СЕВАСТОПОЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ

Лекция № 5

«ПРИМЕНЕНИЕ OPC ТЕХНОЛОГИЙ»

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

2 Что такое OPC? Где используются OPC?

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.

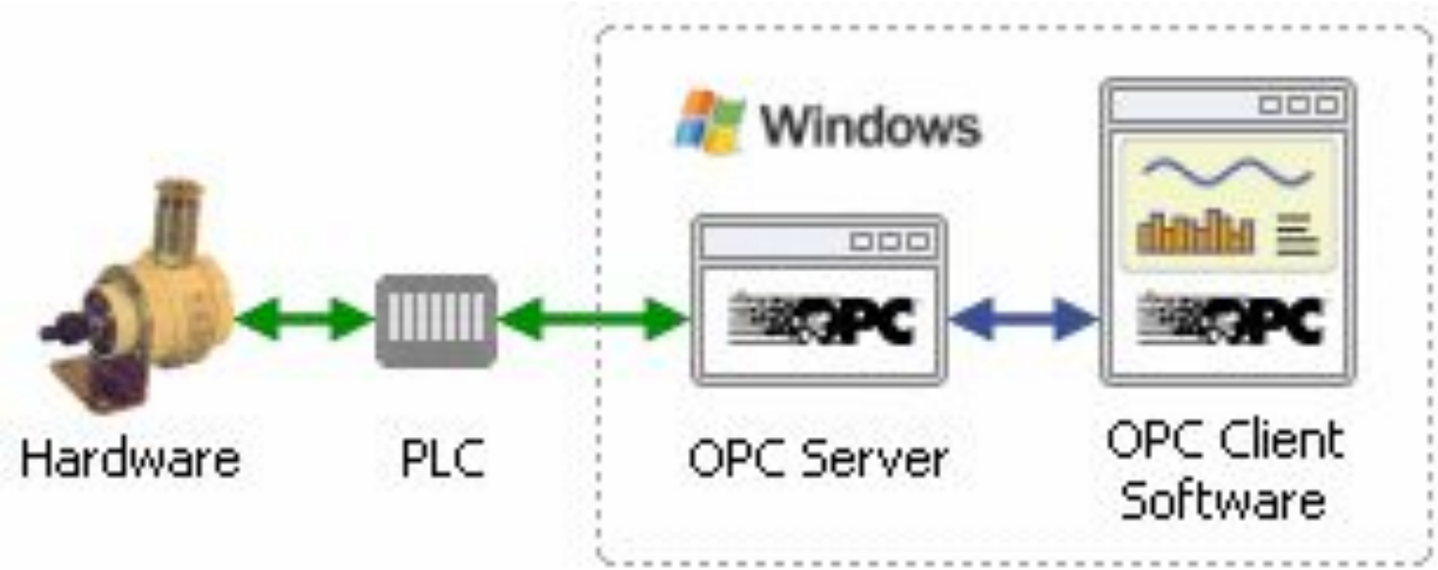
Основная цель — предоставить разработчикам промышленных программ универсальный фиксированный интерфейс (то есть набор функций) обмена данными с любыми устройствами. В то же время разработчики устройств предоставляют программу, реализующую этот интерфейс (набор функций).

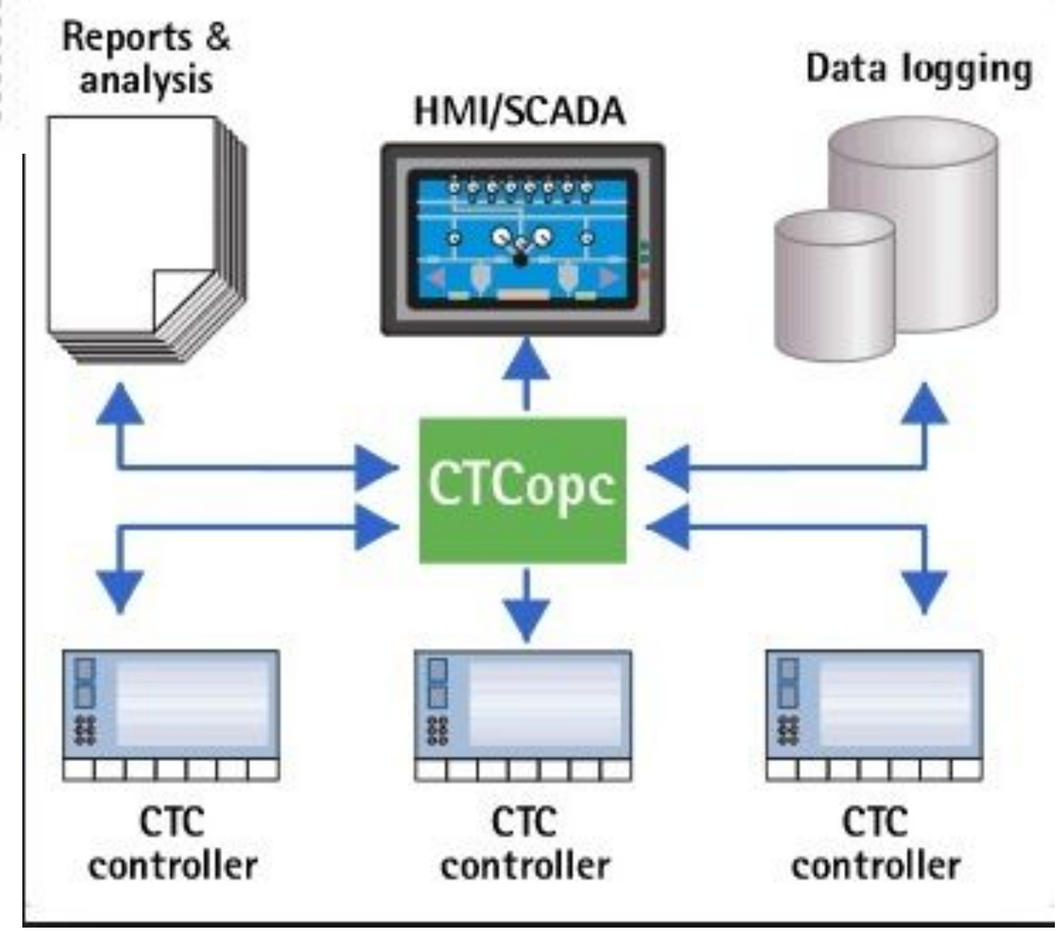
3 Что такое OPC? Где используются OPC?

Структура системы на базе OPC

4 Что такое OPC? Где используются OPC?

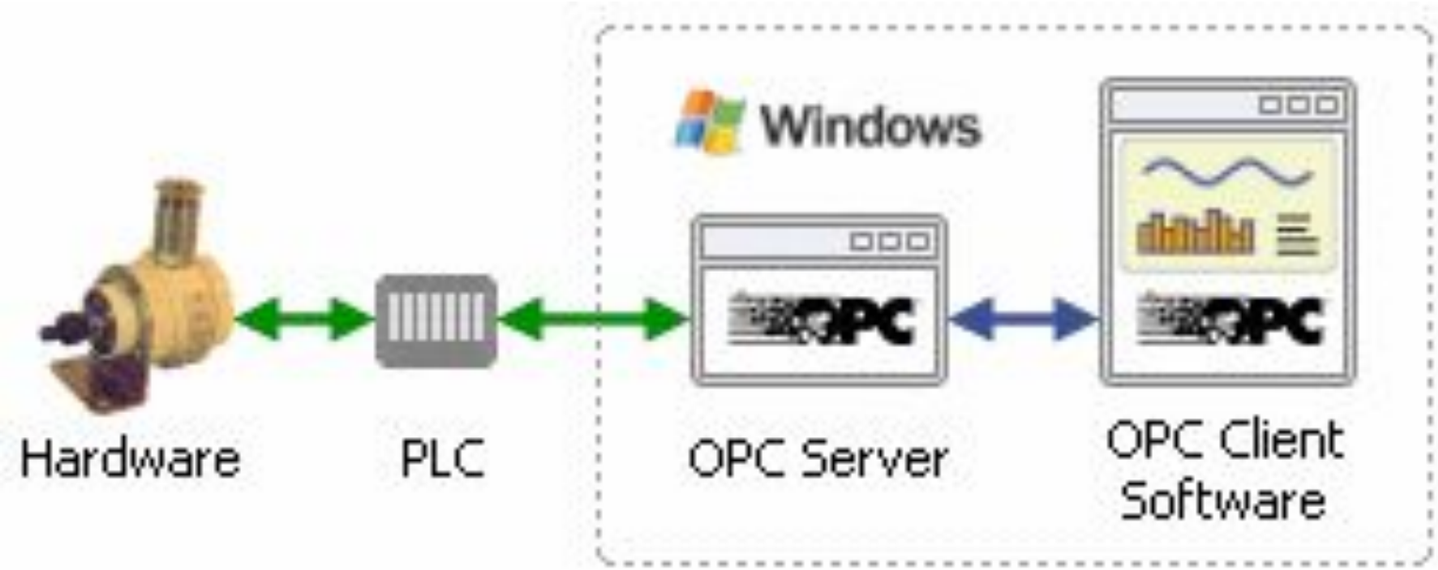
Структура системы на базе OPC

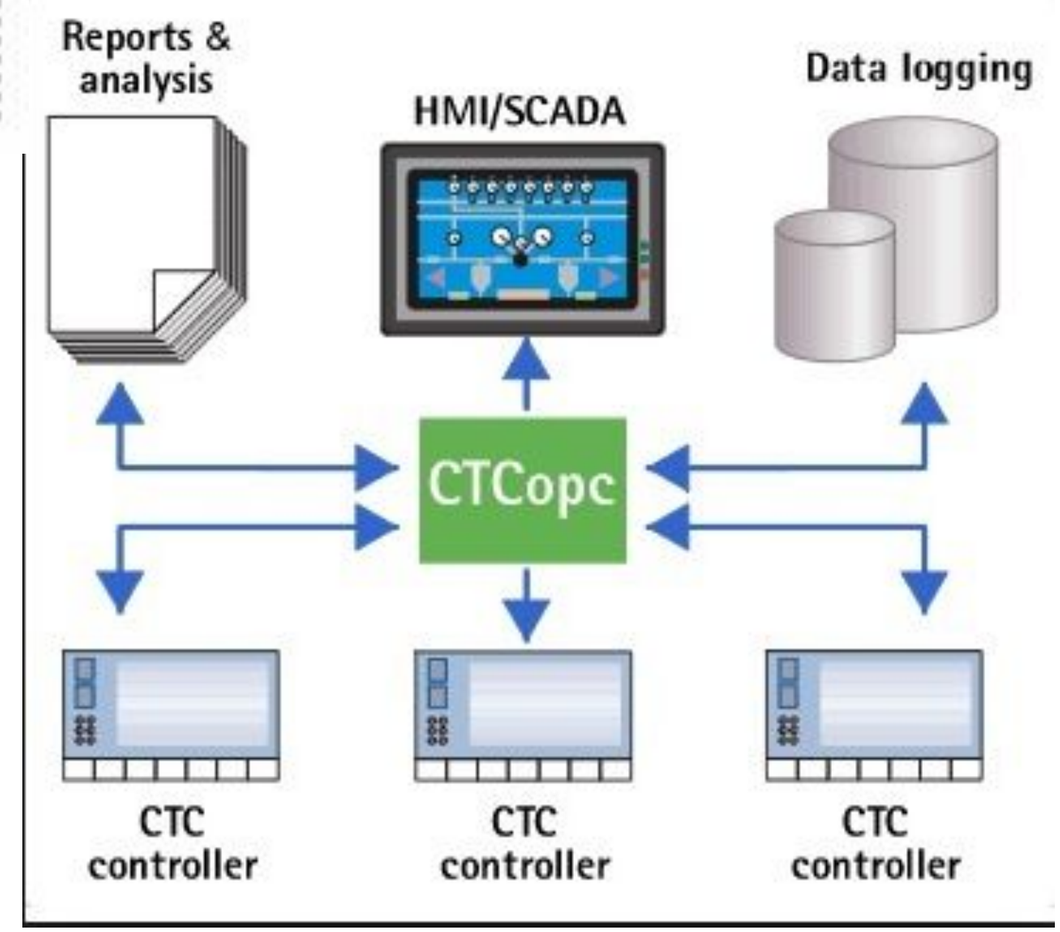




5 Что такое OPC? Где используются OPC?

Структура системы на базе OPC





6 Что такое OPC? Где используются OPC?

Распределенные объекты управления

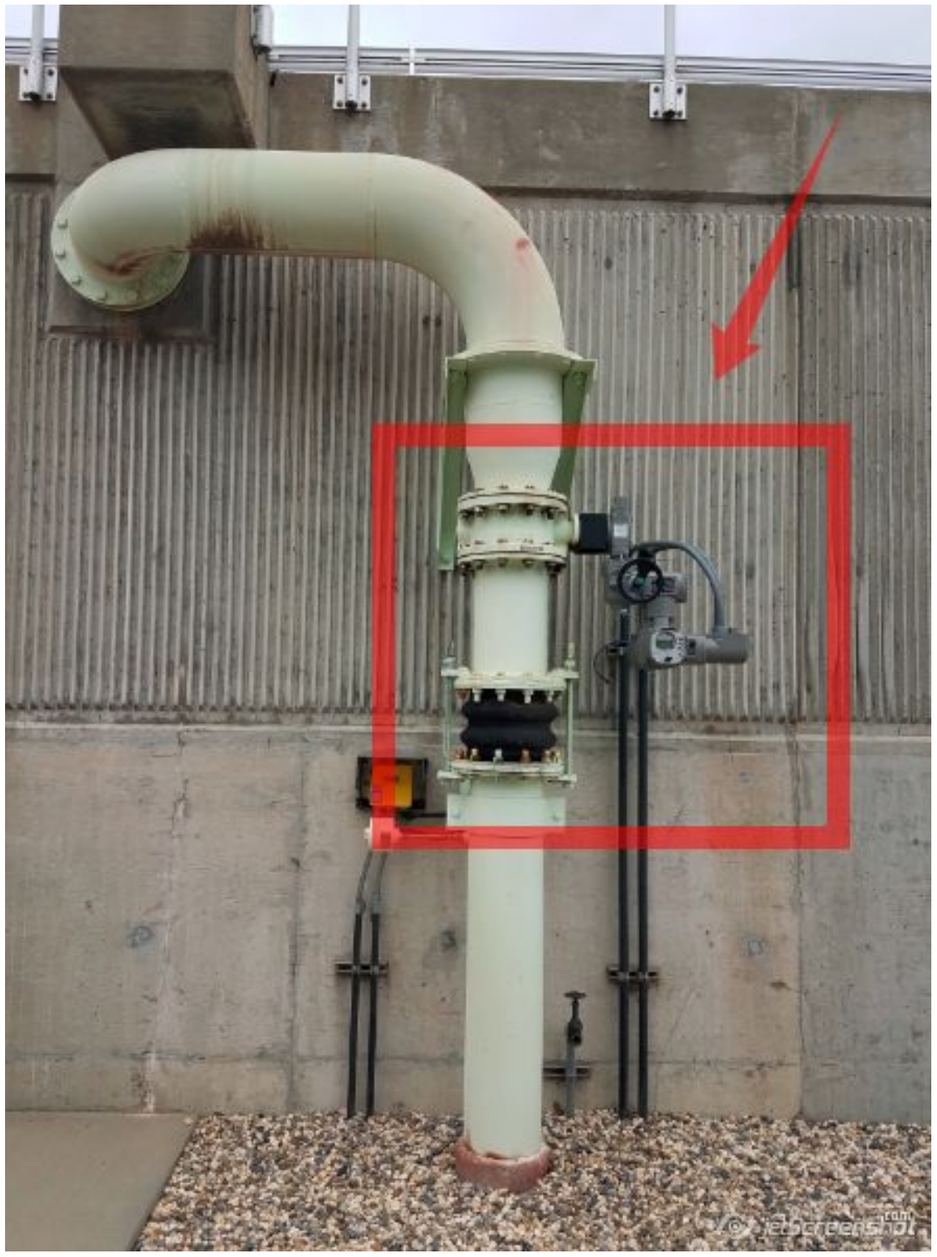
7 Что такое OPC? Где используются OPC?

Типовая структруа OPC (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 OPC? Где используются OPC?

Пример использования



9 Что такое OPC? Где используются OPC?

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.

12 ОРС Клиент/Сервер

Преимущество использования для производителя оборудования

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

До создания ОРС-технологии производителю промышленного оборудования приходилось создавать и поддерживать множество драйверов для наиболее распространенных систем автоматизации (или договариваться с производителями этих систем). Применение ОРС-технологии позволяет отказаться от создания драйверов и заменяет их одним универсальным ОРС-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-ввода 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.

16 OPC Сервер

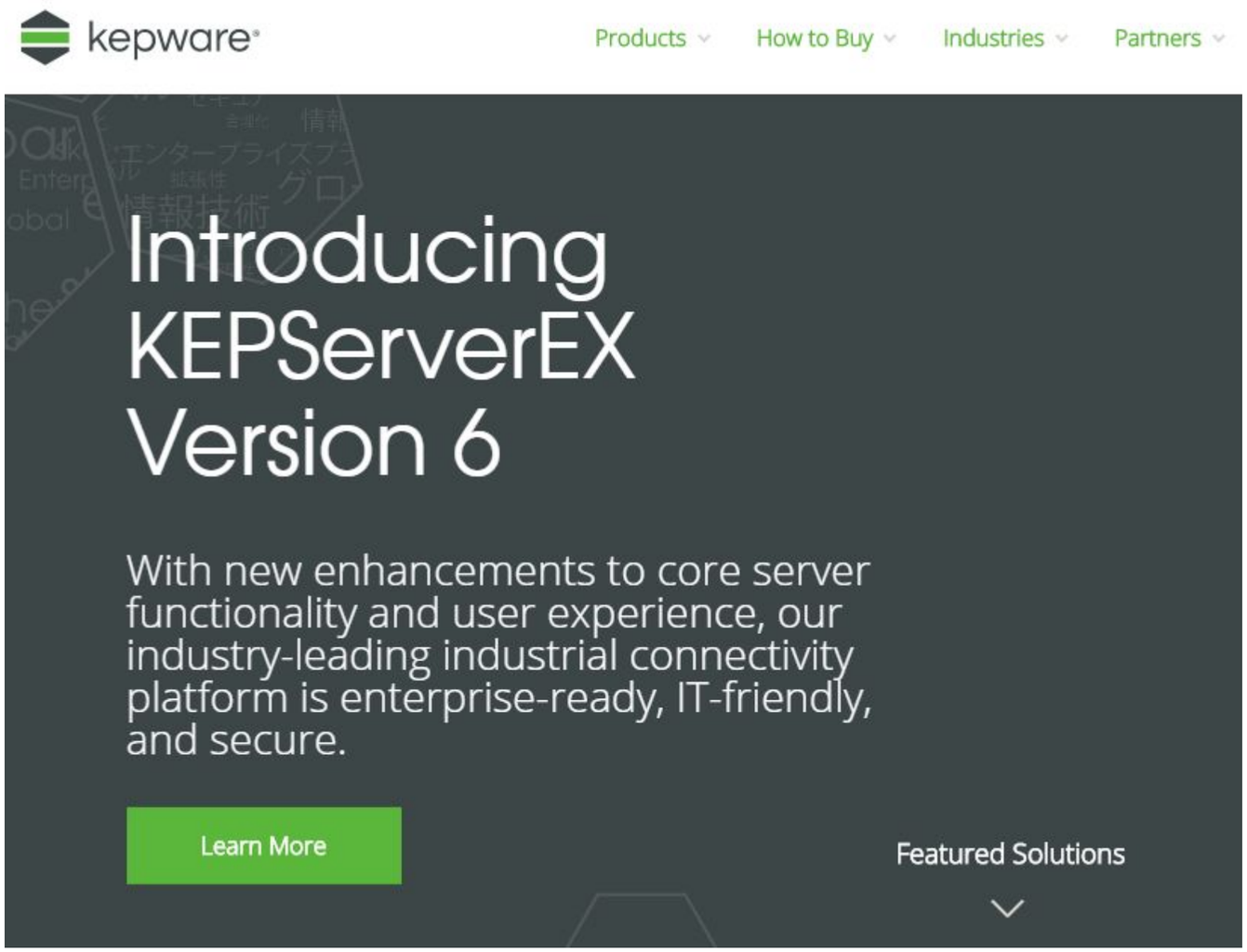
Режимы чтения данных

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

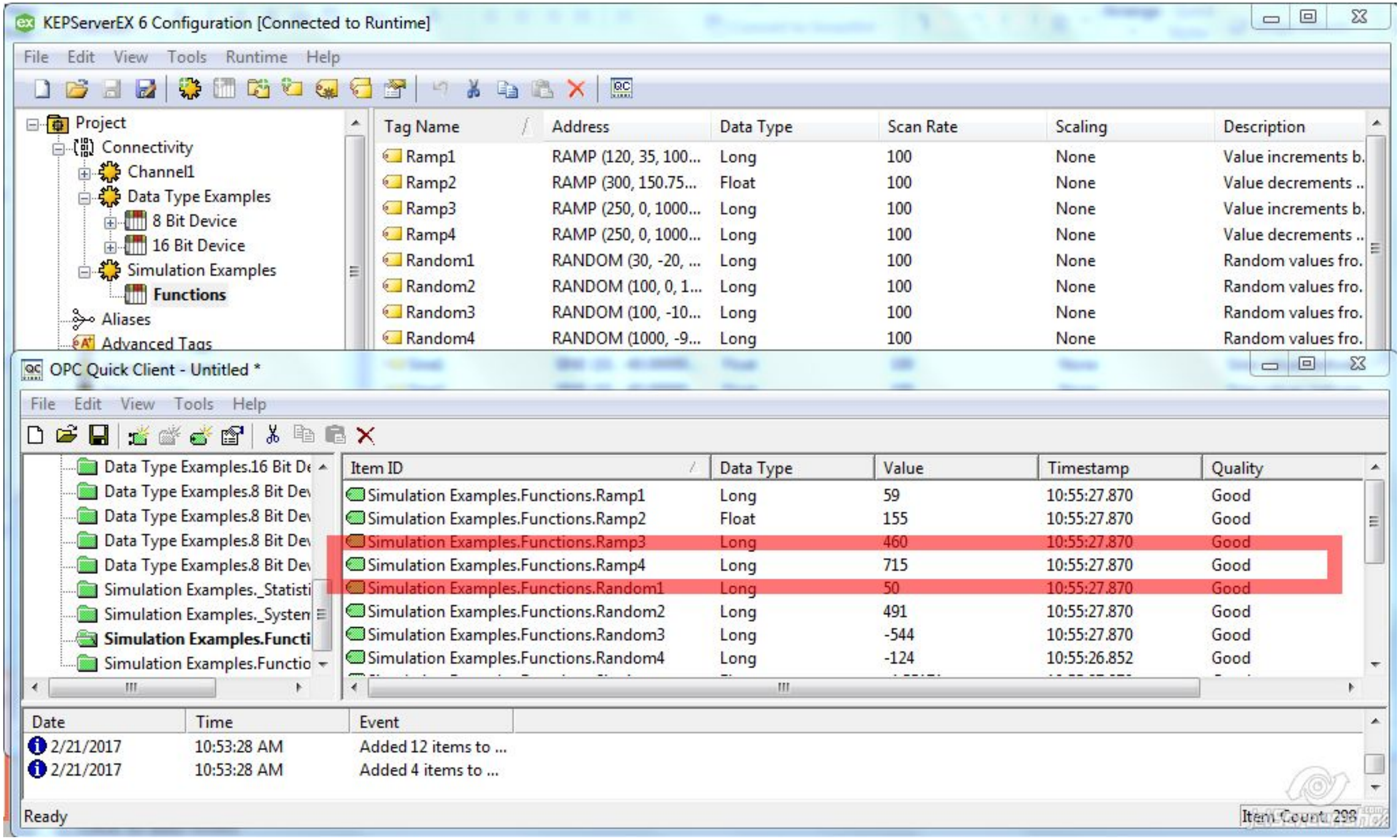
Kepware Server



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

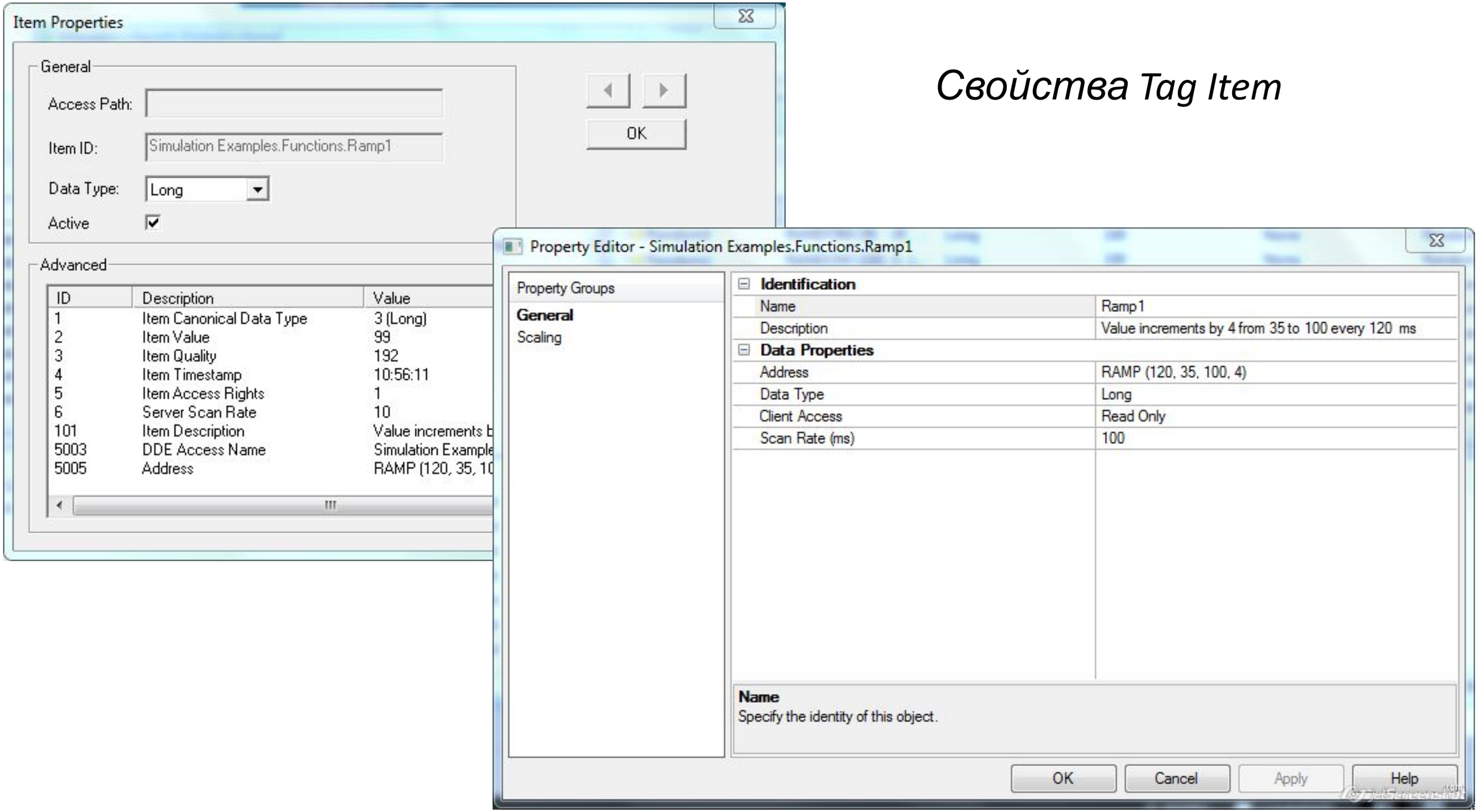
18 OPC Сервер

Kepware Server



19 OPC Сервер

Kepware Server



Свойства Tag Item

OPC Клиент

Kepware ClientACE

21 ОРС 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 соединения

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 ОРС Клиент

Kepware ClientACE + Visual Studio C# WinForm application

Добавить метод Connect

FormMain.cs* | FormMain.cs [Design]*

ClientACEWinForm.FormMain | FormMain()

```
5   using System.Drawing;
6   using System.Linq;
7   using System.Text;
8   using System.Threading.Tasks;
9   using System.Windows.Forms;
10  using System.Windows.Forms;
11  using Kepware.ClientAce.OpcDaClient;
12
13  namespace ClientACEWinForm
14  {
15      public partial class FormMain : Form
16      {
17          public FormMain()
18          {
19              InitializeComponent();
20          }
21
```

Solution Explorer

Search Solution Explorer (Ctrl+;)

- References
  - Kepware.ClientAce.OpcClient
  - Microsoft.CSharp
  - System
  - System.Core
  - System.Data
  - System.Data.DataSetExtensions
  - System.Deployment
  - System.Drawing
  - System.Windows.Forms
  - System.Xml

Properties

FormMain.cs File Properties

23 ОРС 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 ОРС Клиент

Kepware ClientACE + Visual Studio C# WinForm application

Подписка на события

```
        private void SubscribeToOPCDAServerEvents()
        {
            DAServer.DataChanged += new
Kepware.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);
        }
```

25 ОРС Клиент

Kepware ClientACE + Visual Studio C# WinForm application

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 ОРС Клиент

Kepware ClientACE + Visual Studio C# WinForm application

Обновление состояния подписки

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

27 ОРС Клиент

Kepware ClientACE + Visual Studio C# WinForm application

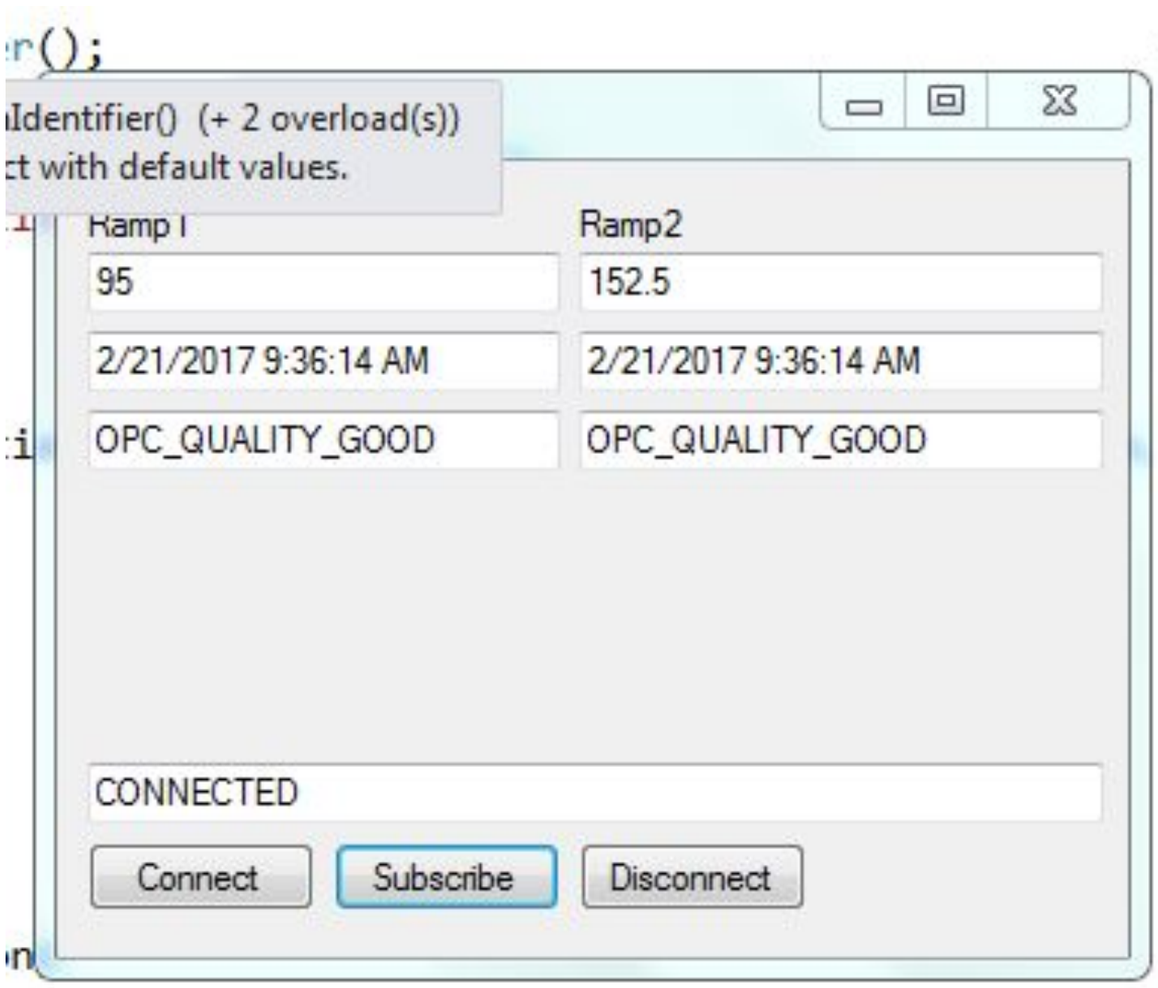
Обработка события OnDataChanges

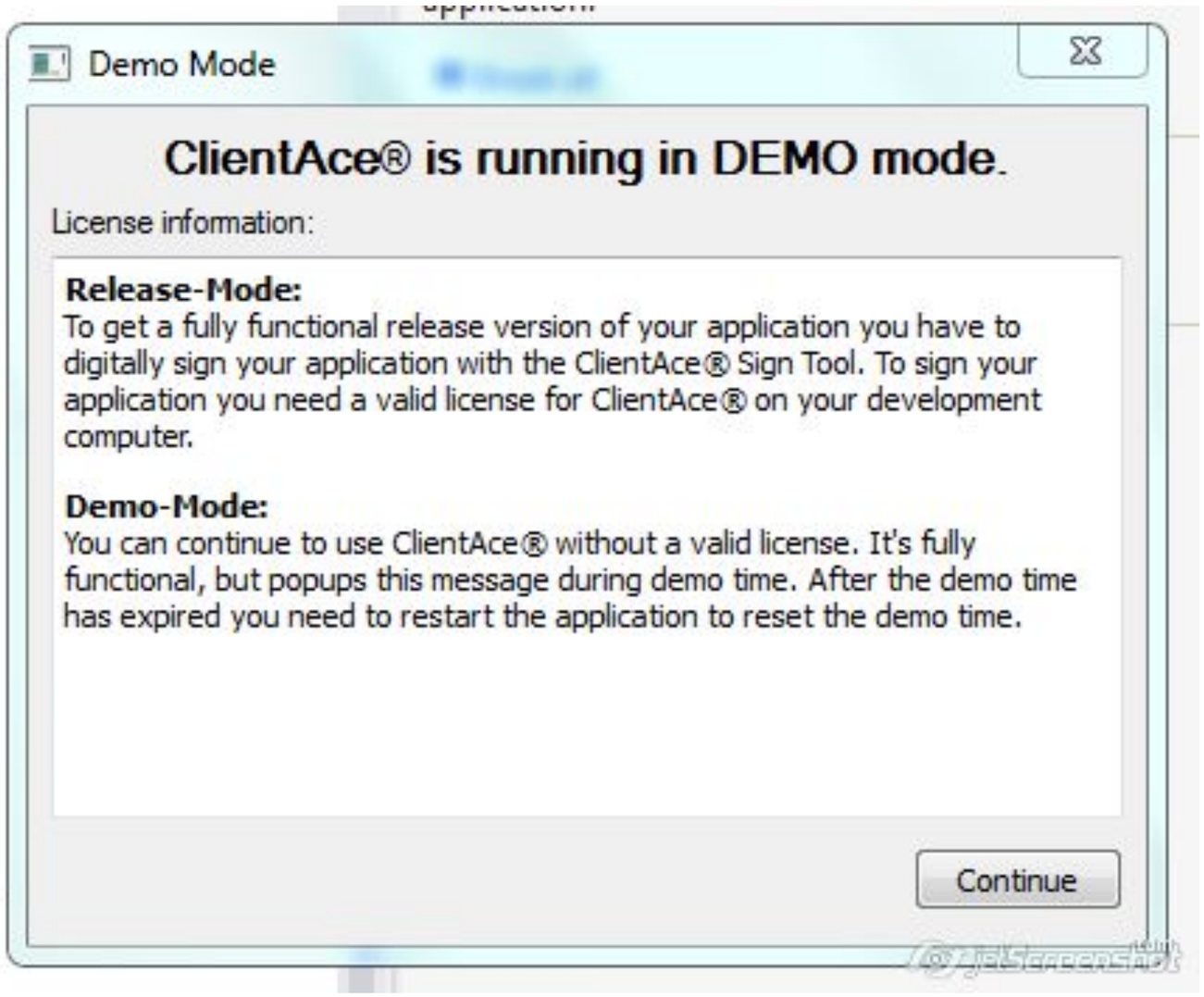
```
        public void DataChanged(int clientSubscription, bool allQualitiesGood, bool noErrors,
ItemValueCallback[] ItemValues)
        {
            foreach (ItemValueCallback item in ItemValues)
            {
                switch ((int)item.ClientHandle)
                {
                    case 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 ОРС Клиент

Kepware ClientACE + Visual Studio C# WinForm application

WinForm in action





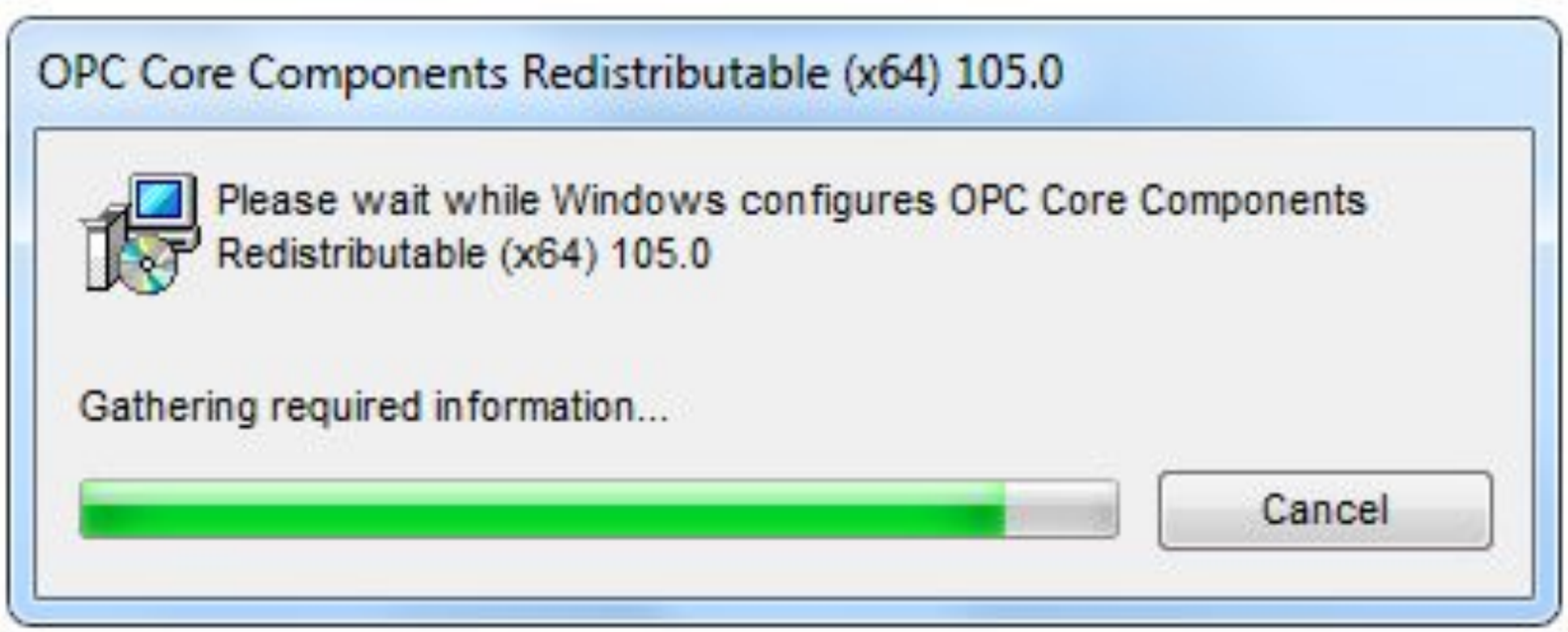
29 OPC Клиент

Matlab + OPC

Установка пакета

Command Window

```
>> opcregister('-silent')
```



30 OPC Клиент

Matlab + OPC

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'}
    OPCSpecification: {'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
```

32 OPC Клиент

Matlab + OPC

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)
```

ОРС Клиент

Matlab + OPC

Настройка параметров мониторинга

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

OPC Клиент

Matlab + OPC

Запуск мониторинга

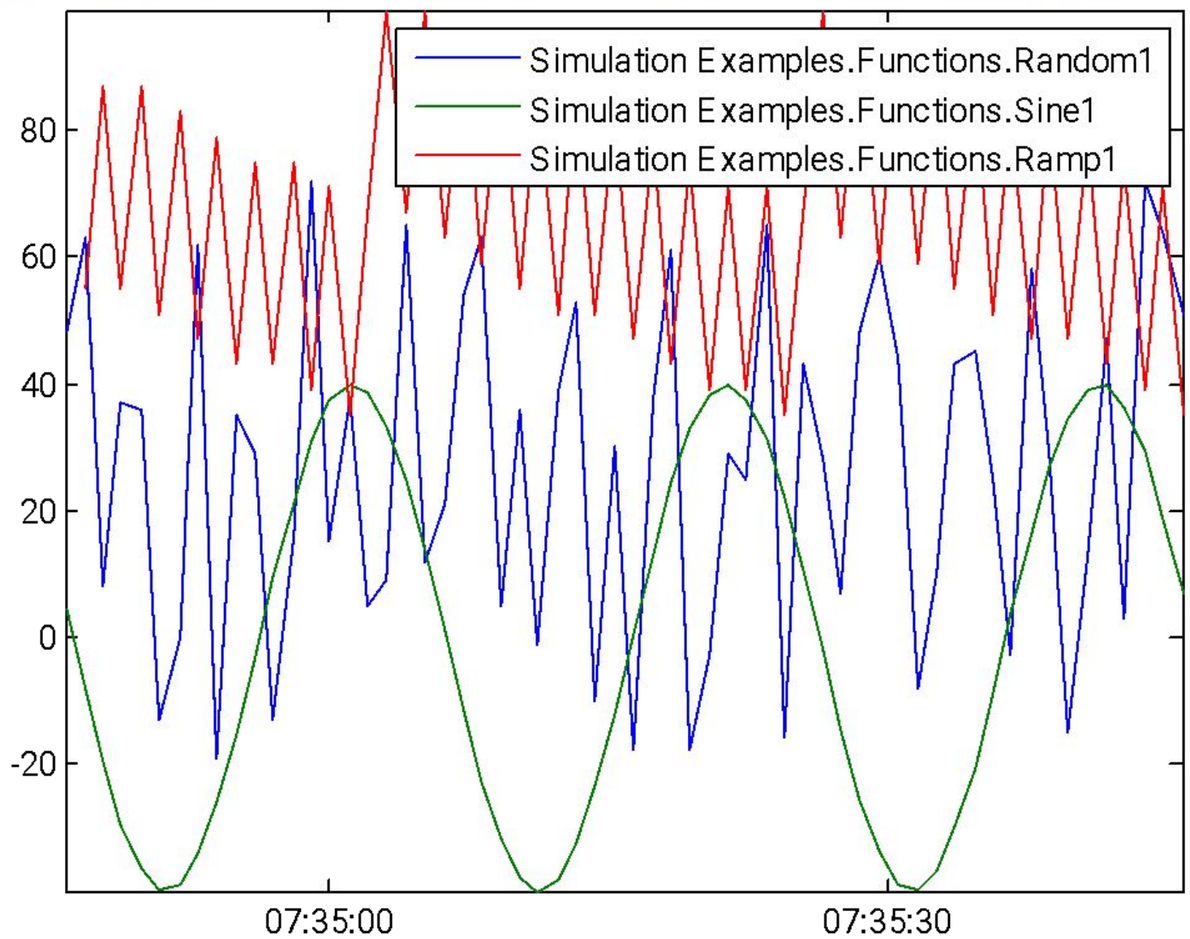
```
start(grp)
wait(grp)
[logIDs,logVal,logQual,logTime,logEvtTime] = getdata(grp,'double');
```

ОРС Клиент

Matlab + OPC

Результаты мониторинга

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



ОРС Клиент

Matlab + OPC

Закрытие соединения

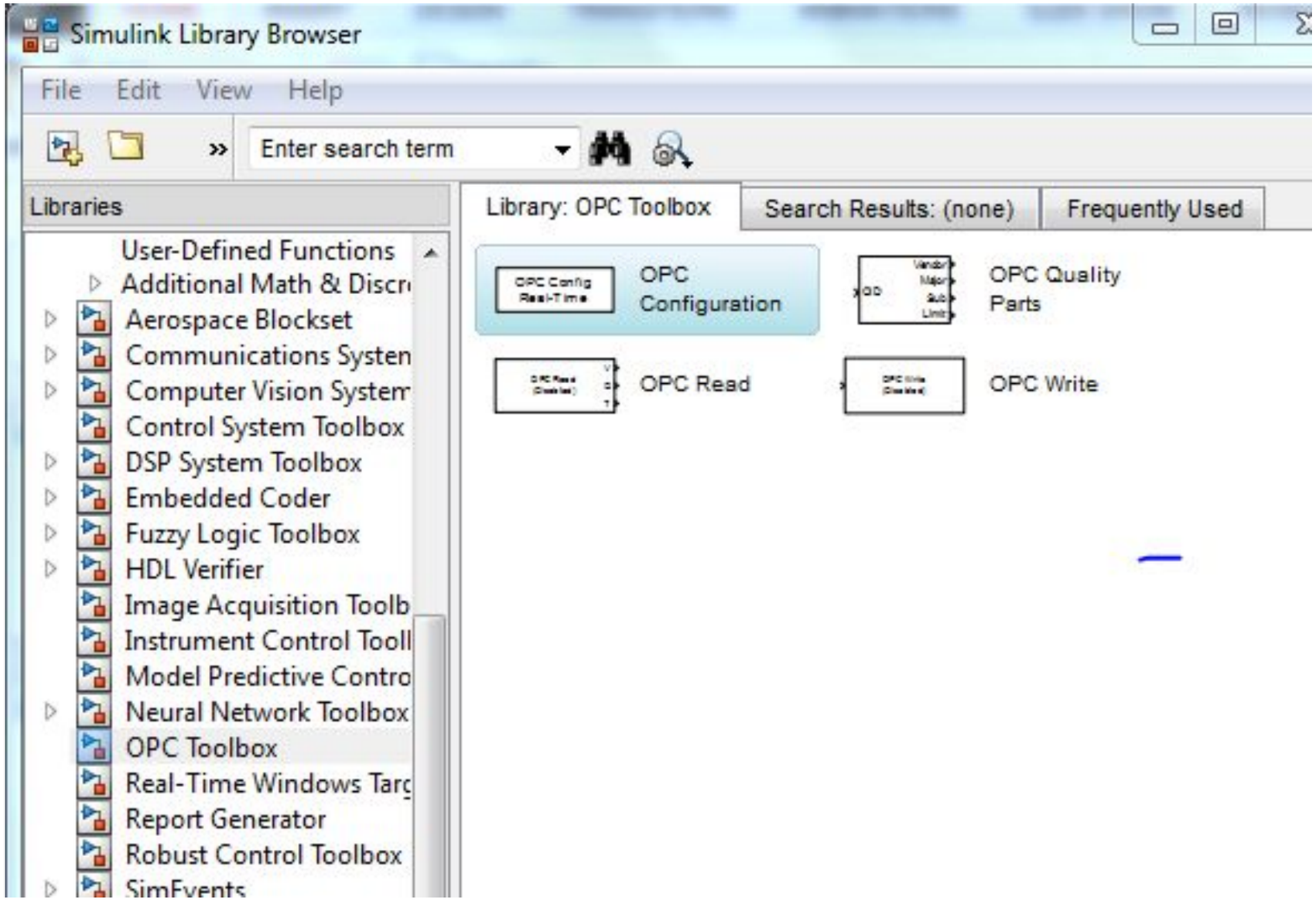
```
disconnect(da)
delete(da)
```

Недостаток метода – offline режим сбора данных

37 OPC Клиент

Matlab + OPC

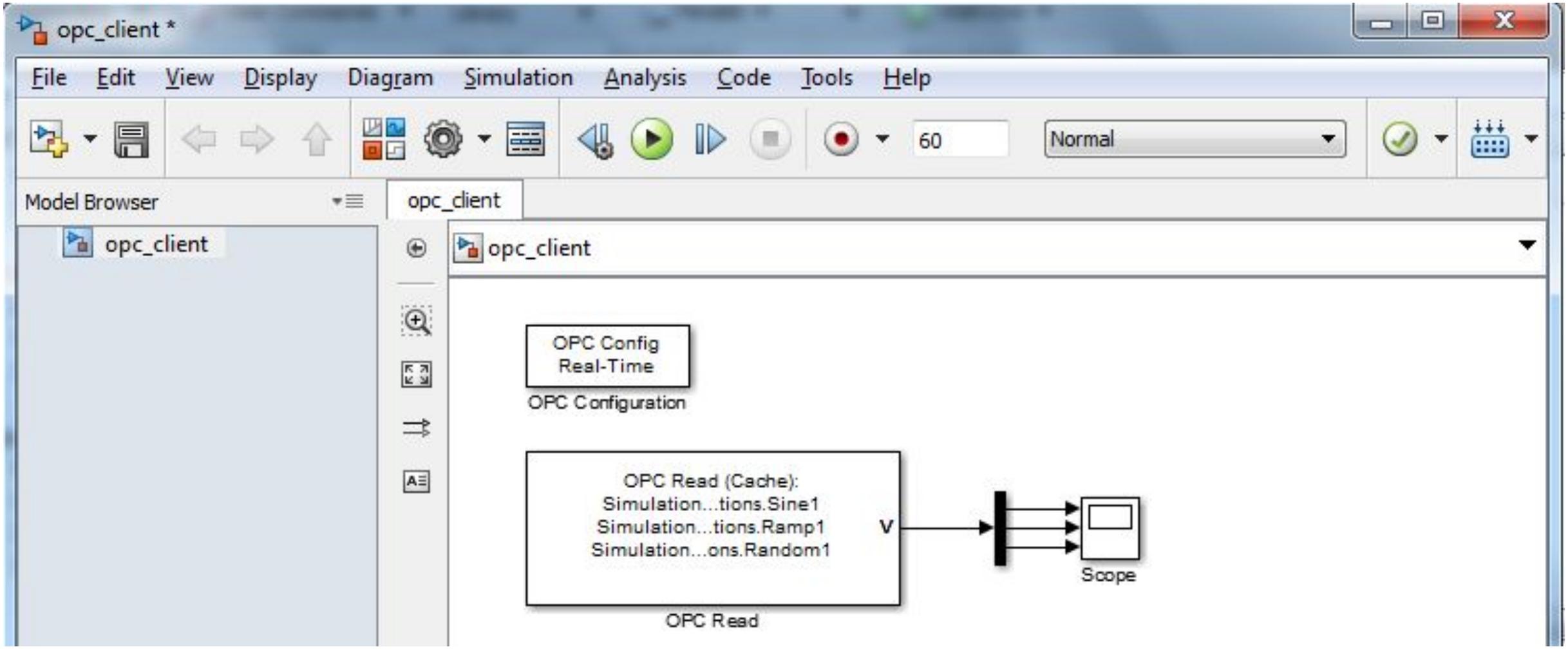
Применение Simulink модели в качестве OPC клиента



38 ОРС Клиент

Matlab + OPC

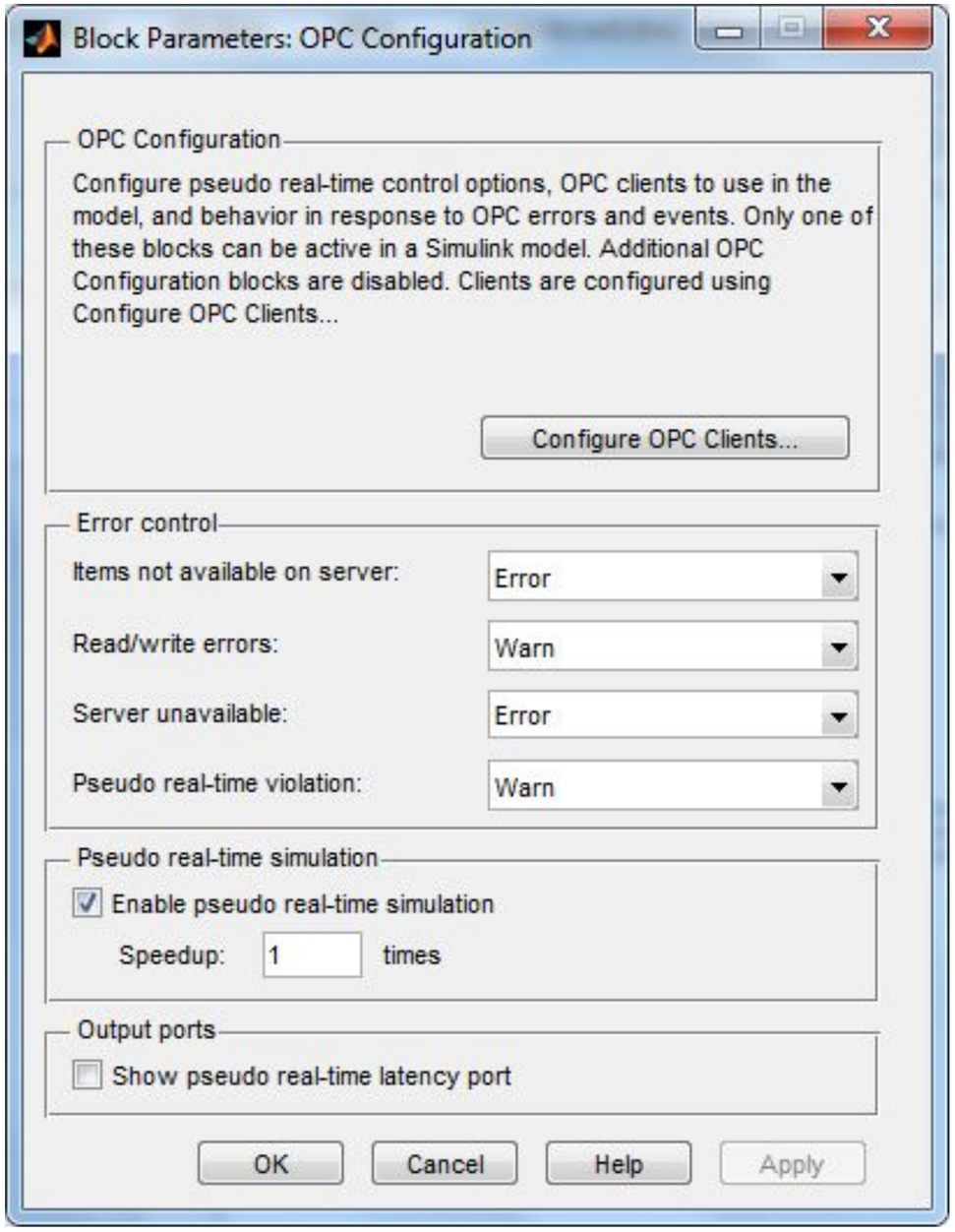
Создание модели

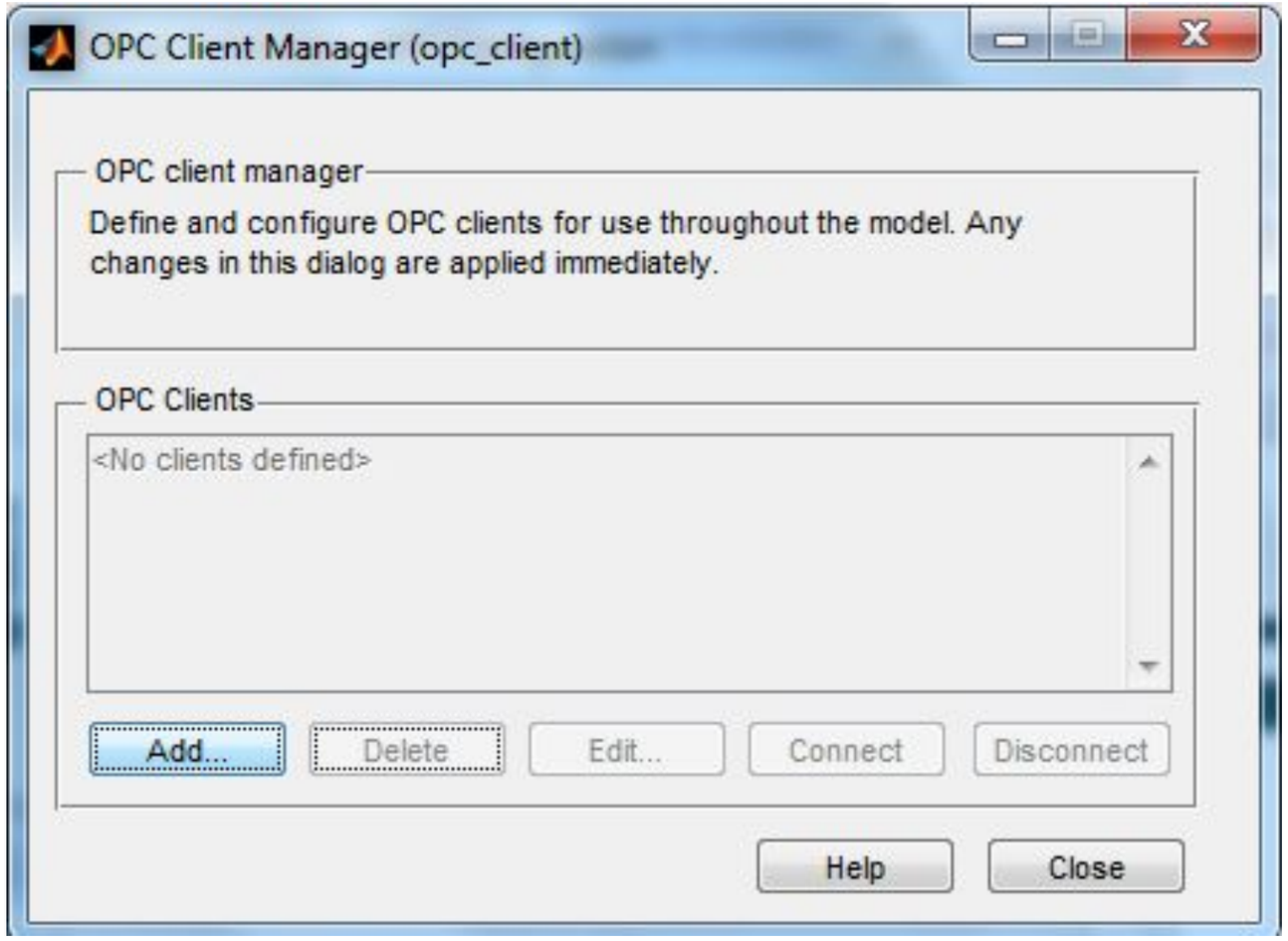


39 OPC Клиент

Matlab + OPC

Настройка конфигуратора

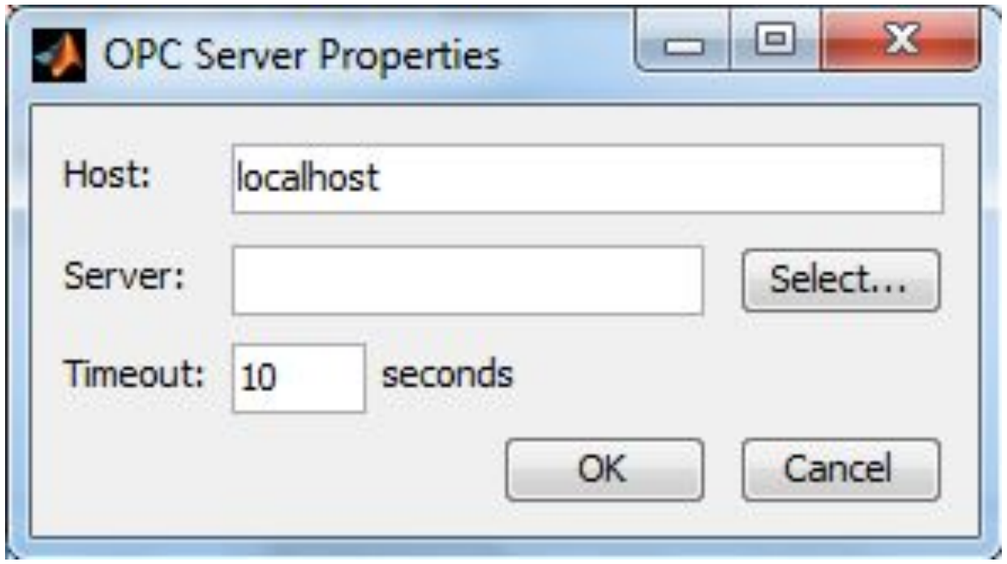


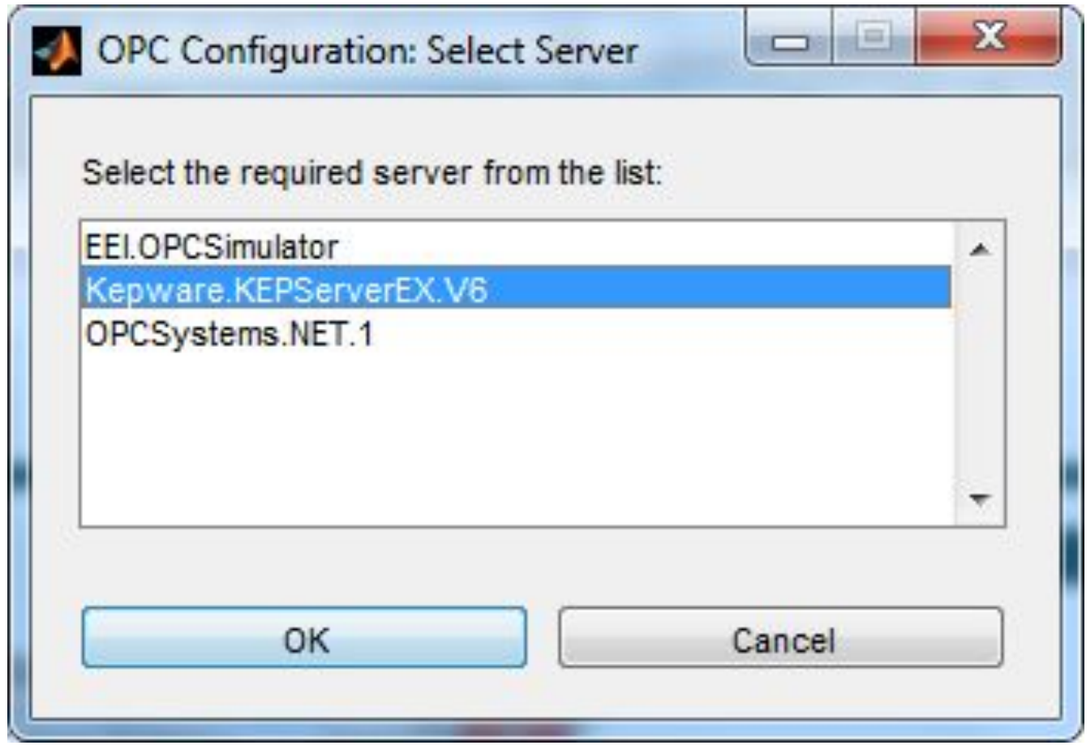


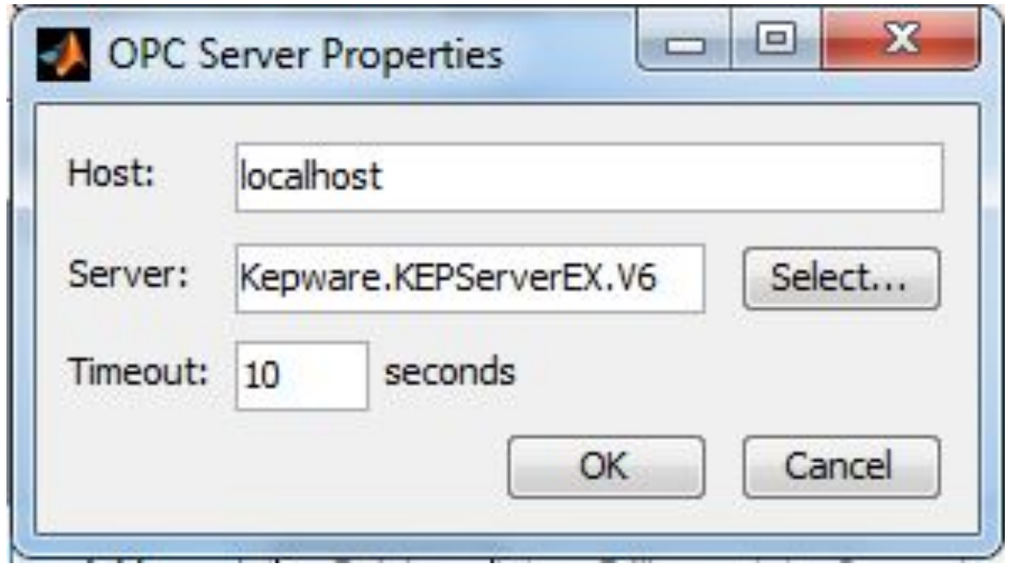
ОРС Клиент

Matlab + OPC

Настройка конфигуратора



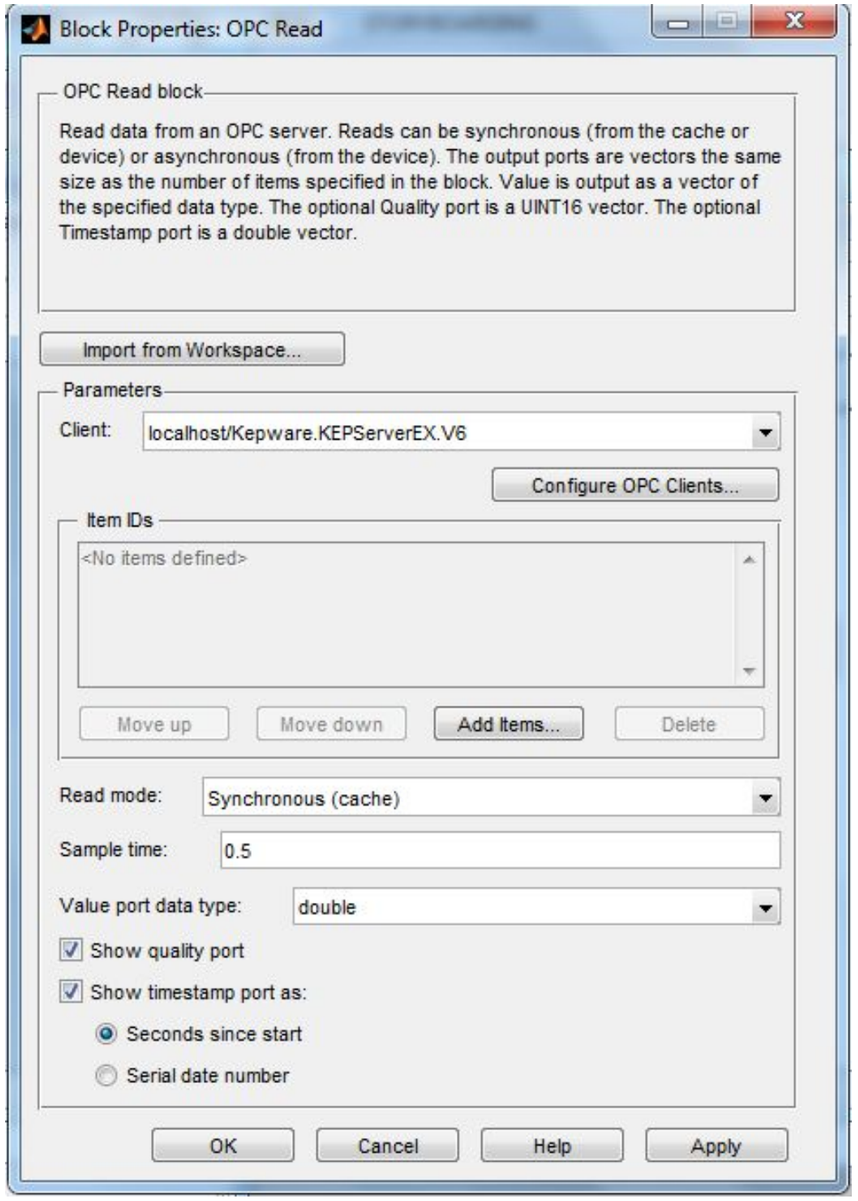


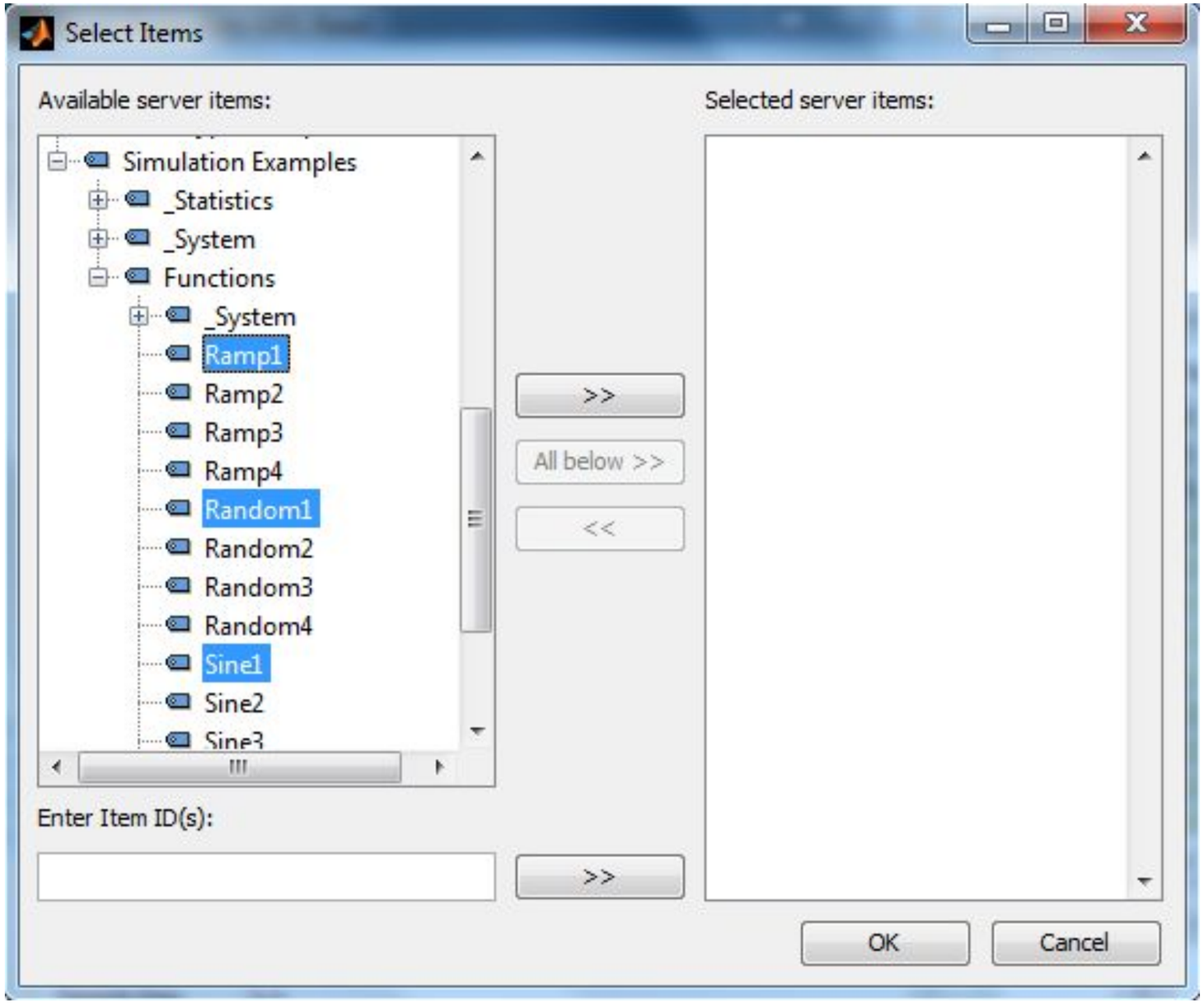


41 ОРС Клиент

Matlab + OPC

Подключение к тегам

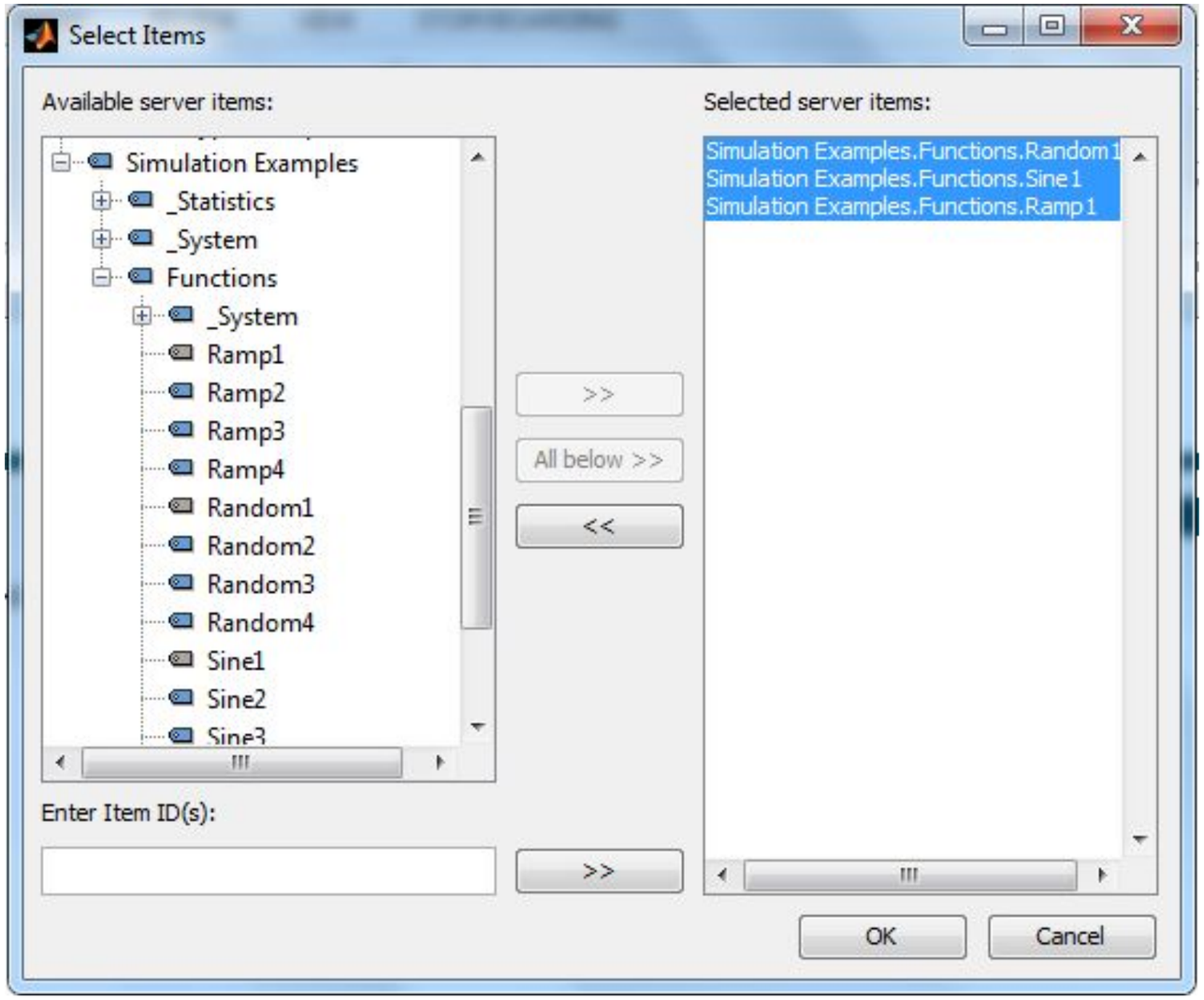


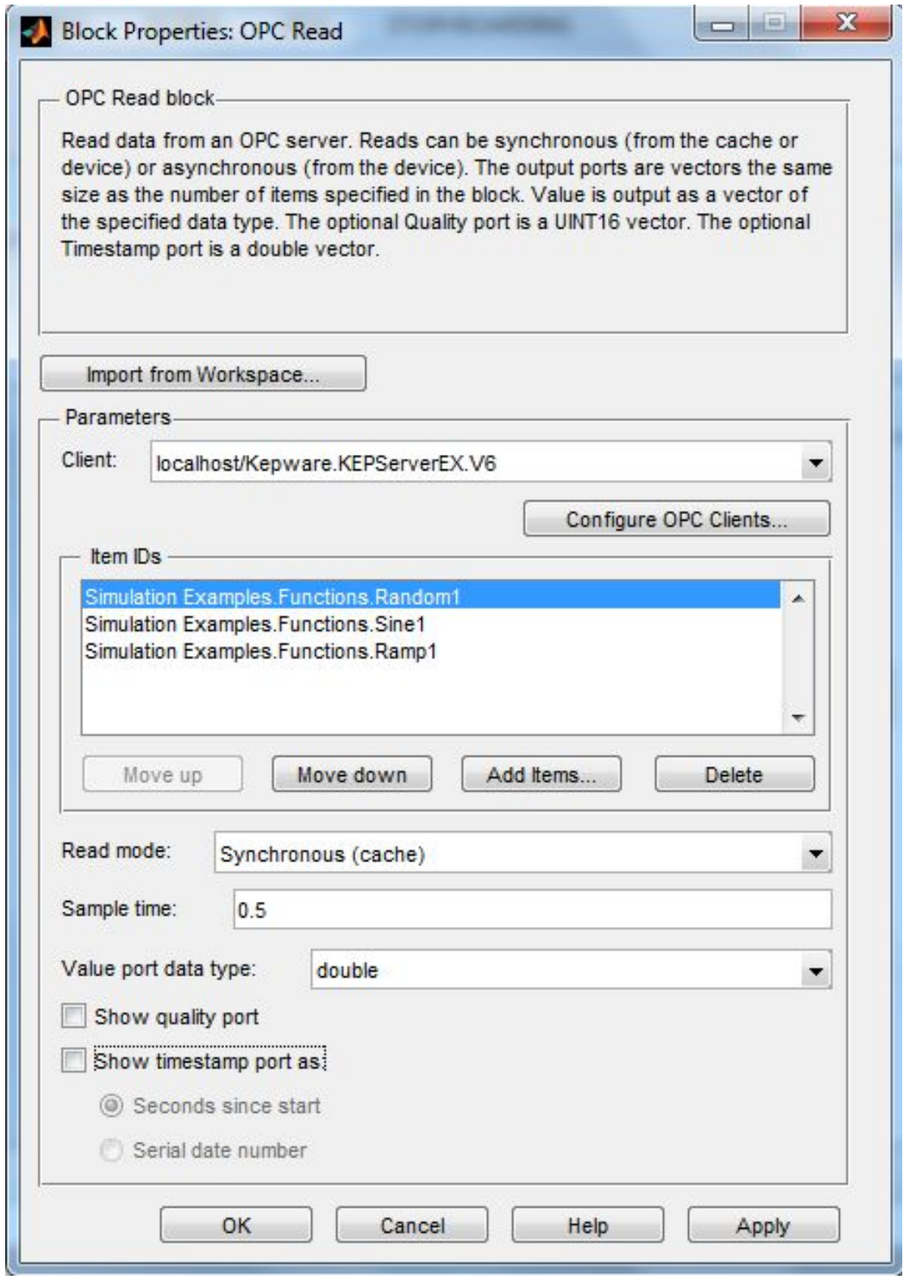


42 OPC Клиент

Matlab + OPC

Подключение к тегам

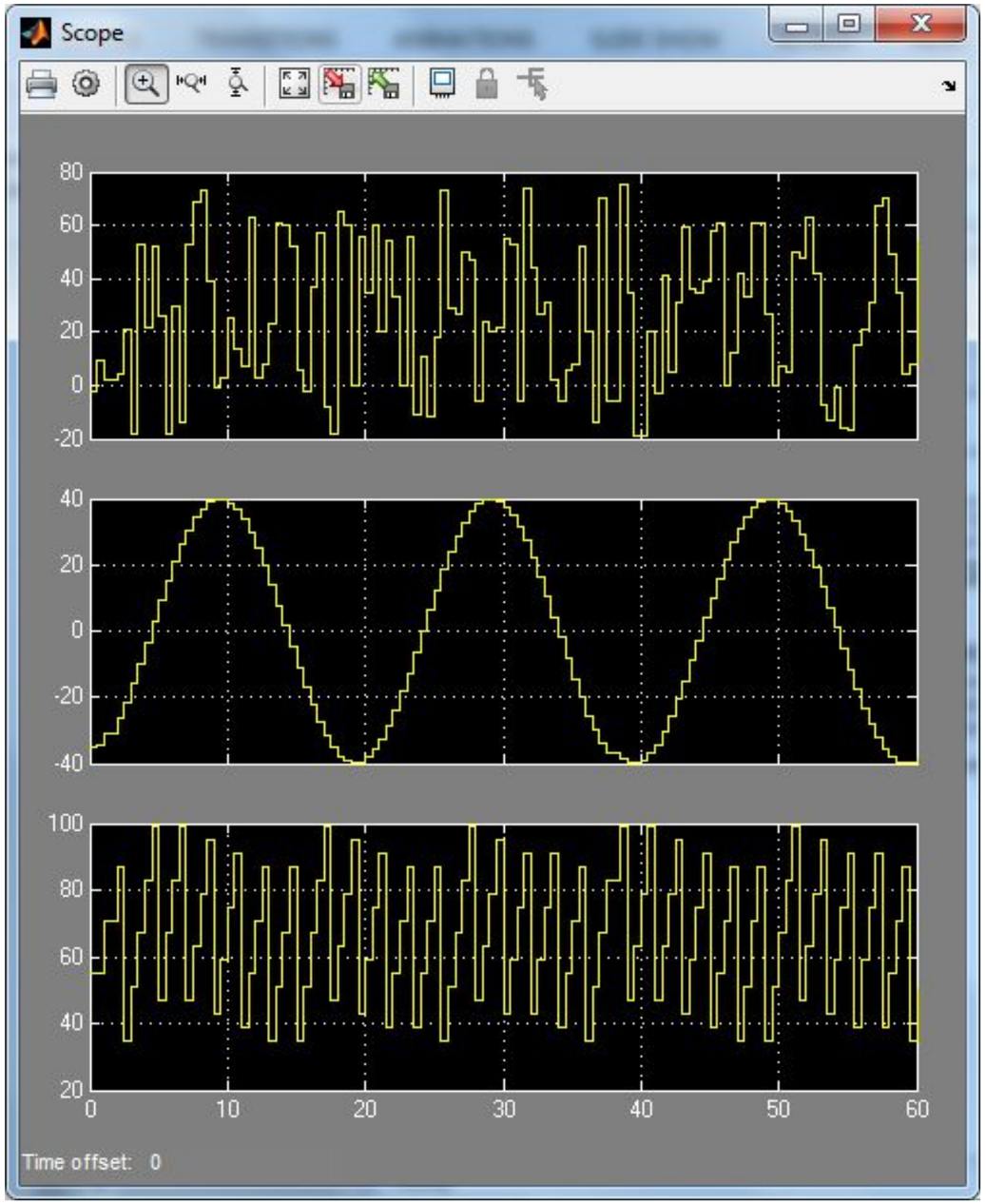


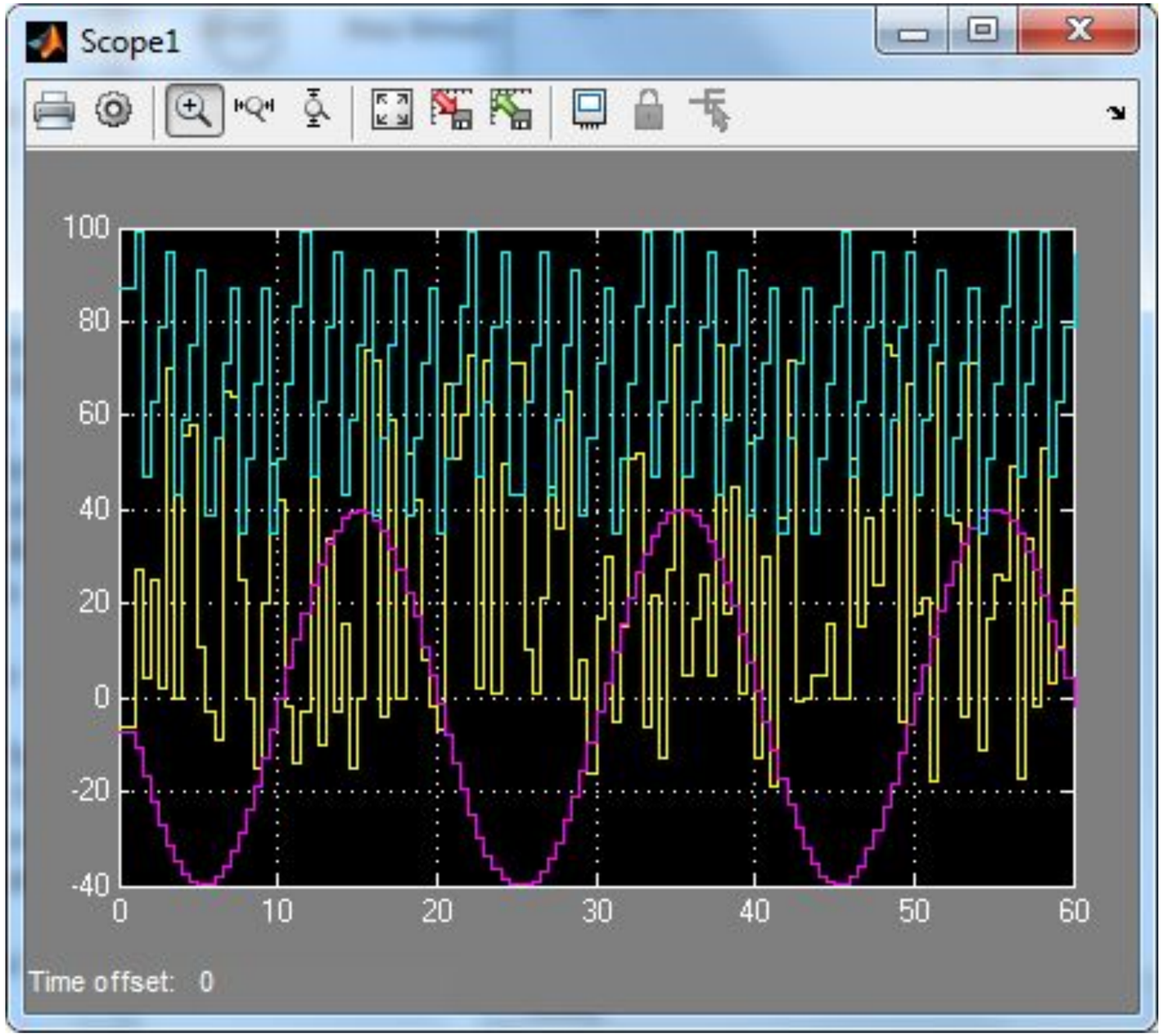


ОРС Клиент

Matlab + OPC

Запуск модели

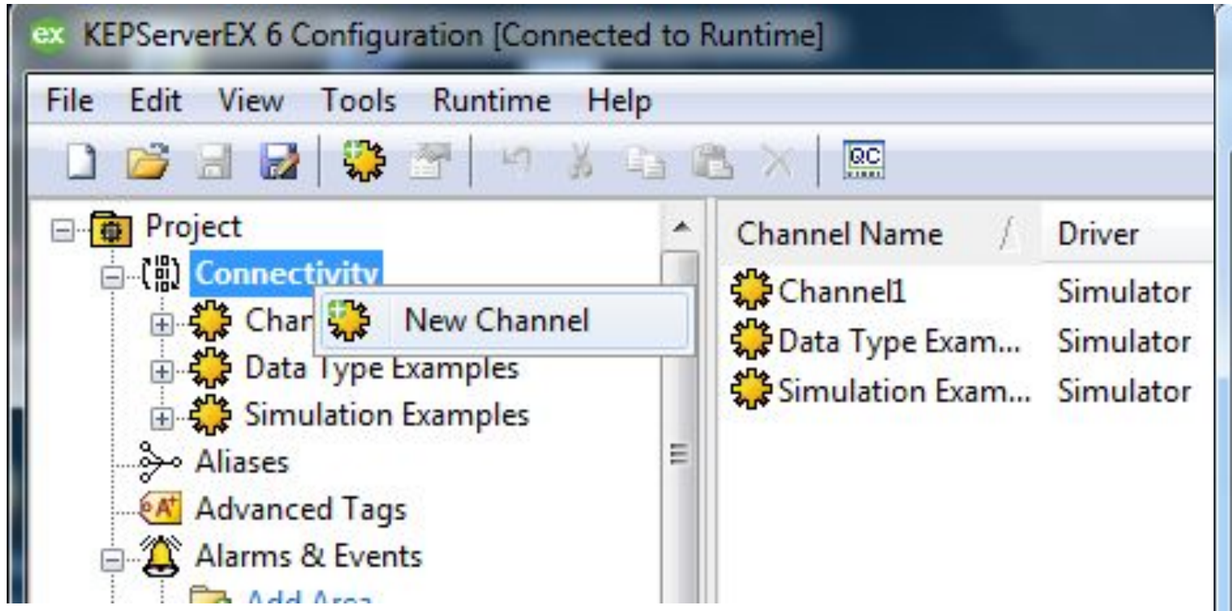


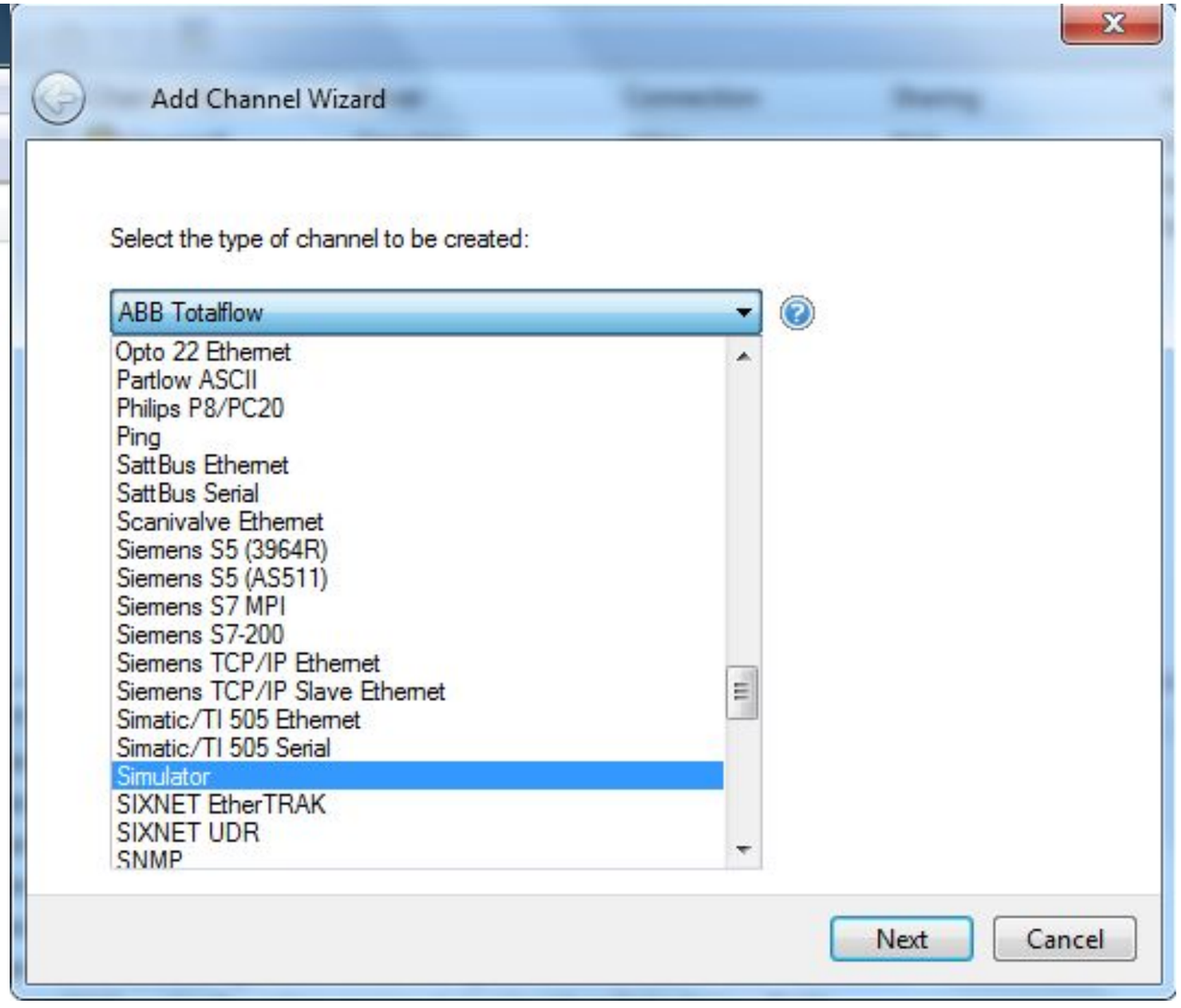


44 ОРС Клиент

Matlab + OPC

Создание тега для записи

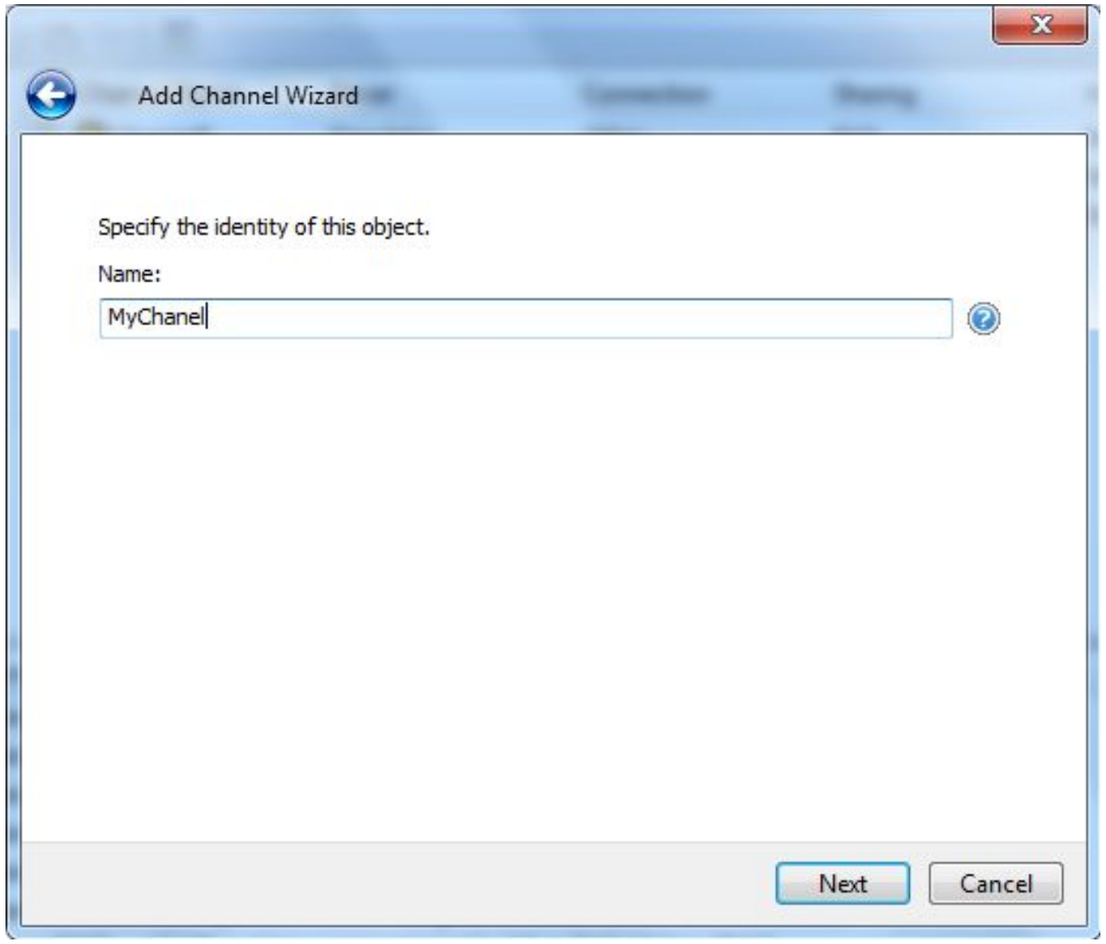


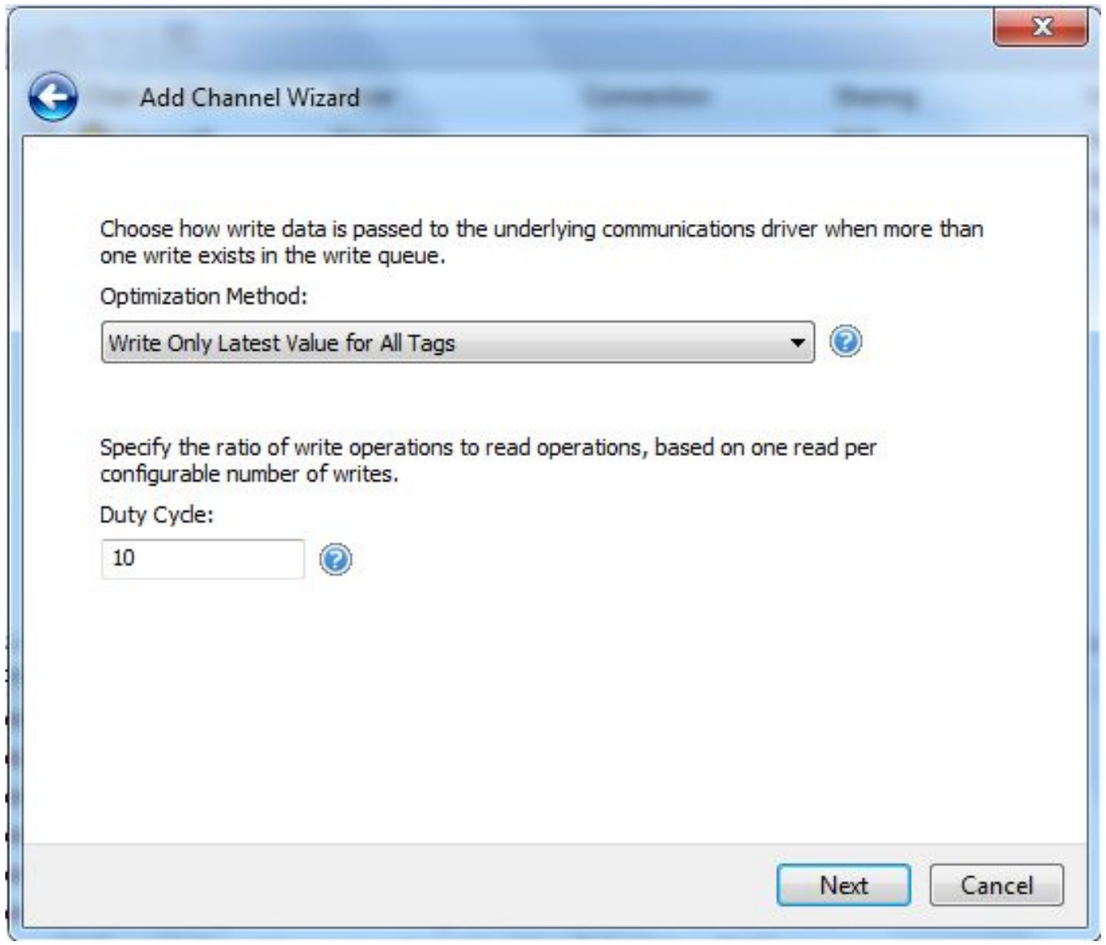


ОРС Клиент

Matlab + OPC

Создание тега для записи

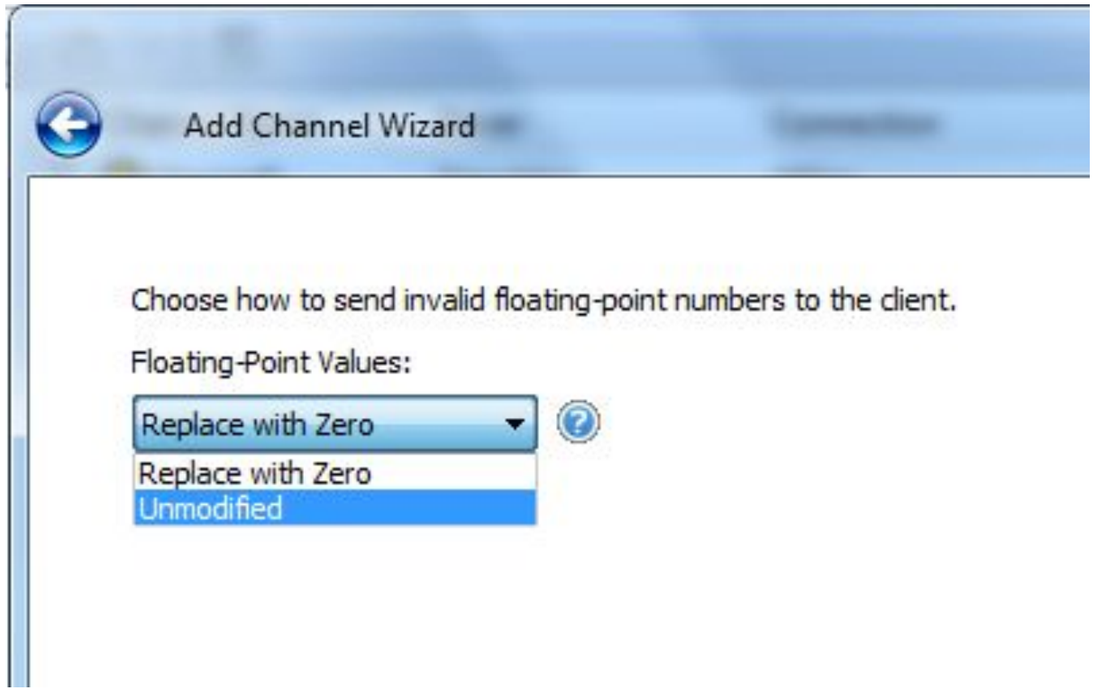


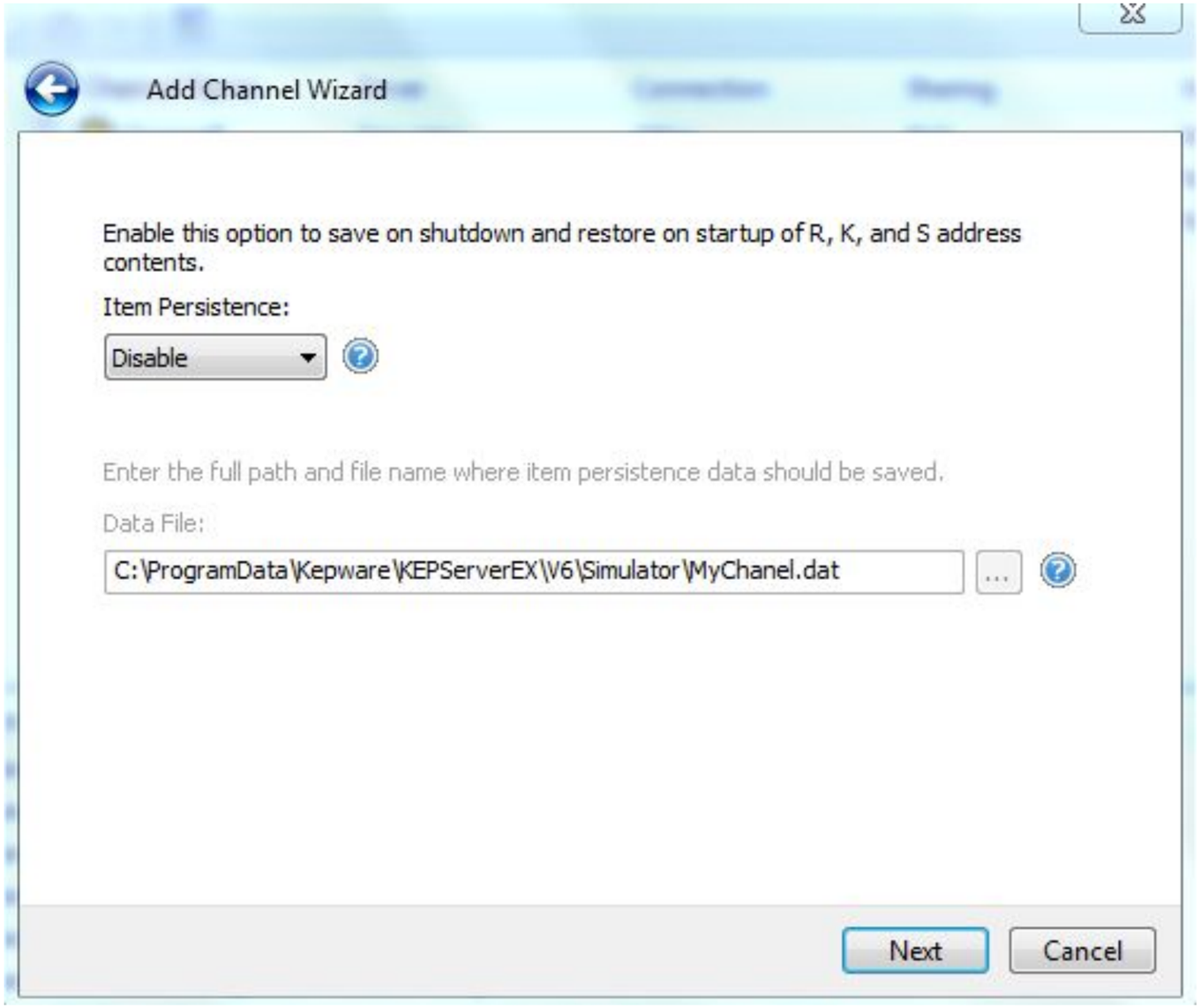


ОРС Клиент

Matlab + OPC

Создание тега для записи

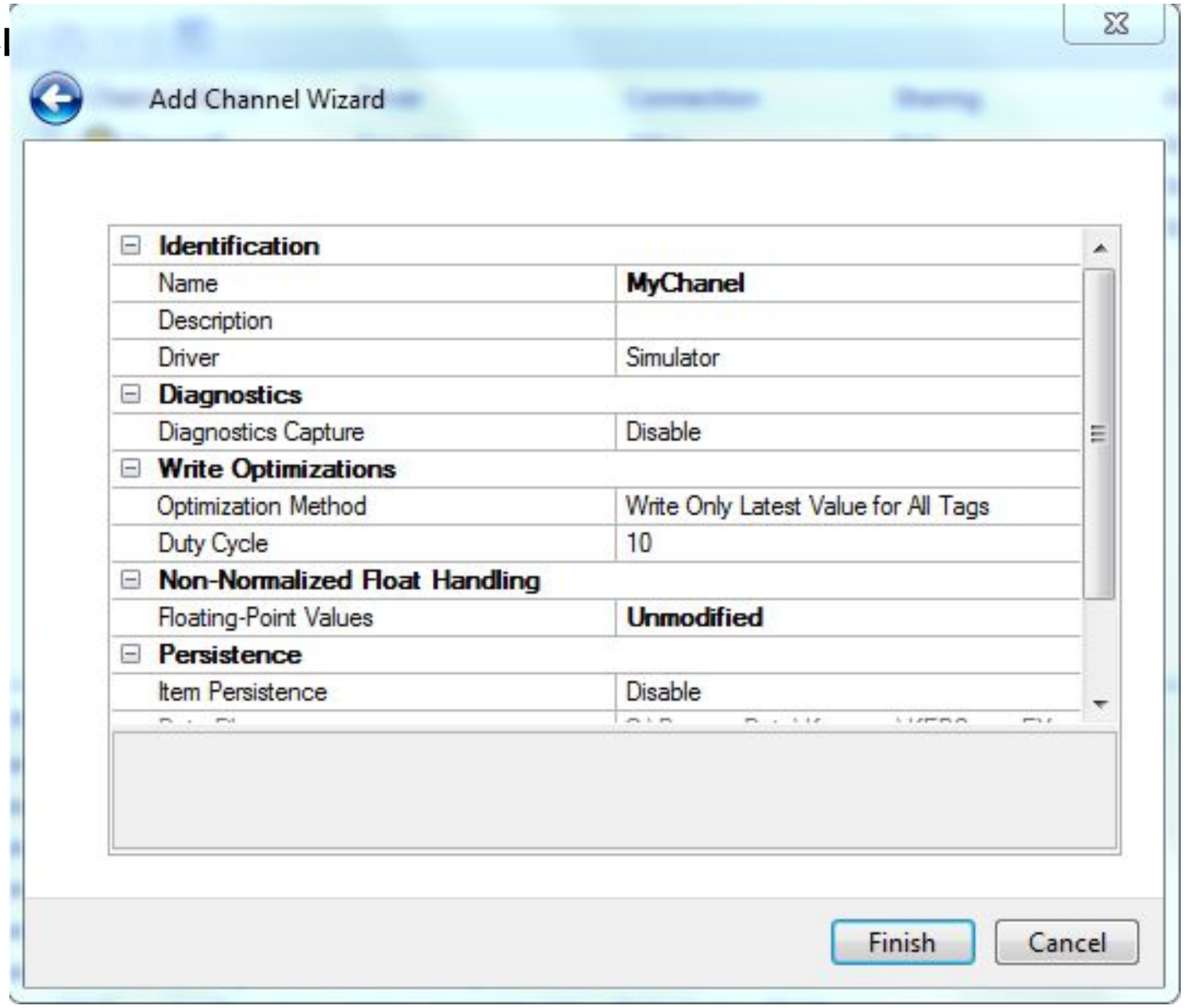




47 ОРС Клиент

Matlab + OPC

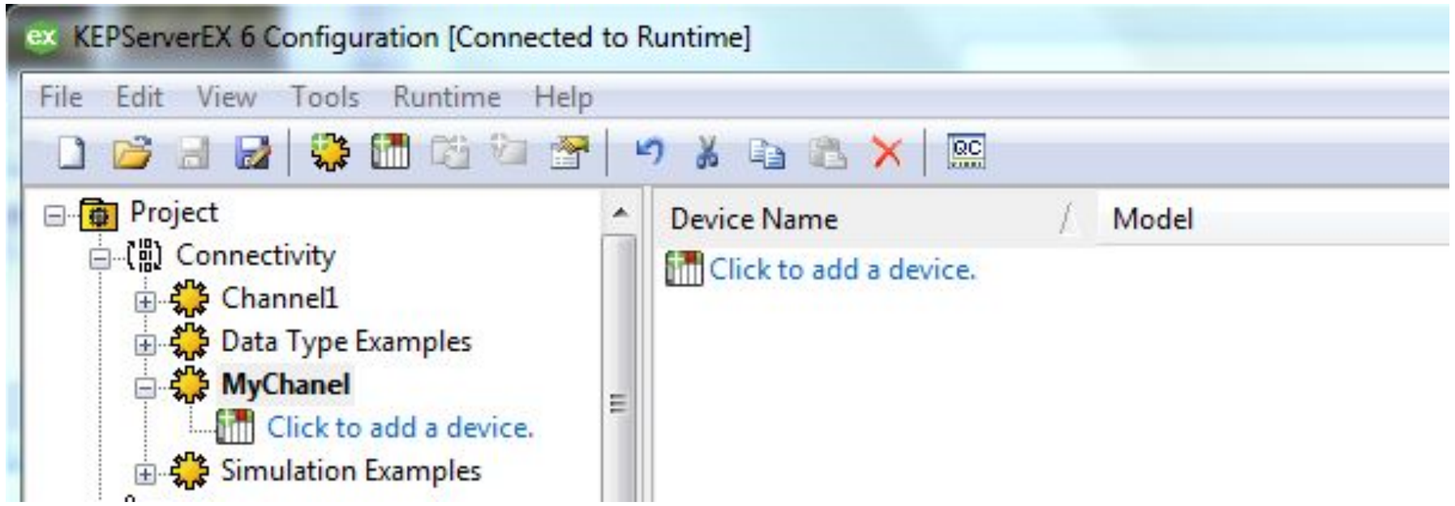
Создание тега для записи

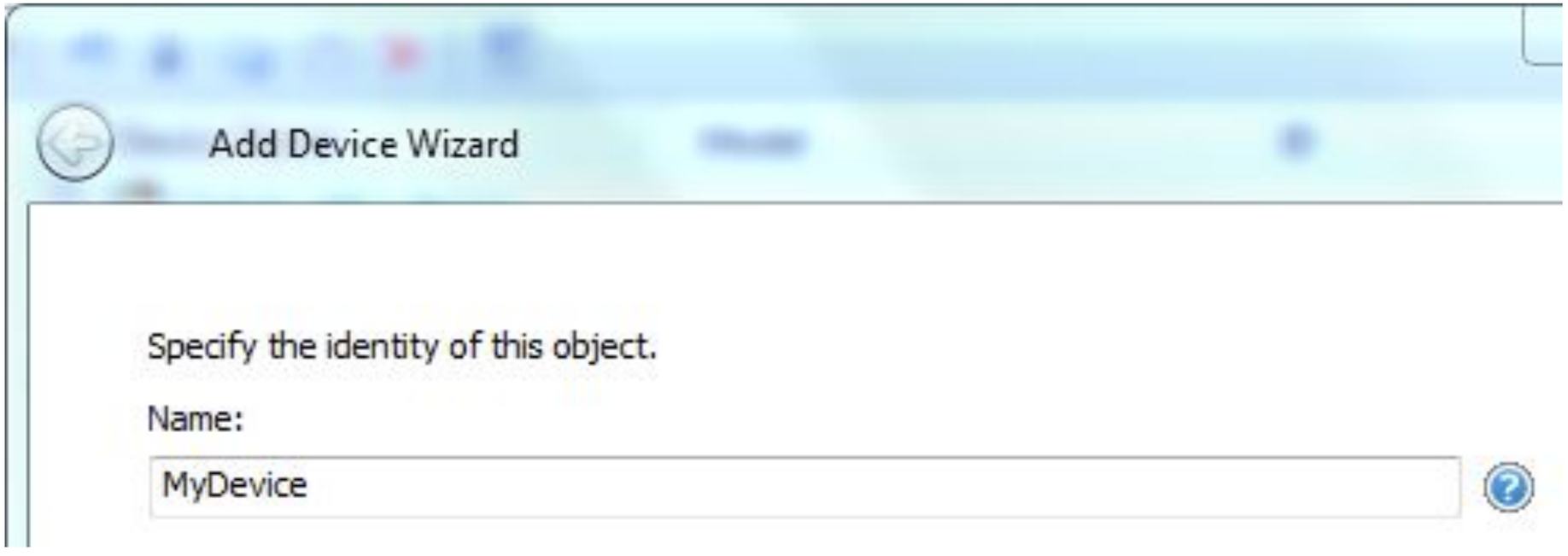


48 ОРС Клиент

Matlab + OPC

Создание тега для записи

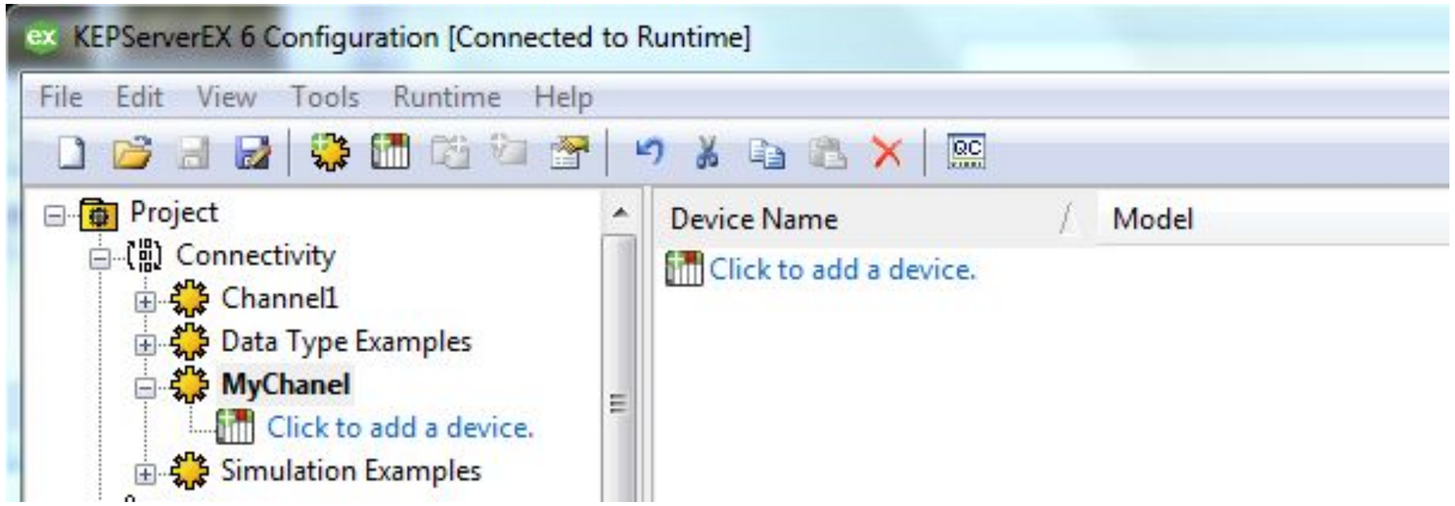


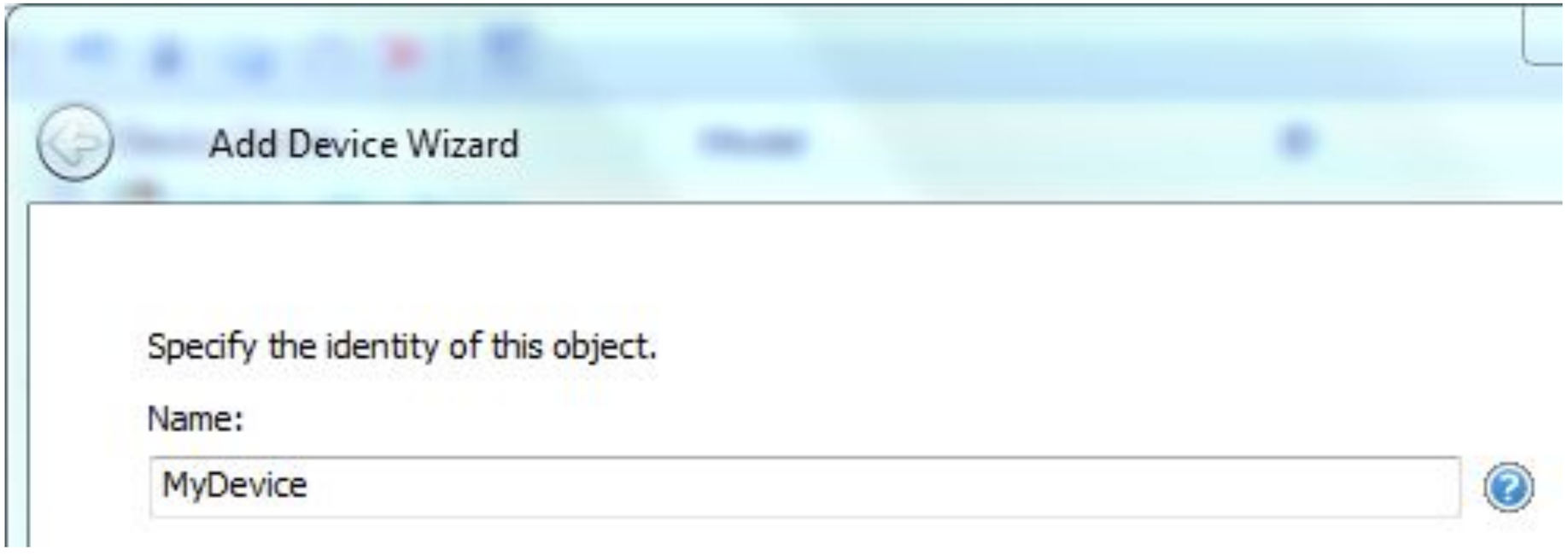


49 ОРС Клиент

Matlab + OPC

Создание тега для записи

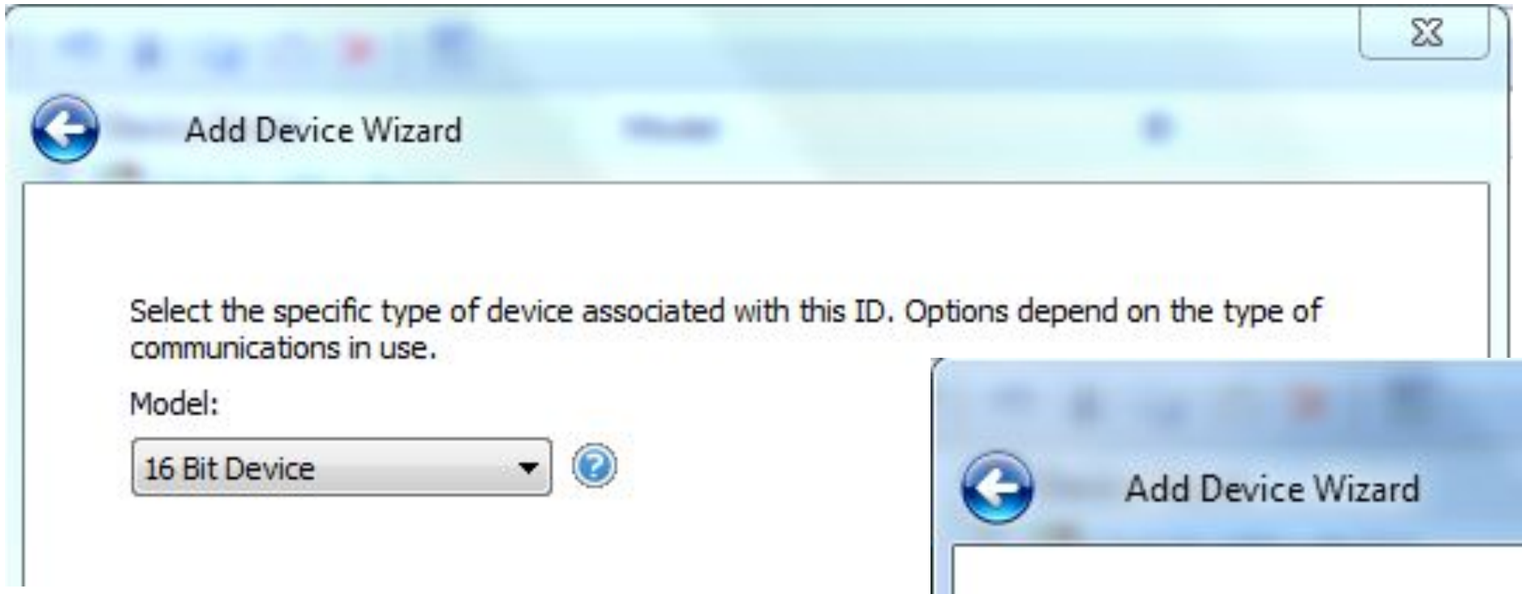


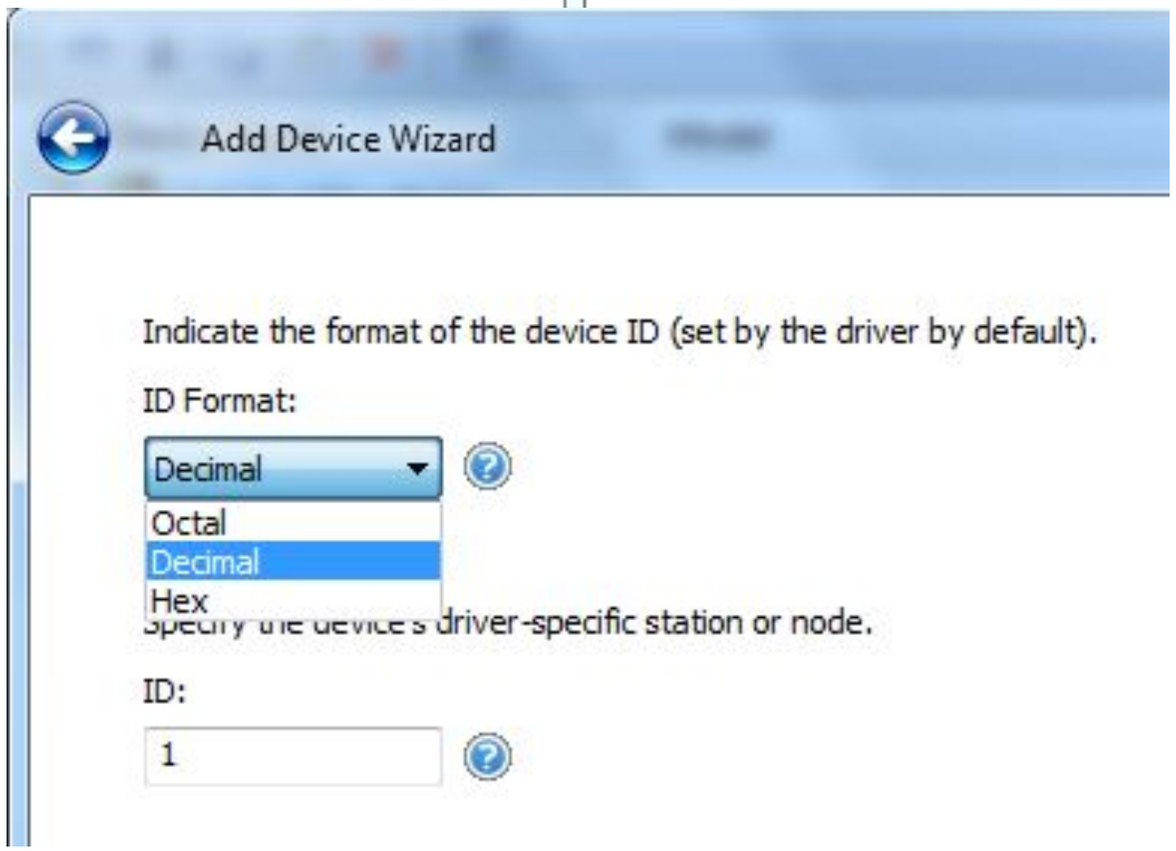


ОРС Клиент

Matlab + OPC

Создание тега для записи

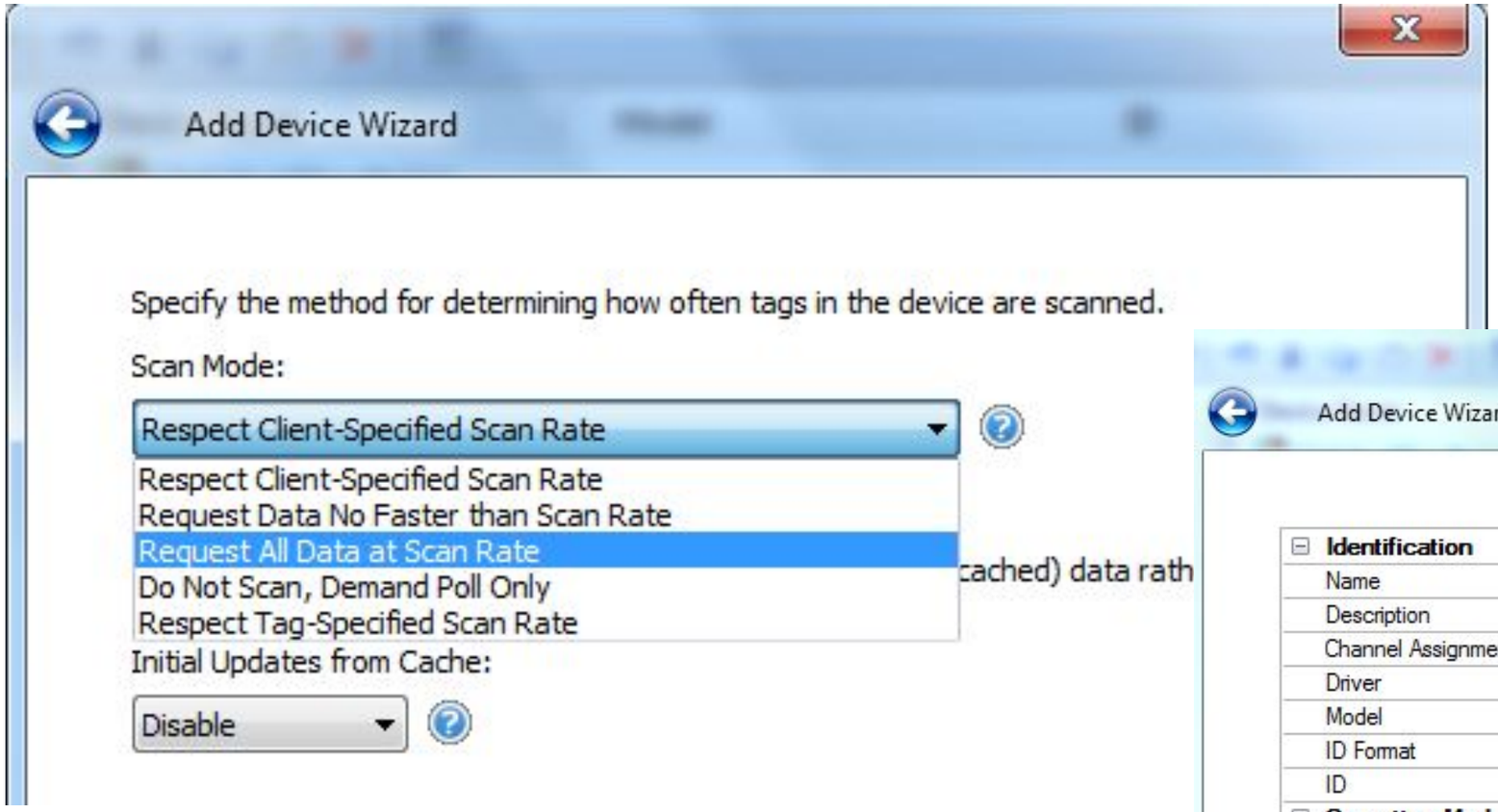


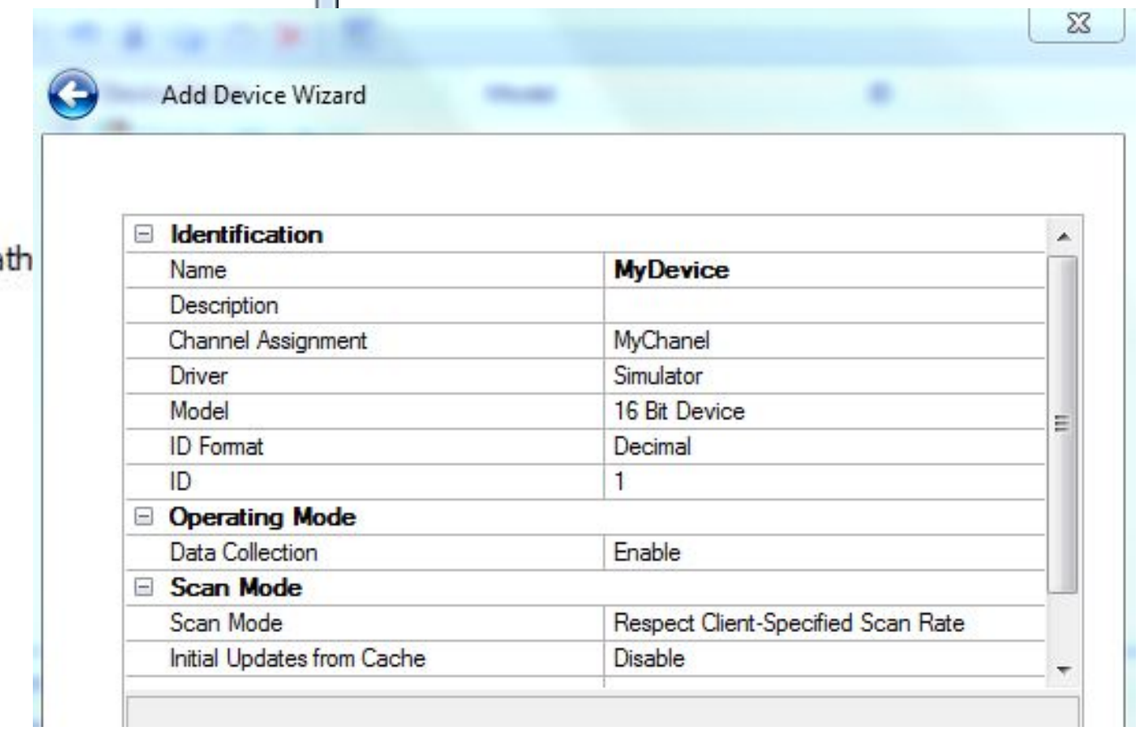


ОРС Клиент

Matlab + OPC

Создание тега для записи

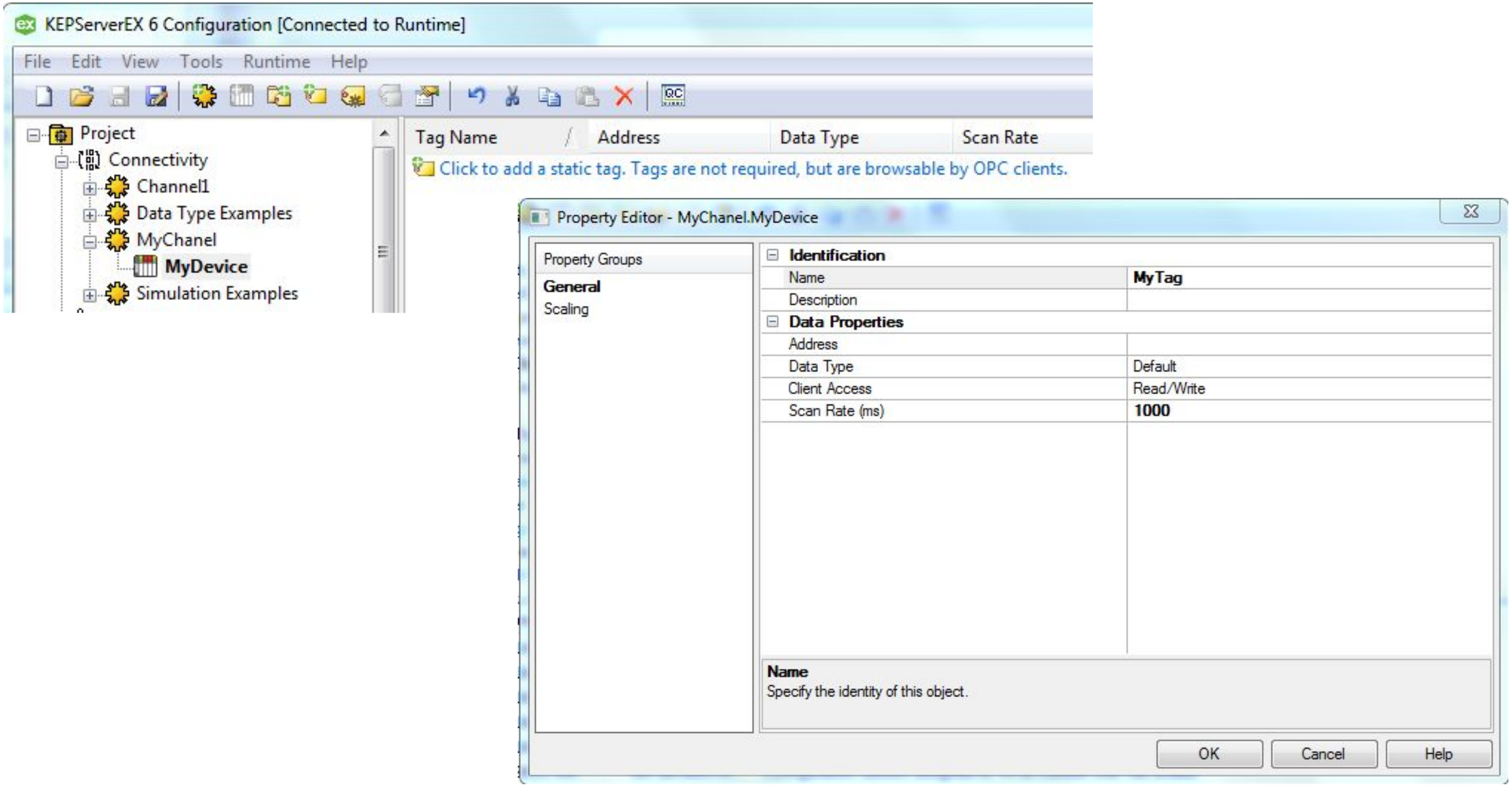




ОРС Клиент

Matlab + OPC

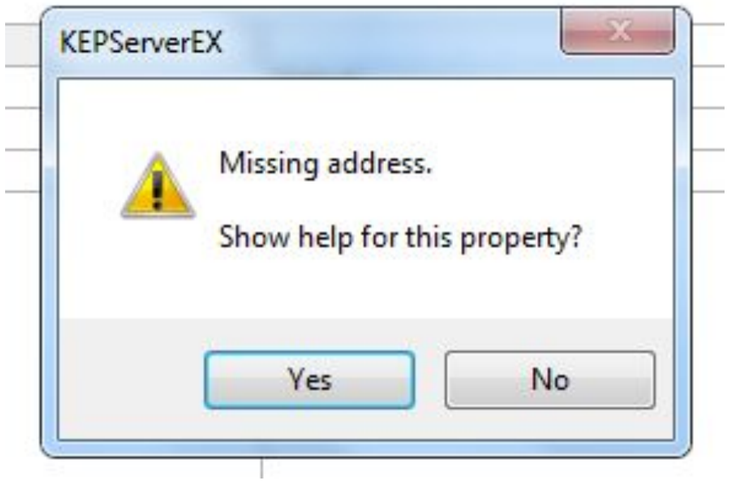
Создание тега для записи



ОРС Клиент

Matlab + OPC

Создание тега для записи



16-Bit Device Addresses

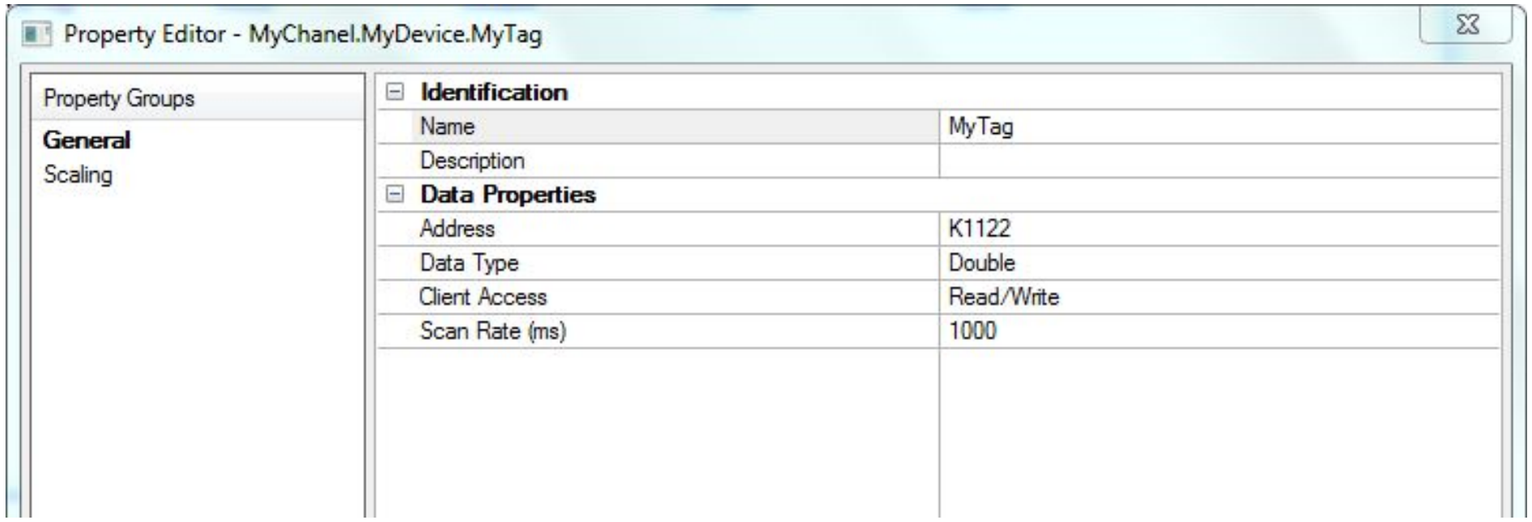
The memory configuration for the 16-Bit Device is simulated as a block of word locations numbered from 0 to 9999. Each word can be addressed as an offset from the start of the block. The default data types for each format are shown in **bold**.

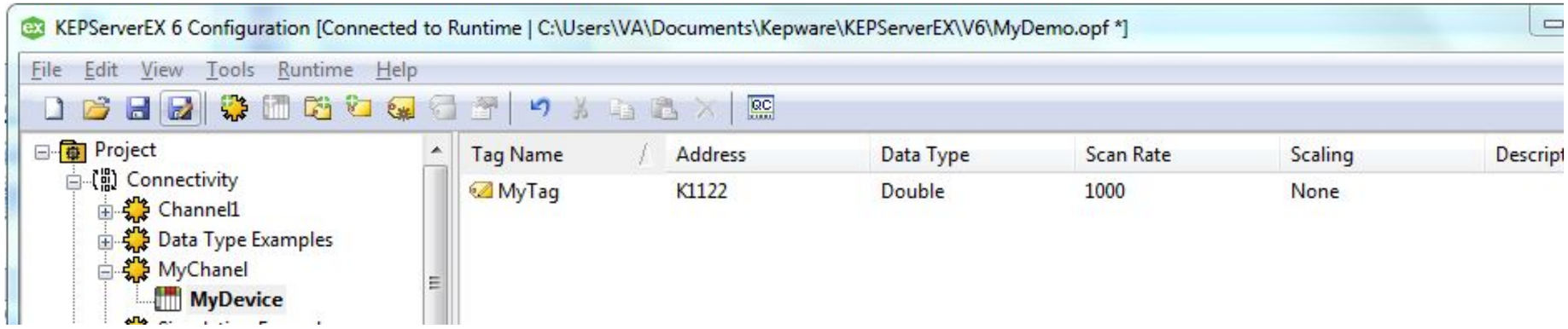
Device Type	Range	Data Type	Access
Registers	R0000-R9999 R0000-R9998 R0000-R9996	**Word**, Short, BCD DWord, Long, LBCD, Float LLong, QWord, Double, Date, Boolean	Read/Write
Constants	K0000-K9999 K0000-K9998 K0000-K9996	**Word**, Short, BCD DWord, Long, LBCD, Float LLong, QWord, Double, Date, Boolean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

ОРС Клиент

Matlab + OPC

Создание тега для записи

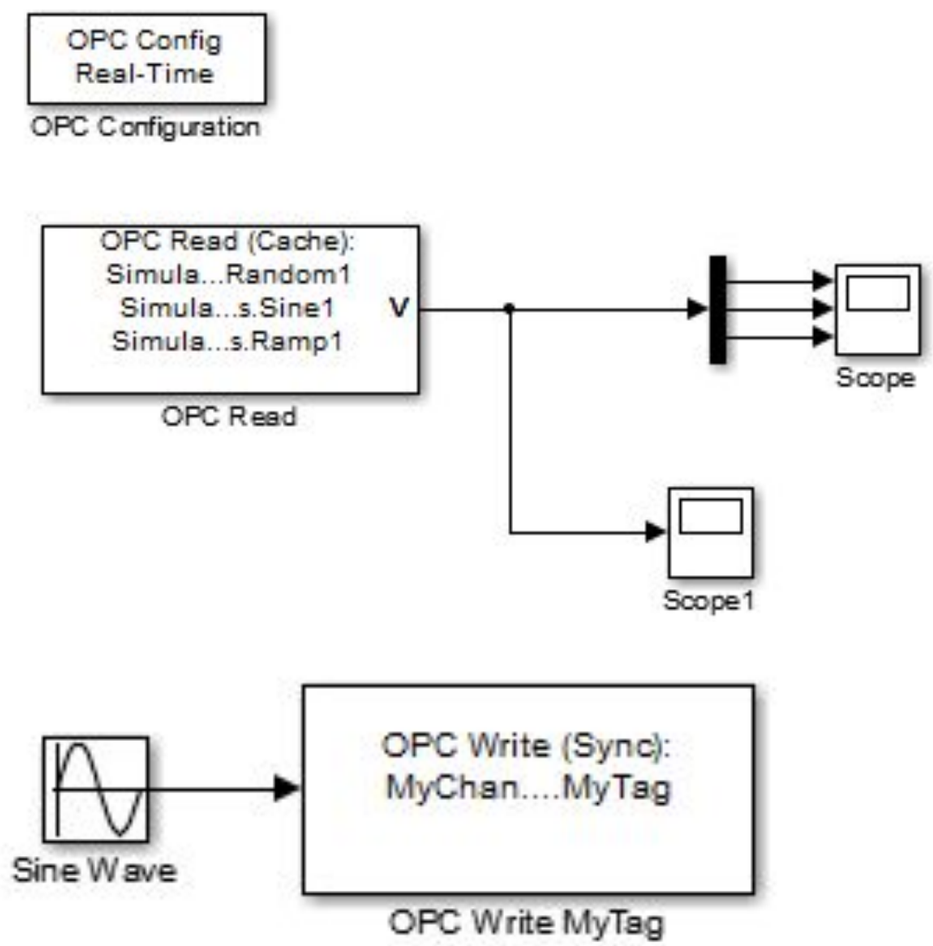


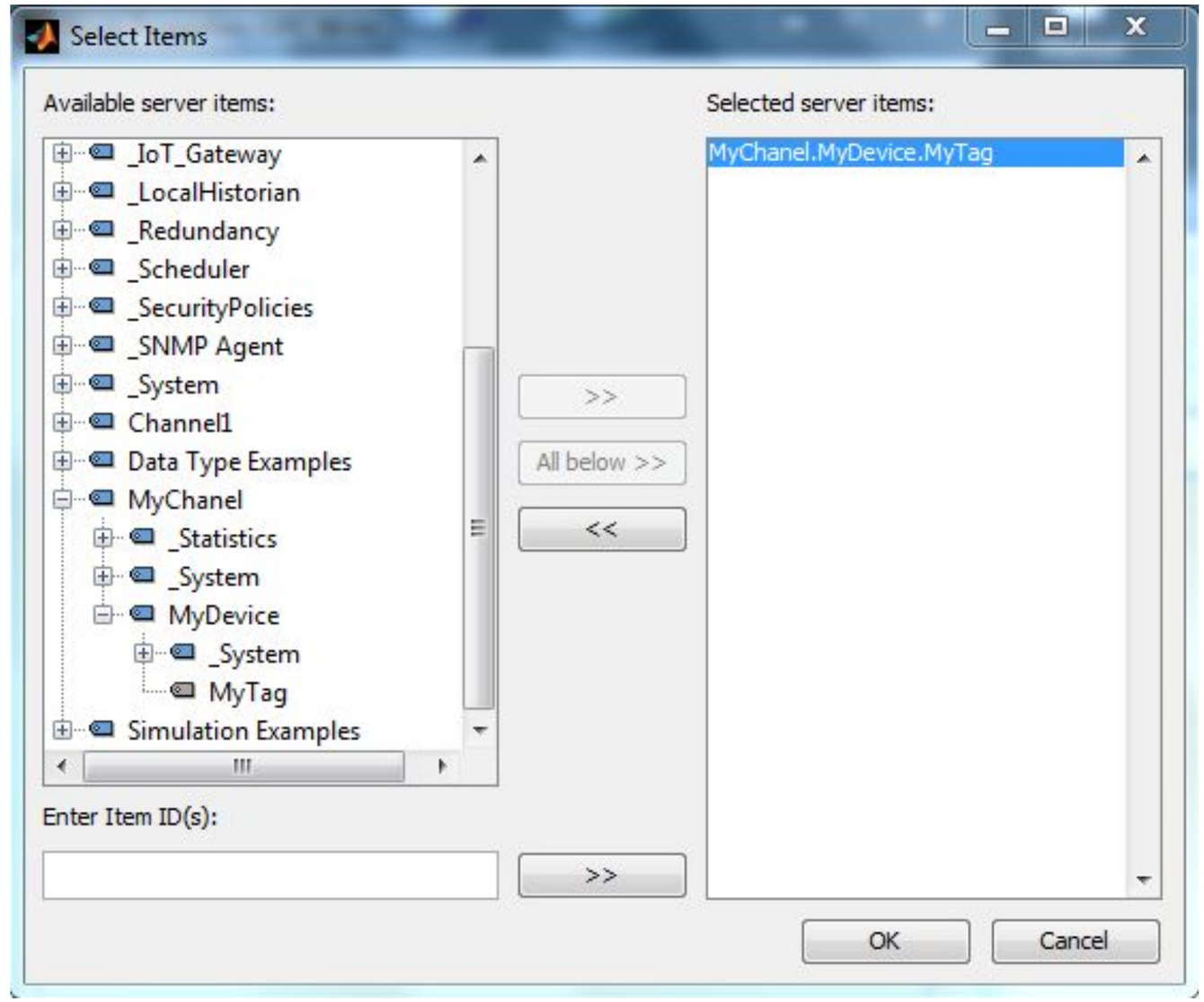


55 ОРС Клиент

Matlab + OPC

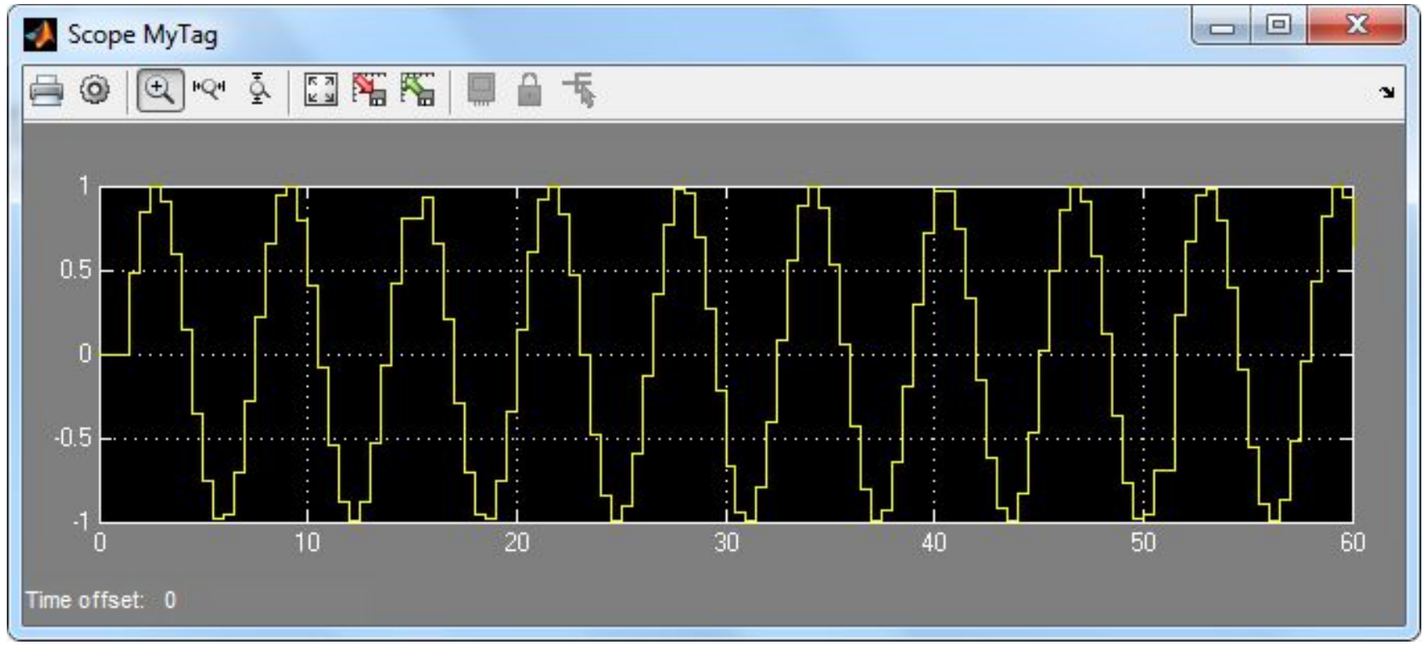
Модификация Simulink модели





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