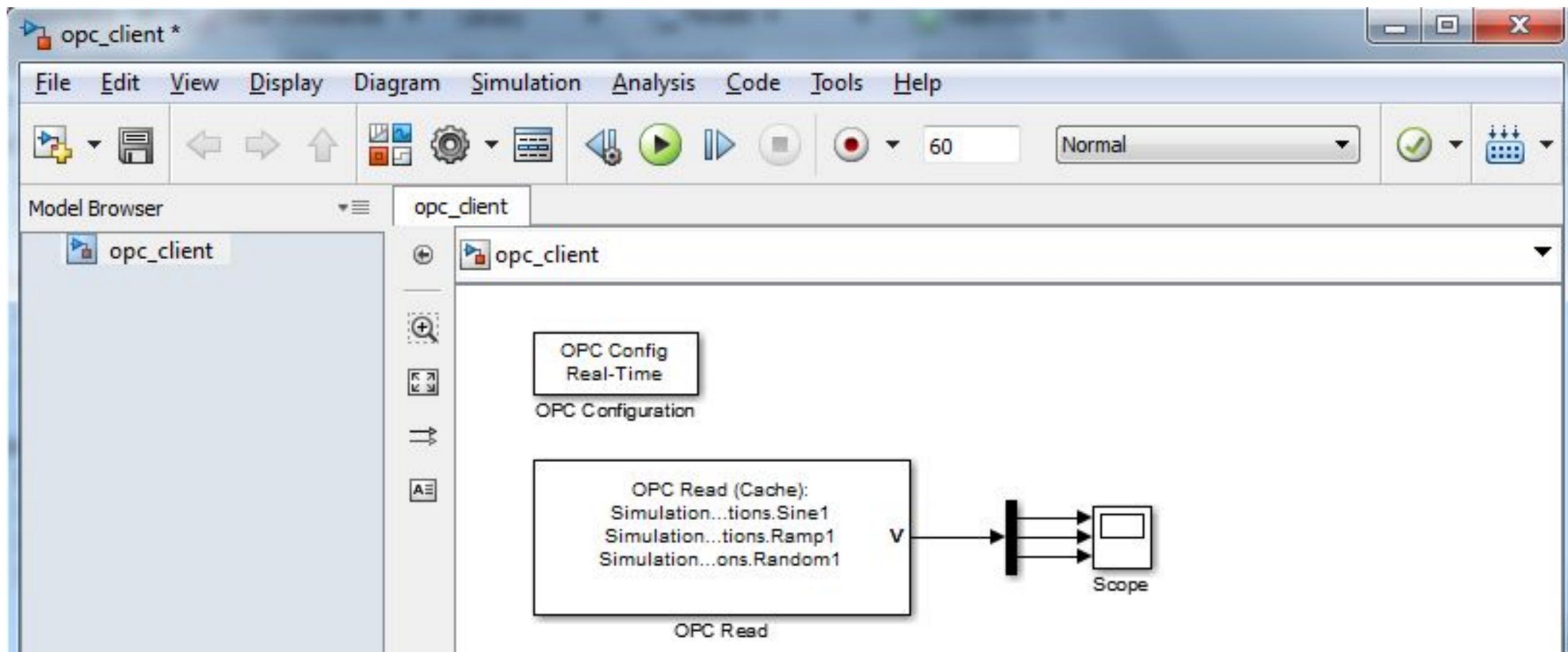
Применение Simulink модели в качестве OPC клиента



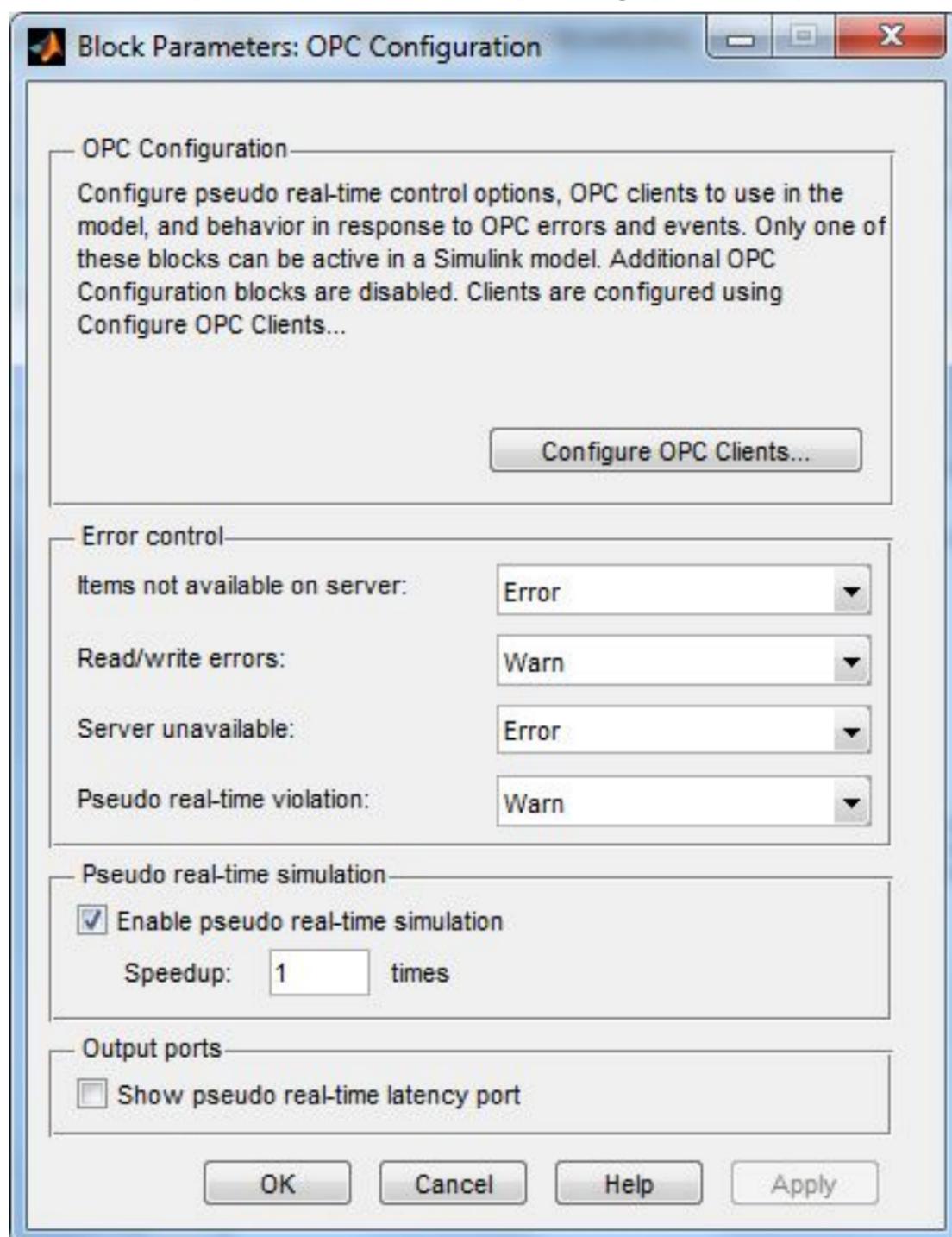
38 OPC Клиент

Matlab + OPC

Создание модели



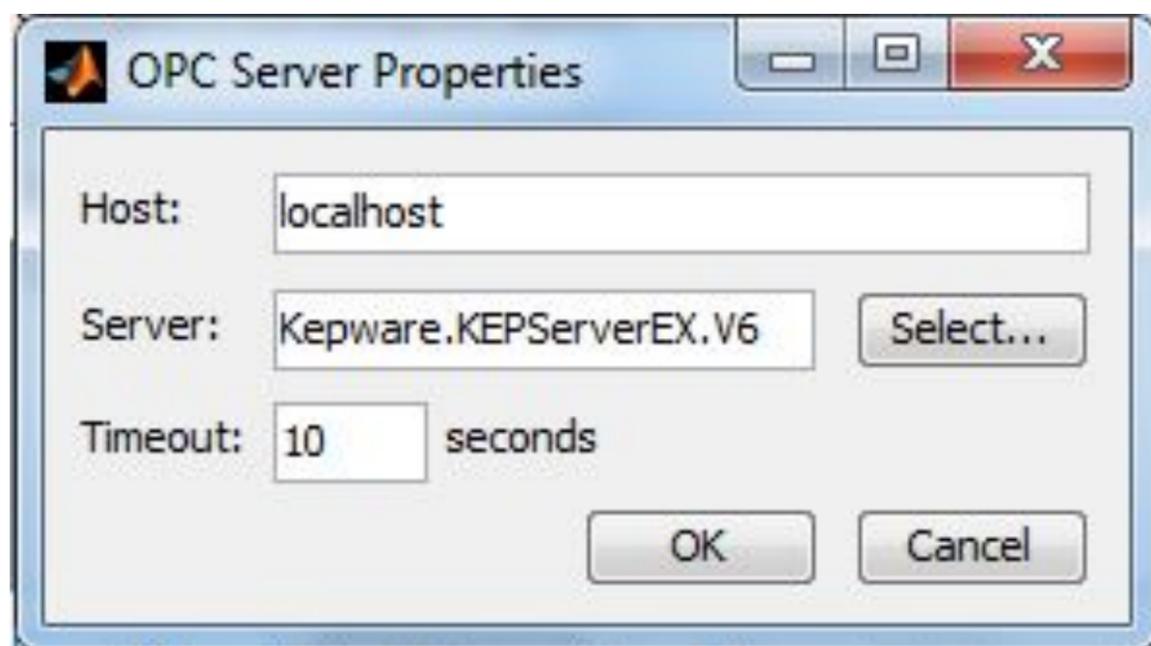
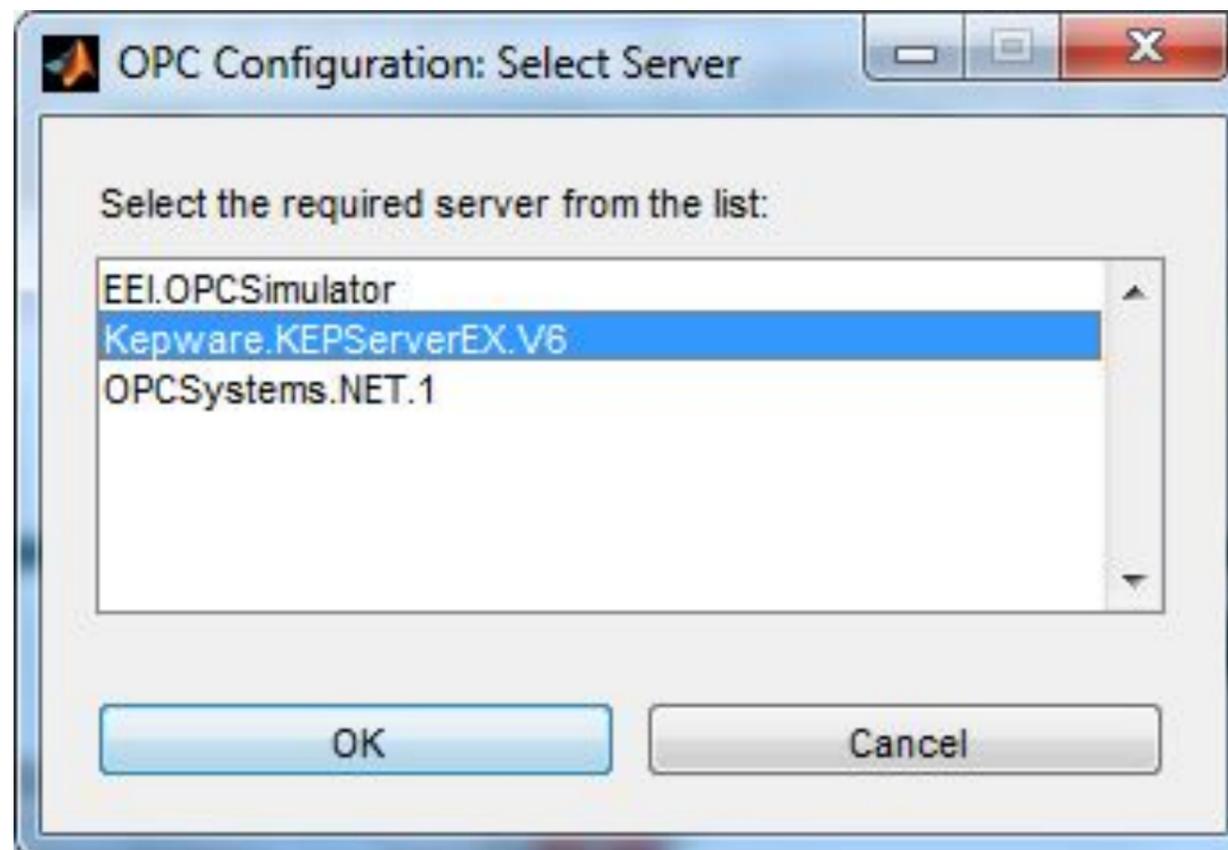
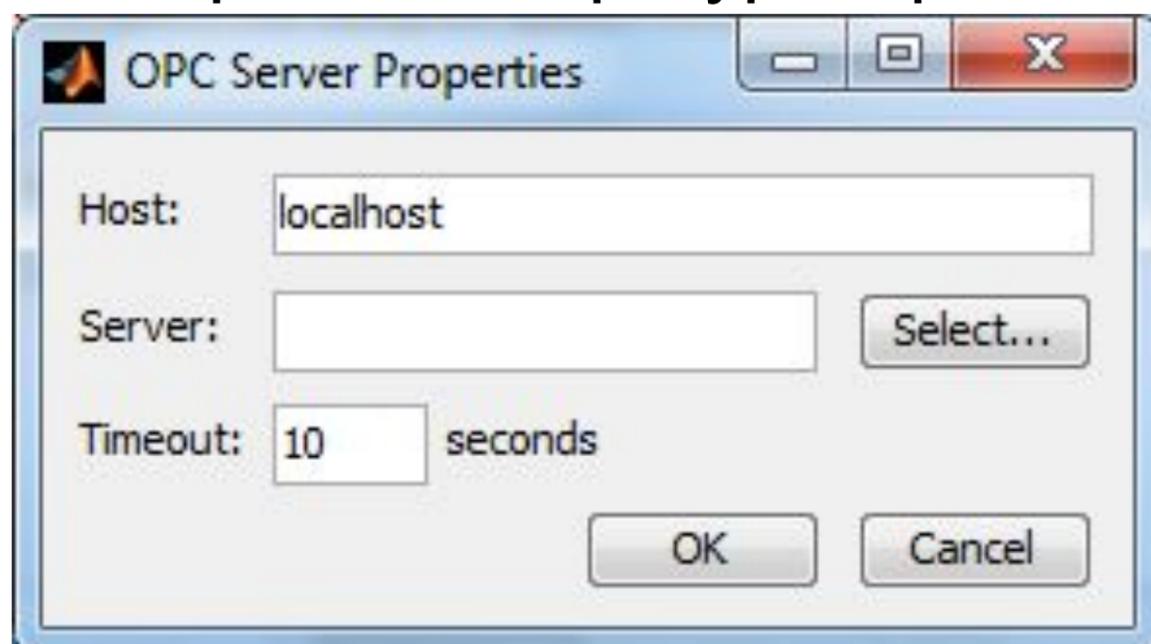
Настройка конфигуратора



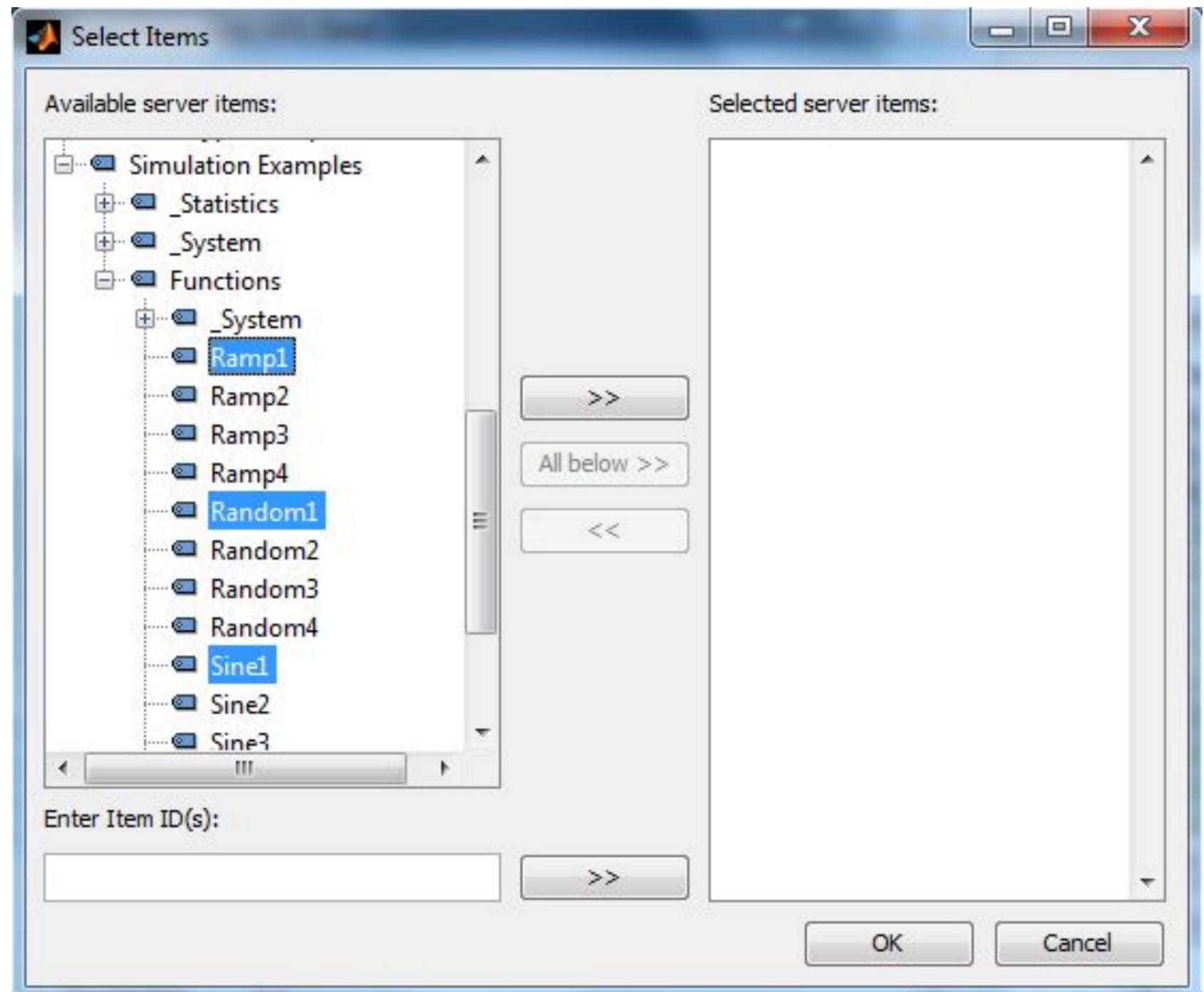
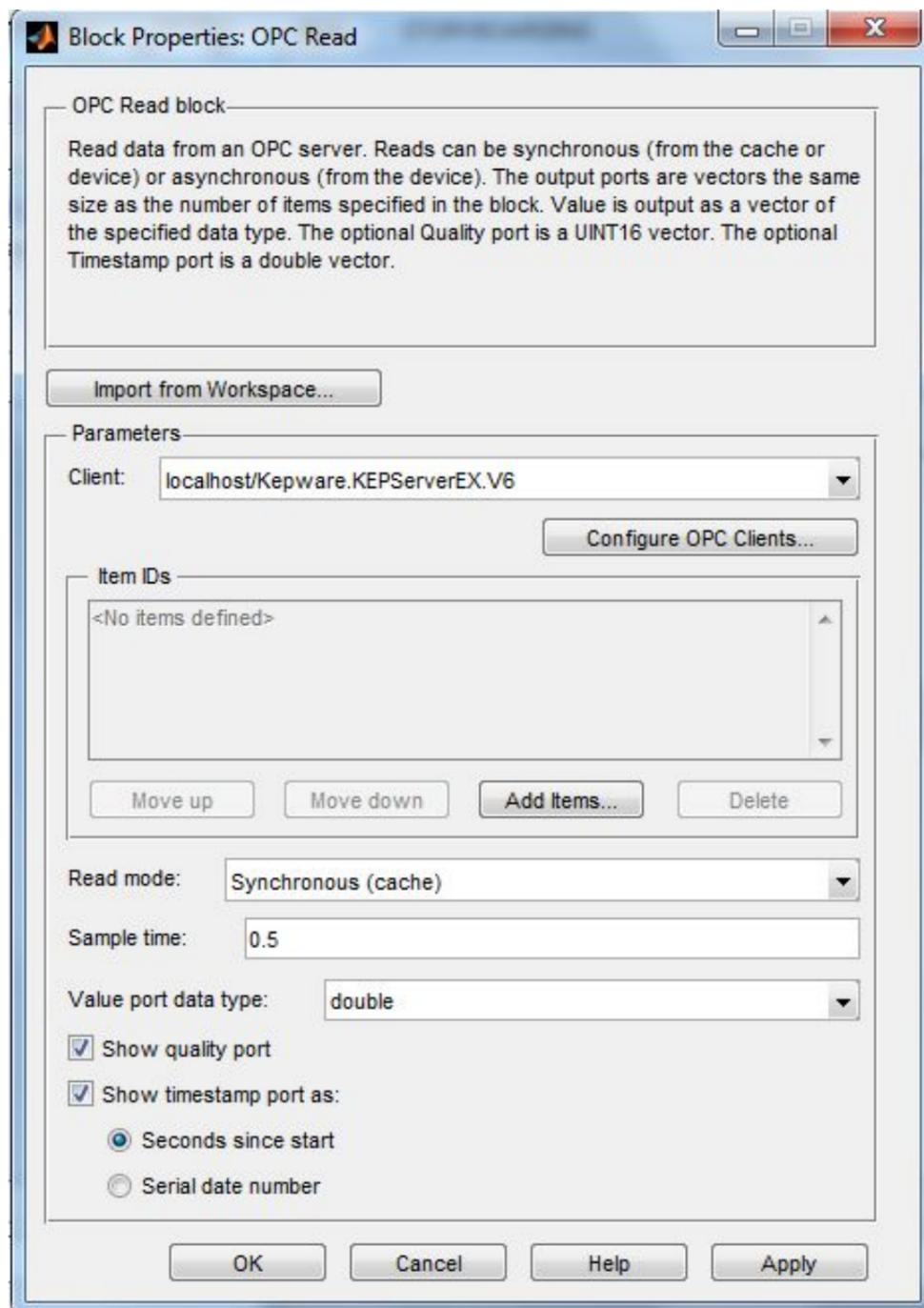
40 OPC Клиент

Matlab + OPC

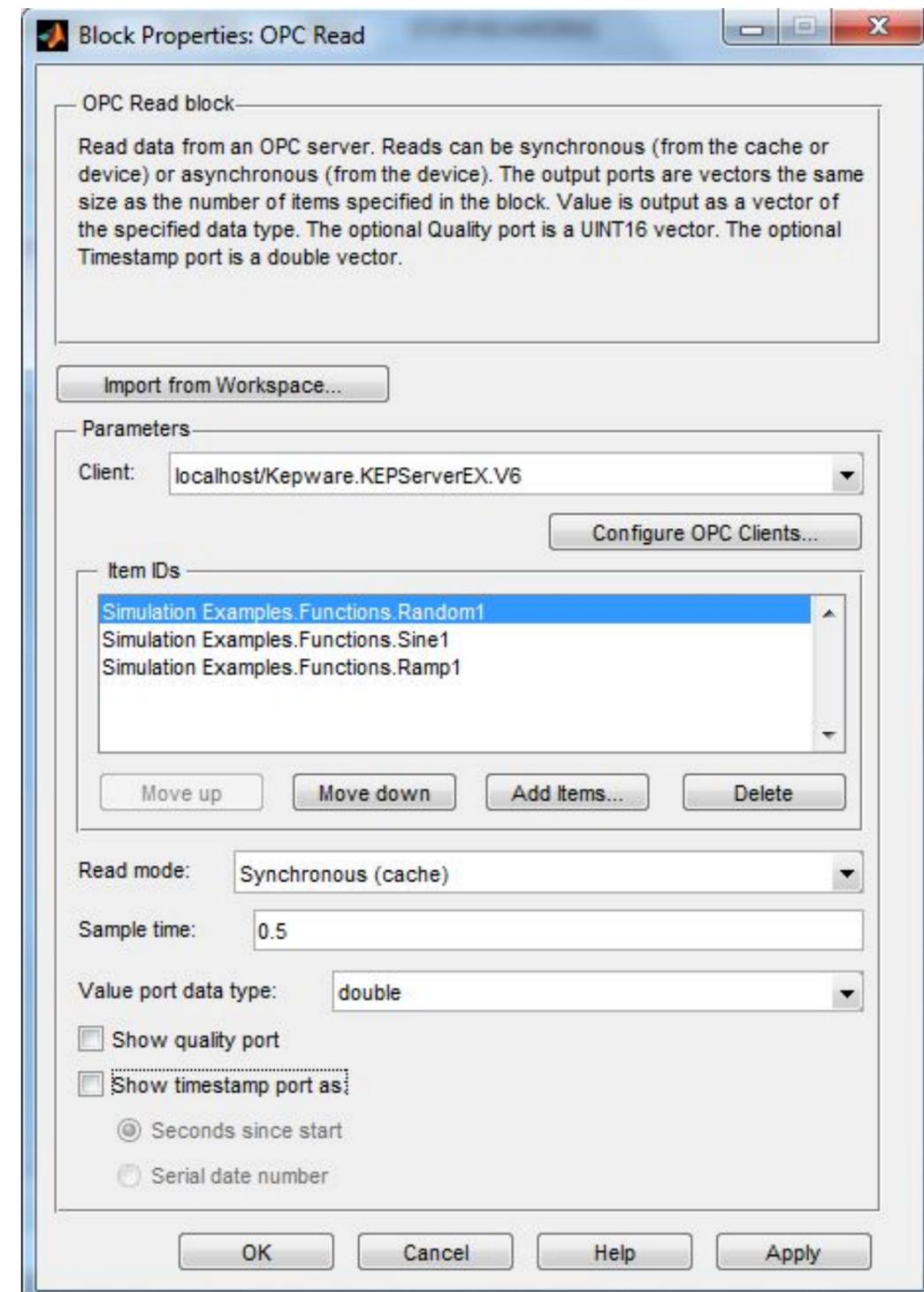
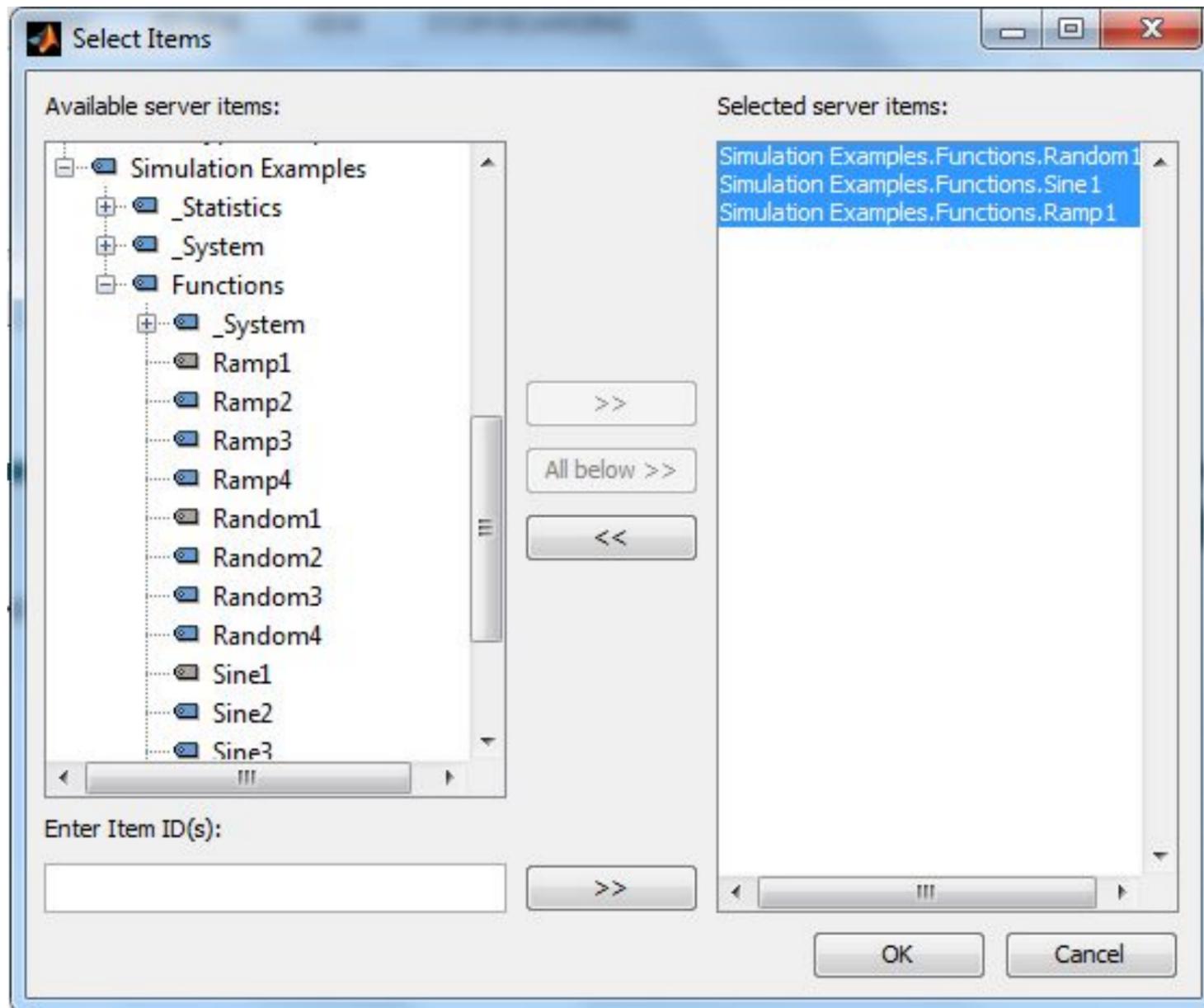
Настройка конфигуратора



Подключение к тегам



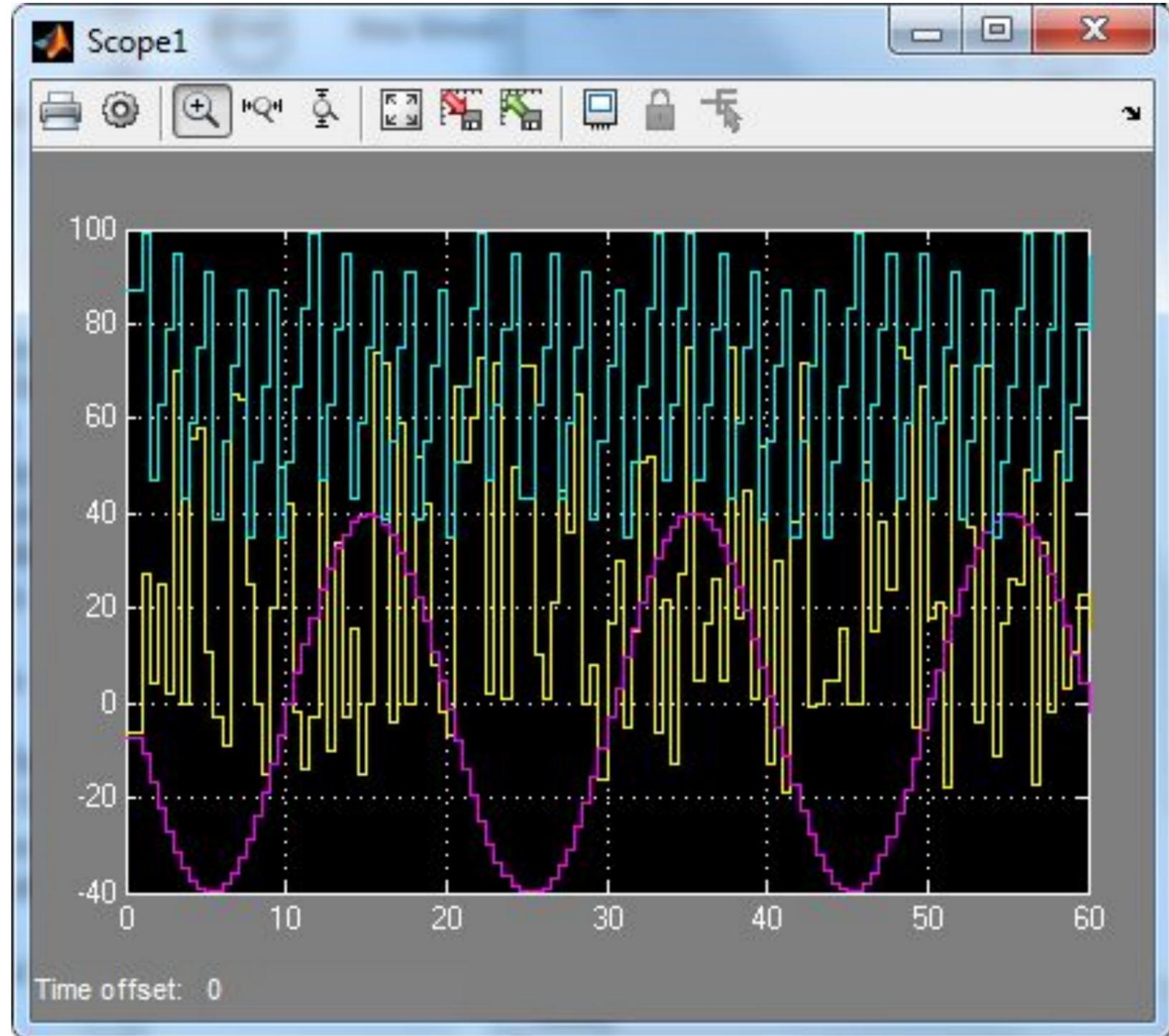
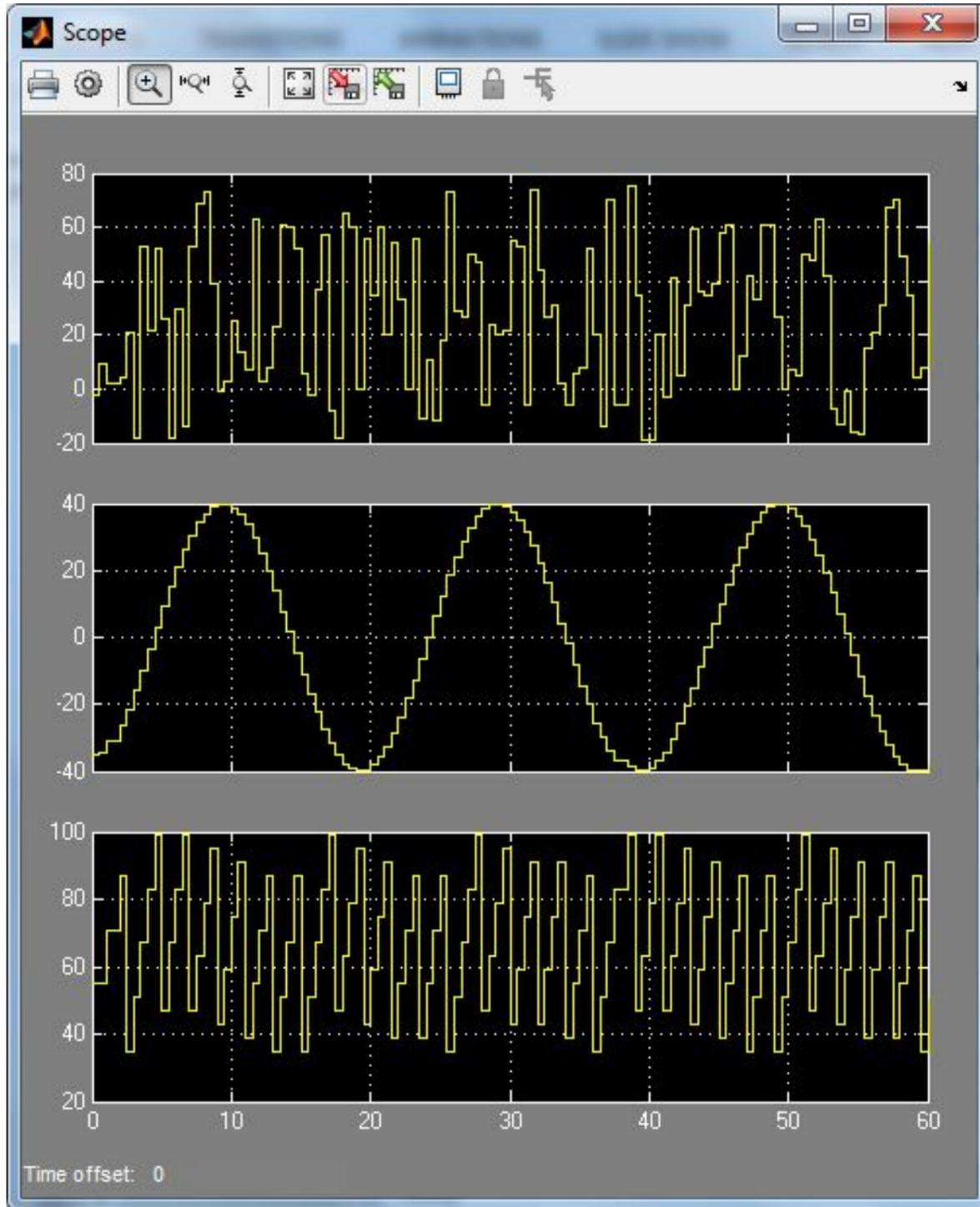
Подключение к тегам



43 OPC Клиент

Matlab + OPC

Запуск модели



44 OPC Клиент

Matlab + OPC

Создание тега для записи

The screenshot displays the KEPServerEX 6 Configuration software interface. On the left, a tree view shows the project structure under 'Connectivity', with 'New Channel' selected. On the right, a table lists existing channels:

Channel Name	Driver
Channel1	Simulator
Data Type Exam...	Simulator
Simulation Exam...	Simulator

The 'Add Channel Wizard' dialog box is open, showing a list of channel types. The 'Simulator' option is selected. The list includes:

- ABB Totalflow
- Opto 22 Ethernet
- Partlow ASCII
- Philips P8/PC20
- Ping
- SattBus Ethernet
- SattBus Serial
- Scanivalve Ethernet
- Siemens S5 (3964R)
- Siemens S5 (AS511)
- Siemens S7 MPI
- Siemens S7-200
- Siemens TCP/IP Ethernet
- Siemens TCP/IP Slave Ethernet
- Simatic/TI 505 Ethernet
- Simatic/TI 505 Serial
- Simulator**
- SIXNET EtherTRAK
- SIXNET UDR
- SNMP

The 'Next' and 'Cancel' buttons are visible at the bottom of the wizard.

45 OPC Клиент

Matlab + OPC

Создание тега для записи

Specify the identity of this object.

Name:

 ?

Next Cancel

Choose how write data is passed to the underlying communications driver when more than one write exists in the write queue.

Optimization Method:

 ?

Specify the ratio of write operations to read operations, based on one read per configurable number of writes.

Duty Cycle:

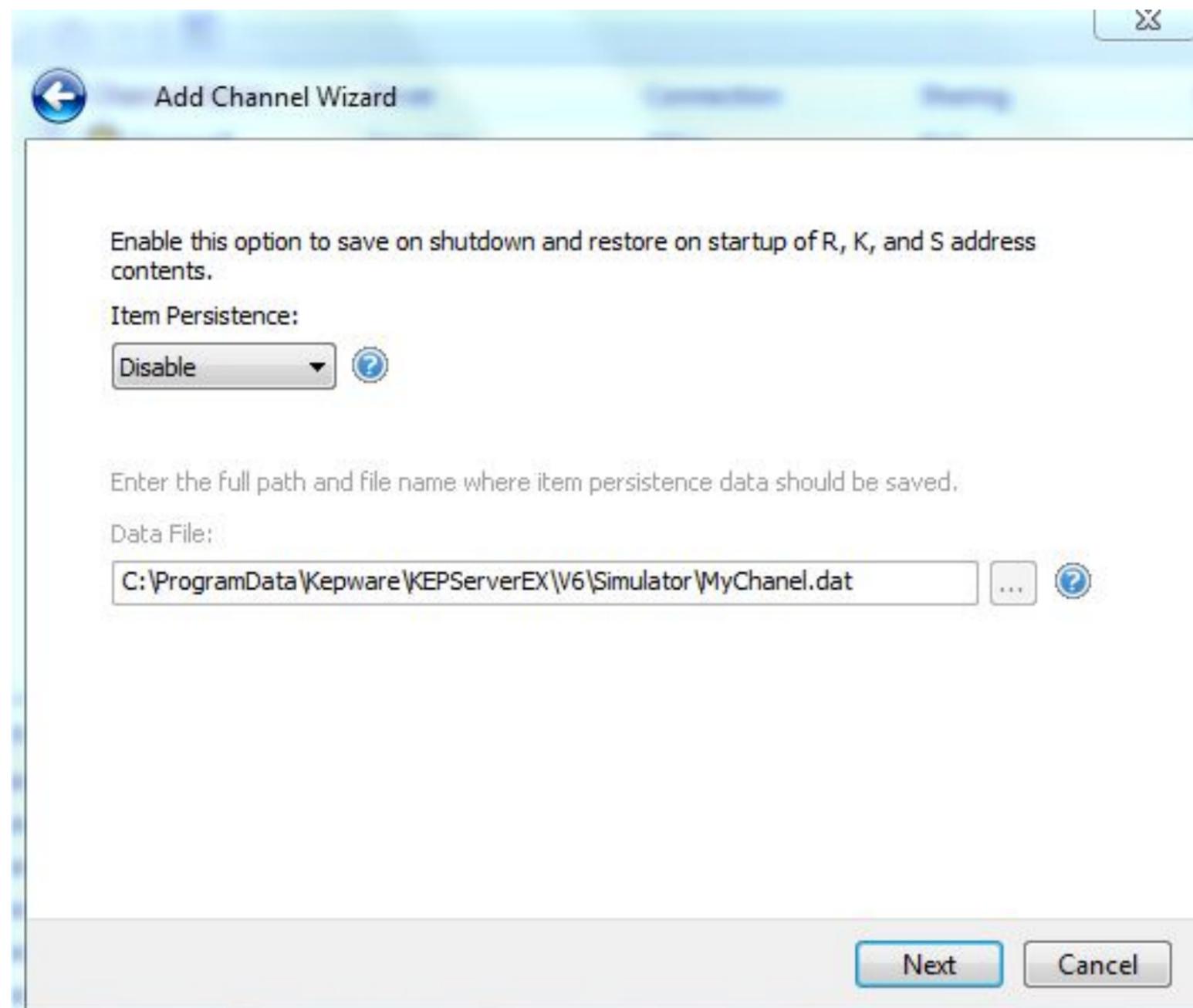
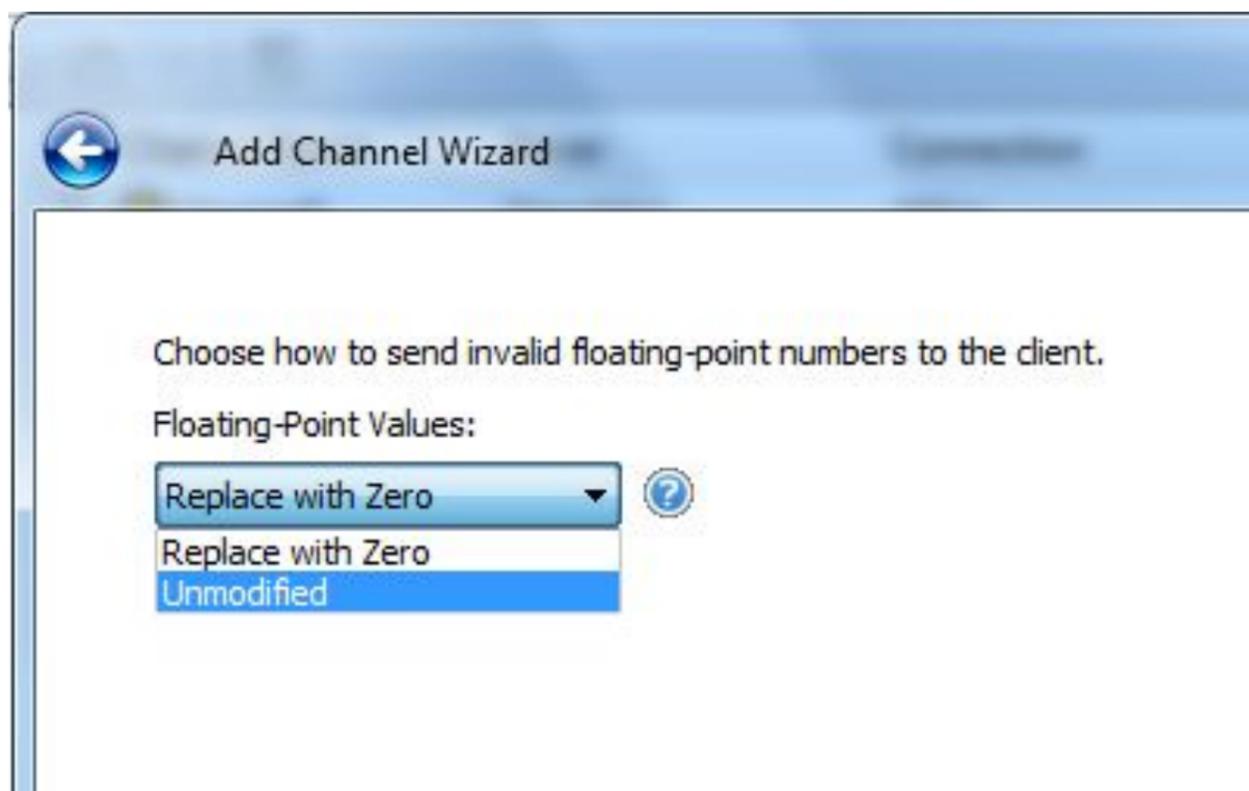
 ?

Next Cancel

46 OPC Клиент

Matlab + OPC

Создание тега для записи



47 OPC Клиент

Matlab + OPC

Создание тега для записи

The screenshot shows the 'Add Channel Wizard' dialog box. The title bar reads 'Add Channel Wizard'. The dialog is divided into several sections, each with a collapse icon on the left:

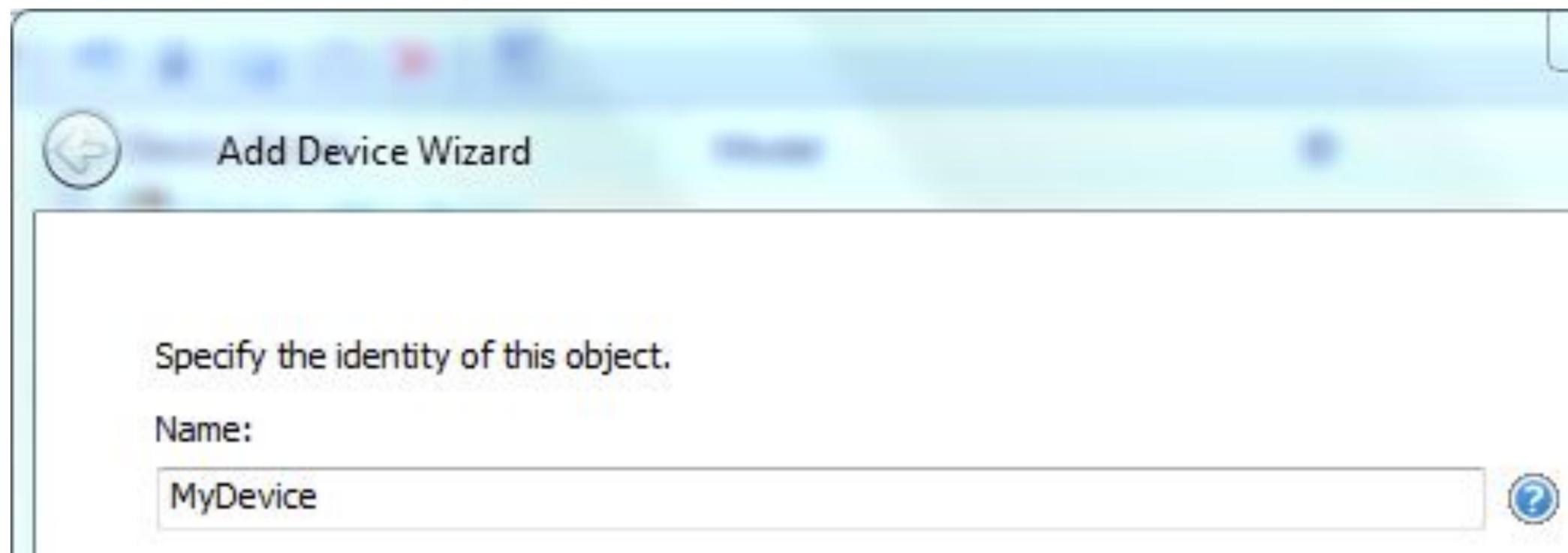
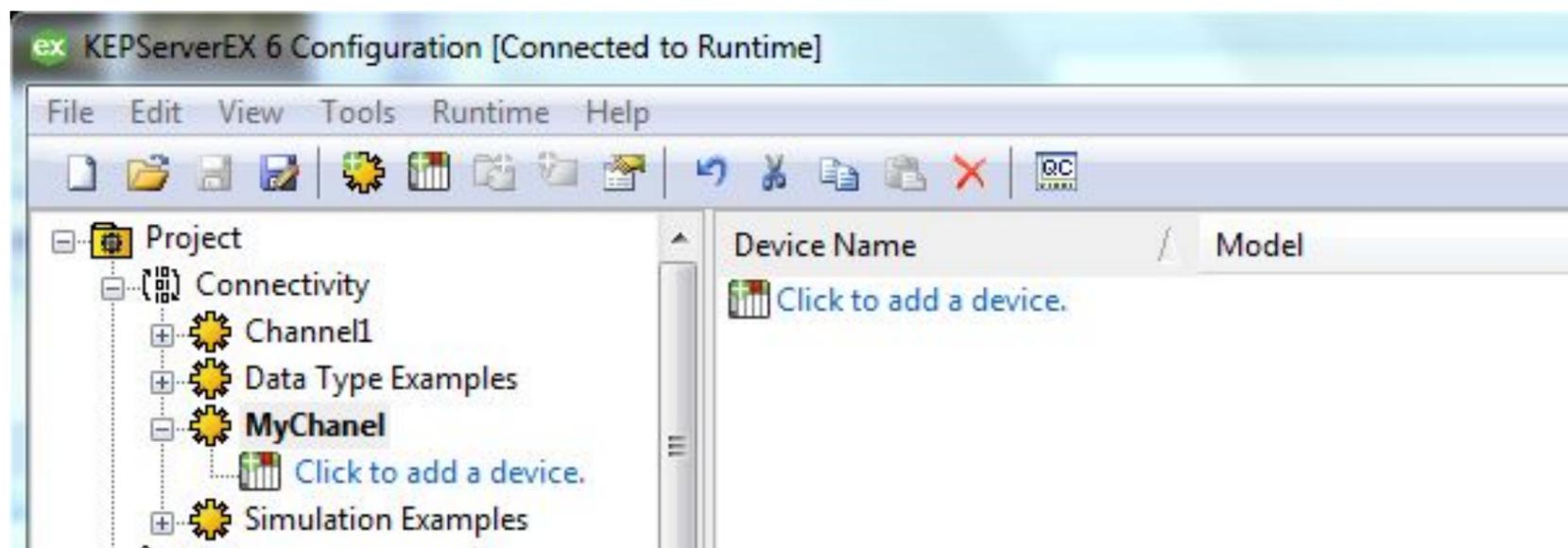
- Identification**
 - Name: MyChanel
 - Description: (empty)
 - Driver: Simulator
- Diagnostics**
 - Diagnostics Capture: Disable
- Write Optimizations**
 - Optimization Method: Write Only Latest Value for All Tags
 - Duty Cycle: 10
- Non-Normalized Float Handling**
 - Floating-Point Values: Unmodified
- Persistence**
 - Item Persistence: Disable

At the bottom right of the dialog, there are two buttons: 'Finish' and 'Cancel'.

48 OPC Клиент

Matlab + OPC

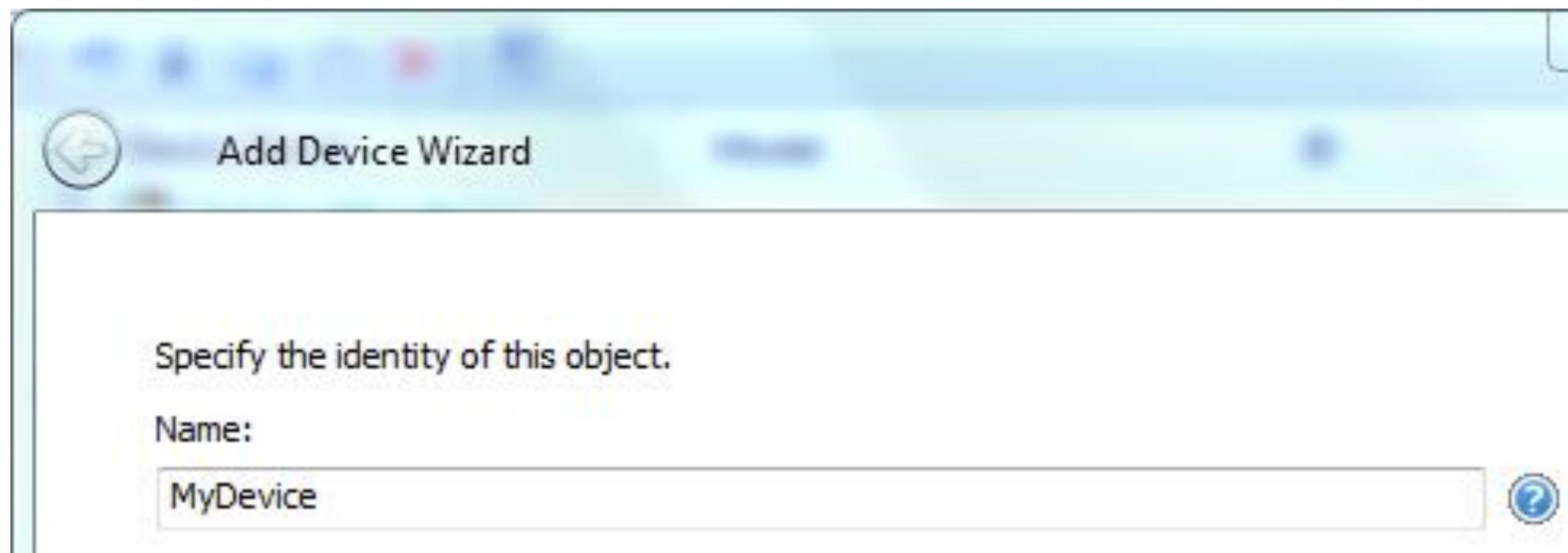
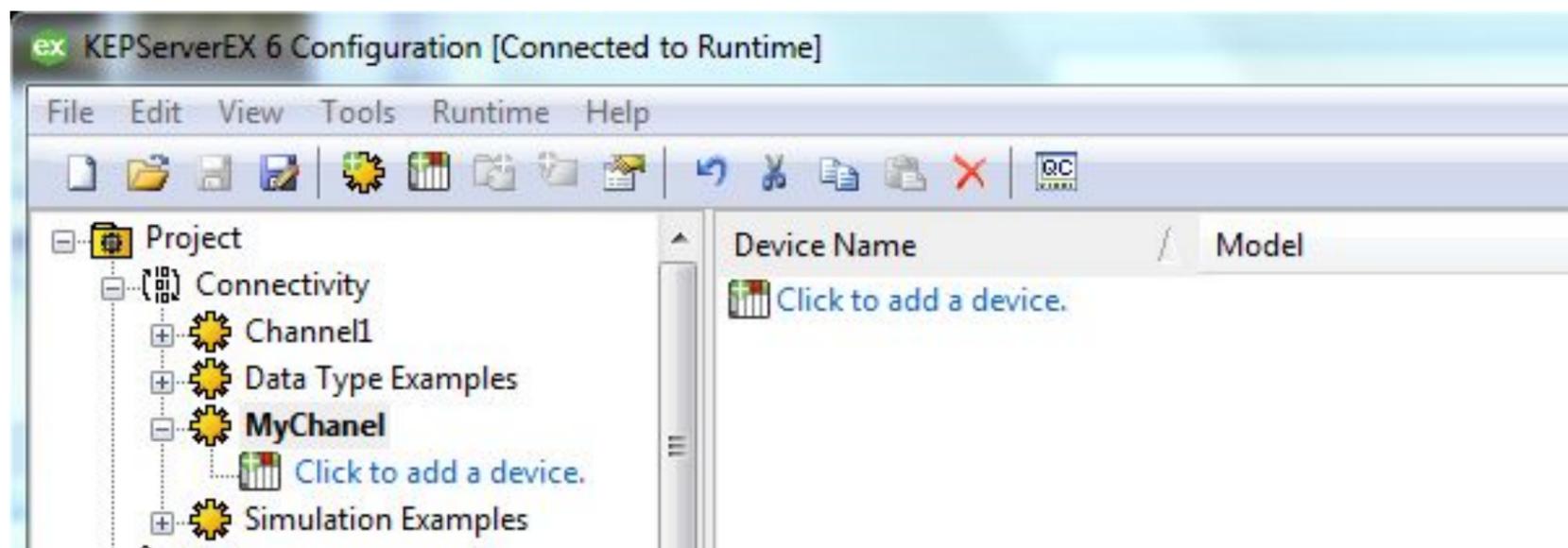
Создание тега для записи



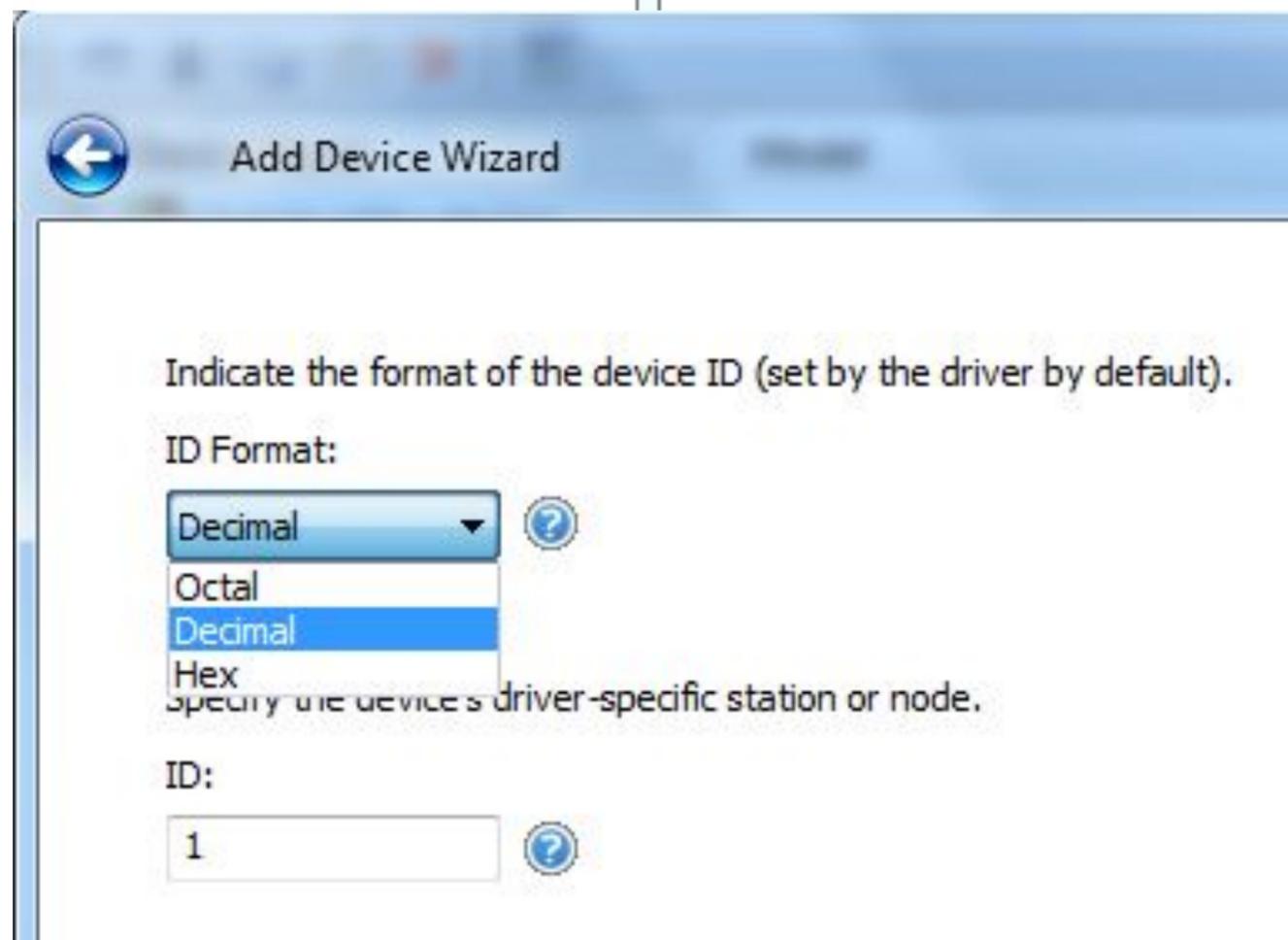
49 OPC Клиент

Matlab + OPC

Создание тега для записи



Создание тега для записи



51 OPC Клиент

Matlab + OPC

Создание тега для записи

Specify the method for determining how often tags in the device are scanned.

Scan Mode:

- Respect Client-Specified Scan Rate
- Request Data No Faster than Scan Rate
- Request All Data at Scan Rate**
- Do Not Scan, Demand Poll Only
- Respect Tag-Specified Scan Rate

Initial Updates from Cache:

Disable

Identification	
Name	MyDevice
Description	
Channel Assignment	MyChanel
Driver	Simulator
Model	16 Bit Device
ID Format	Decimal
ID	1
Operating Mode	
Data Collection	Enable
Scan Mode	
Scan Mode	Respect Client-Specified Scan Rate
Initial Updates from Cache	Disable

Создание тега для записи

The screenshot displays the KEPServerEX 6 Configuration interface. The main window shows a project tree on the left with 'MyDevice' selected. The central pane shows a table for tag configuration:

Tag Name	Address	Data Type	Scan Rate
Click to add a static tag. Tags are not required, but are browsable by OPC clients.			

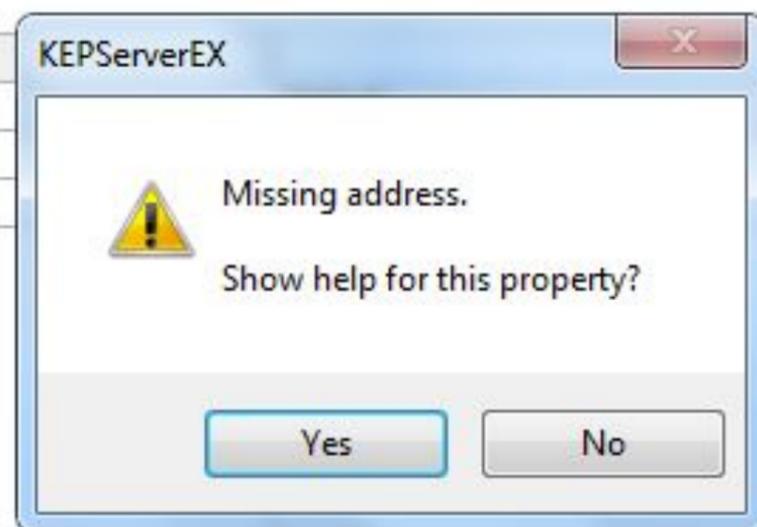
A 'Property Editor - MyChanel.MyDevice' dialog box is open, showing the configuration for a tag named 'MyTag'.

Property Groups	
General	
Scaling	
Identification	
Name	MyTag
Description	
Data Properties	
Address	
Data Type	Default
Client Access	Read/Write
Scan Rate (ms)	1000

At the bottom of the dialog, there is a text field for 'Name' with the instruction: 'Specify the identity of this object.' The dialog includes 'OK', 'Cancel', and 'Help' buttons.

Matlab + OPC

Создание тега для записи

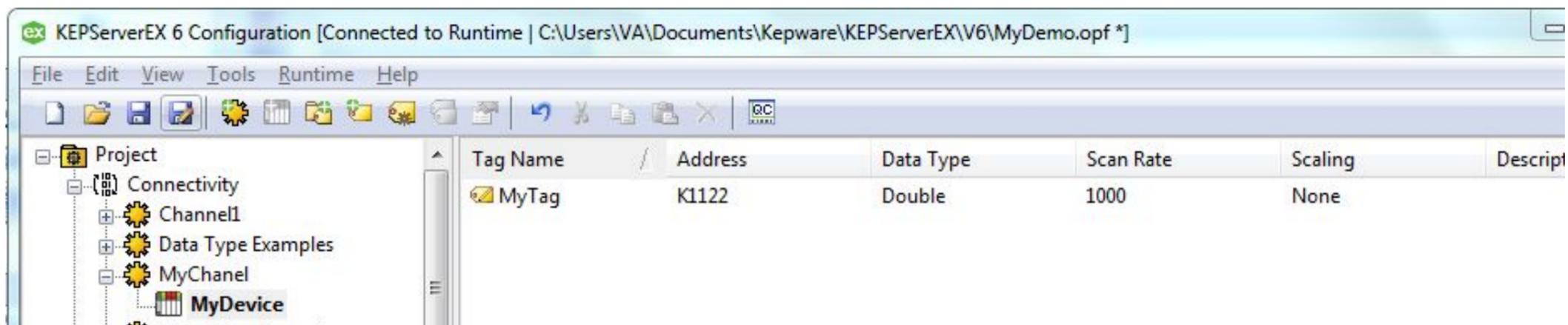
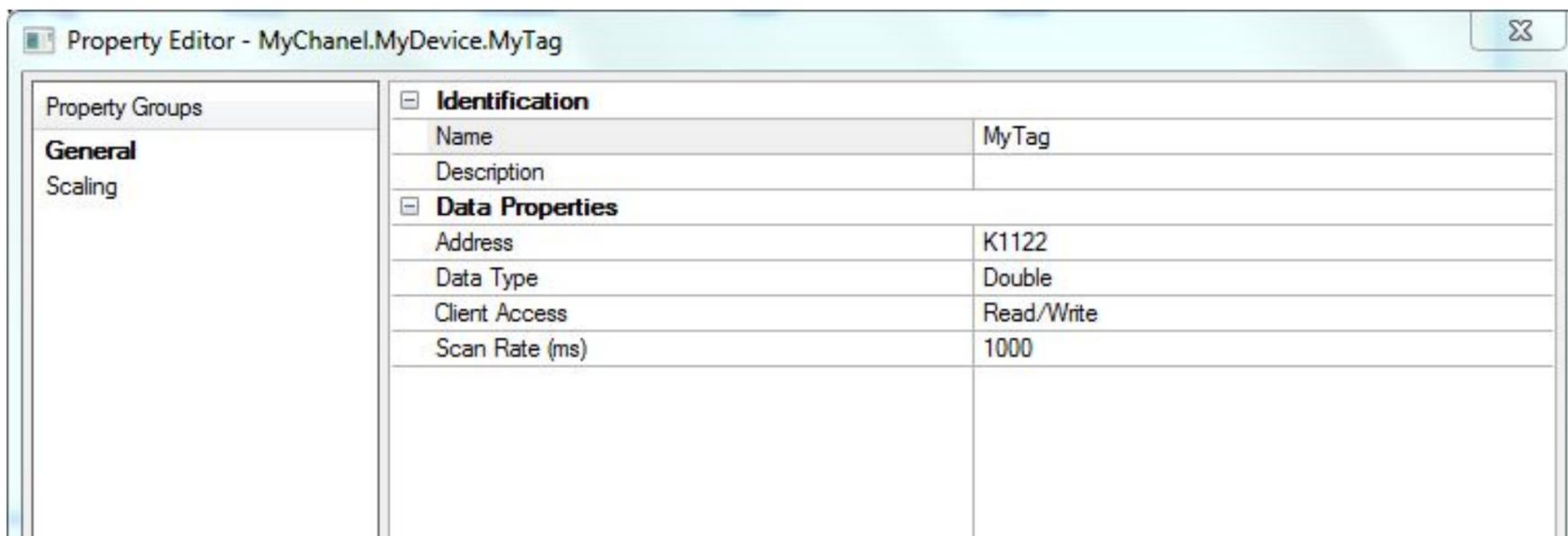


16-Bit Device Addresses

The memory configuration for the 16-Bit Device is simulated as a block of word locations numbered from 0 to 9999. Each word can be addressed as an offset from the start of the block. The default data types for each format are shown in **bold**.

Device Type	Range	Data Type	Access
Registers	R0000-R9999 R0000-R9998 R0000-R9996	Word , Short, BCD DWord, Long, LBCD, Float LLong, QWord, Double, Date, Boolean	Read/Write
Constants	K0000-K9999 K0000-K9998 K0000-K9996	Word , Short, BCD DWord, Long, LBCD, Float LLong, QWord, Double, Date, Boolean	Read/Write
Bits	R0000.00-R9999.15 K0000.00-K9999.15	Boolean	Read/Write
Strings	S000-S999	String	Read/Write

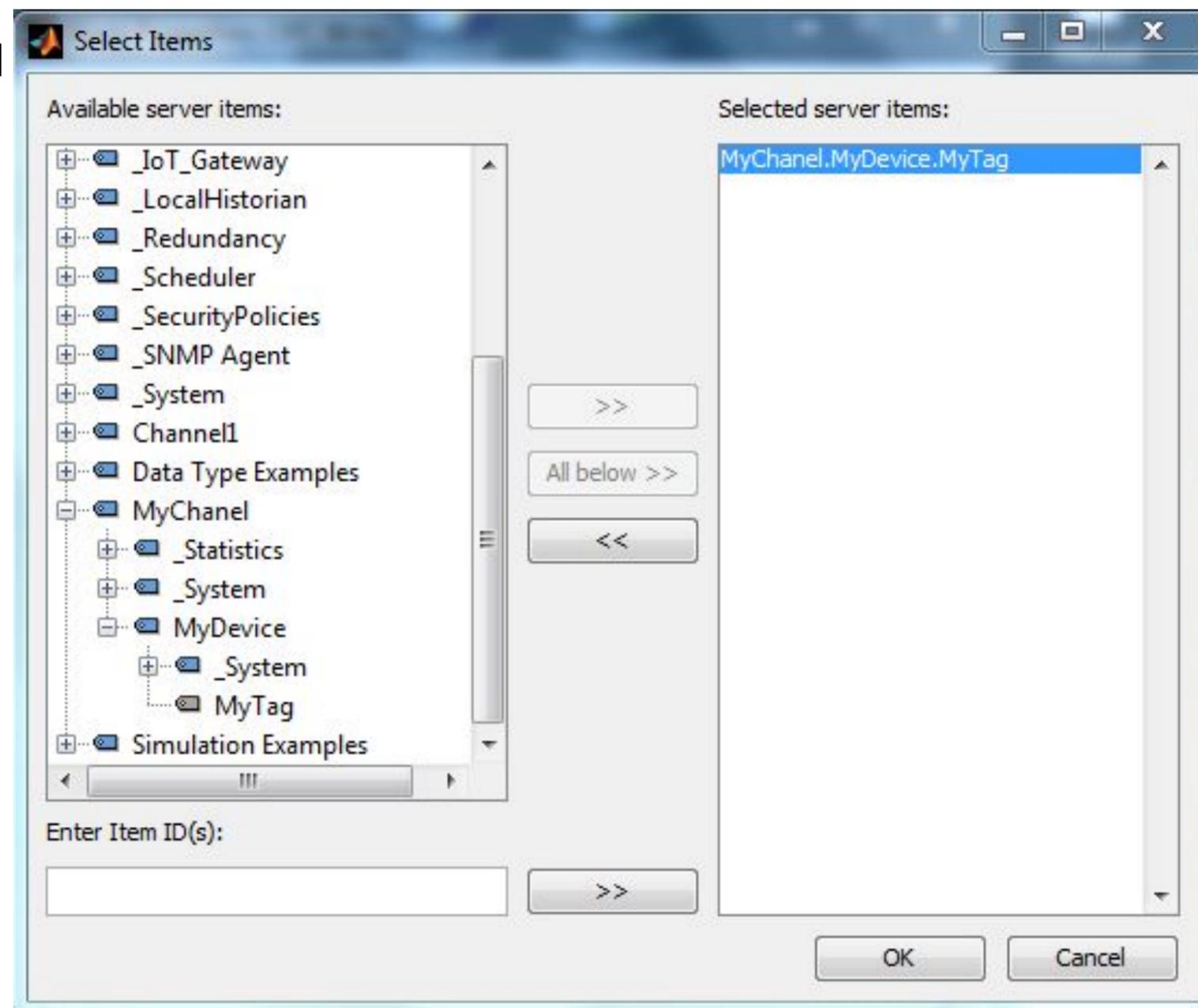
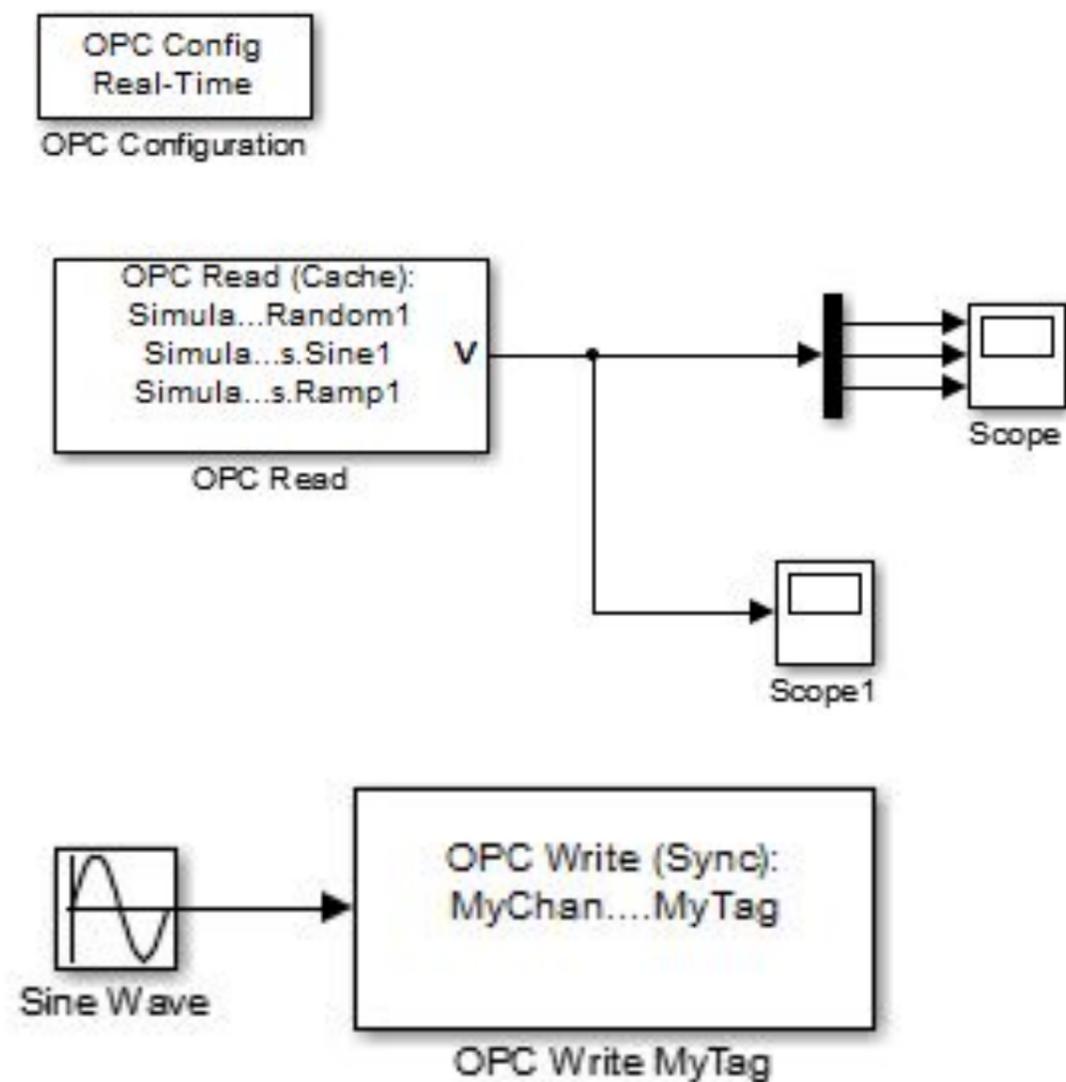
Создание тега для записи



55 OPC Клиент

Matlab + OPC

Модификация Simulink модели



56 OPC Клиент

Matlab + OPC

Запуск Simulink модели

Item ID	Data Type	Value	Timestamp	Quality
MyChanel.MyDevice.MyTag	Double	0.938	09:02:22.223	Good

