

## LabView OPC клиент

*Цель работы:* освоение правил подключения LabView к OPC серверу.

*Задача работы:* построение канала передачи данных LabView - OPC сервер.

*Приборы и принадлежности:* Персональный компьютер, LabView, InSAT modbus OPC сервер, MatLAB.

Работа выполнена в среде **Windows XP**

## ОБЩИЕ СВЕДЕНИЯ

LabVIEW поддерживает работу по спецификации OPC Data Access в среде Windows. LabView имеет

- OPC сервер для подключения к внешнему клиенту,
- Средства для разработки OPC клиента и его подключения к внешнему OPC серверу.

Локальная OPC система поддерживает связь OPC клиента с модулями промышленной сети через OPC сервер с применением **COM** протокола.

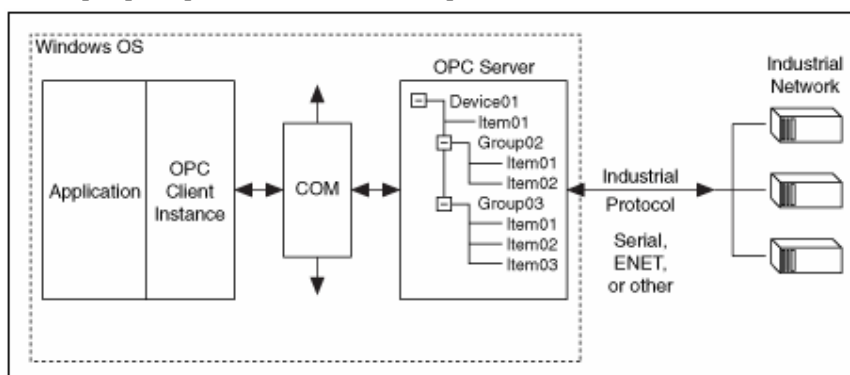
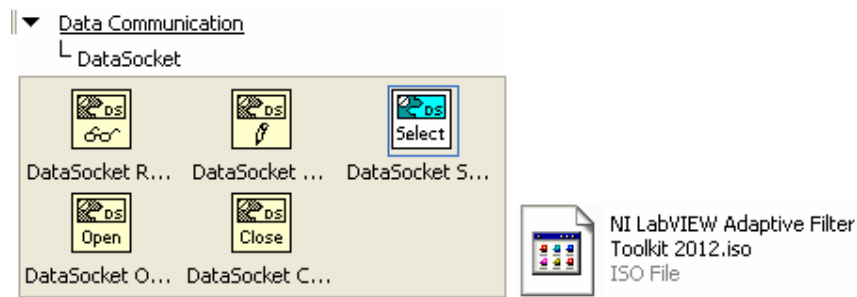


Рис. 1. Пример локальной OPC системы

Для построения удаленной OPC системы с применением **DCOM** протокола, необходимо правильно сконфигурировать следующие элементы системы:

- Windows security, включая установку соответствующих прав доступа пользователей и групп пользователей.
- DCOM
- Firewalls
- OPC сервер
- OPC клиент

Для разработки OPC клиента в LabView можно использовать модули библиотеки **DataSocket** и пакет “**Datalogging and Supervisory Control**” (**DSC**).



**Рис. 2.** Средства для построения OPC клиента в LabView.

Примечание. Datalogging и модули “Supervisory Control” и “Real-Time” поддерживают спецификацию OPC Unified Architecture (UA). Используя виртуальные инструменты **VI** можно построить OPC UA сервер или клиент приложения.

**Note** The LabVIEW Datalogging and Supervisory Control Module and the LabVIEW Real-Time Module also support the OPC UA specification. You can use the [OPC UA](#) VIs to create OPC UA server or OPC UA client applications.

## Labview DataSocket

DataSocket позволяет открыть тег OPC сервера при помощи OPC протокола. DataSocket применяется в следующих случаях, когда

- не требуется автоматическое логирование данных, установки предупреждений (alarming), масштабирования и защиты данных.
- необходимо малое количество переменных - один или два тега.
- не требуется изменять параметры сервера, такие как частота обновлений, или время отклика (deadband).
- версия OPC сервера 1.0 или 2.x.

Примечание. Для более поздних версий необходимо использовать пакет DSC.

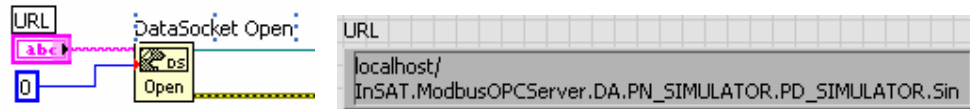
Функции DataSocket Open and DataSocket Close используются для открытия и закрытия соединения передачи данных.

Соединение с OPC сервером можно установить в следующем порядке.

1. Добавьте **DataSocket Open** в блок диаграмму LabView.
2. Подсоедините **URL** ко входу блока, например,

```
opc:\InSAT.ModbusOPCServer.DA\PN_SIMULATOR.PD_SIMULATOR.Sin  
opc:\InSAT.ModbusOPCServer.DA\Shared_Vars.PD_SIMULATOR.Sin  
opc:\InSAT.ModbusOPCServer.DA\Shared_Vars.PD_SIMULATOR.PollDevice  
opc:\National Instruments.OPCTest\item1  
opc:\computer\National Instruments.OPCModbus\Modbus Demo Box.4:0  
opc:\computer\National Instruments.OPCModbus\Modbus Demo  
Box.4:0?updaterate=100&deadband=0.7
```

3. Подсоедините константу mode (режим работы) к блоку DataSocket Open: 0 – чтение; 1 – запись; 2 – чтение/запись; 3 - чтение блоком (**BufferedReader**); 4 – запись блоком (**BufferedReadWrite**).



4. Добавьте в блок диаграмму **DataSocket Read** или **DataSocket Write**.
5. Добавьте в блок диаграмму **DataSocket Close**
6. Соедините блоки диаграммы.

### Пакет DSC (в этой работе заданий по DSC нет)

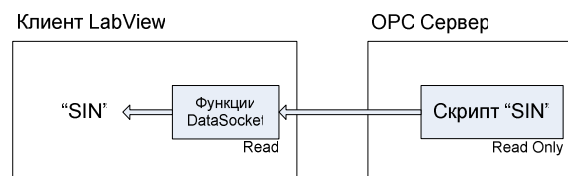
Пакет DSC, имеющий графический интерфейс, обеспечивает быструю разработку распределенной системы измерения, управления и наблюдения. Можно легко установить связь переменных LabVIEW с тегами OPC сервера..

DSC пакет обеспечивает оптимальную работу в случаях, когда

- необходимо автоматическое логирование данных или использование предупреждений, масштабирования и защиты данных.
- имеется большое количество OPC переменных. DSC может работать со всеми OPC серверами установленными на компьютере или в сети, считывать информацию о возможностях сервера и тегах непосредственно с OPC сервера. DSC позволяет изменять параметры сервера, например, частоту обновлений.
- необходим доступ к тегам OPC от общих переменных.
- необходимо подключить OPC сервер версии 3.0.

### ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

**Задание 1.** Считывание и отображение OPC данных в LabView средствами DataSocket по следующей схеме.



1. Настройте OPC сервер на выдачу синусоидального сигнала (см. Рис. 3).

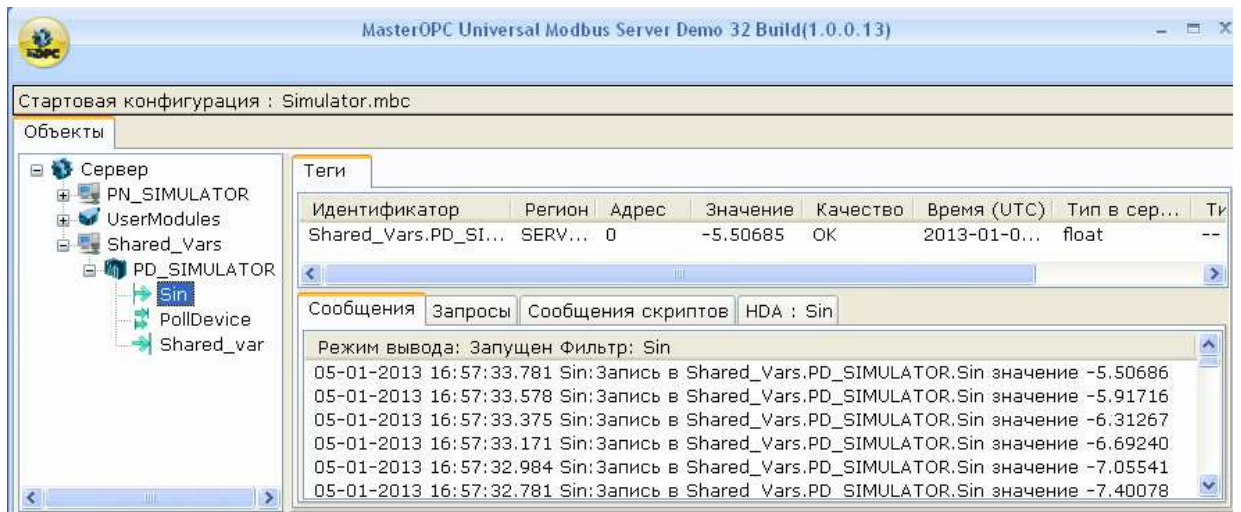


Рис. 3. Состояние переменной SIN вычисляемой OPC сервером.

2. Соберите в LabView схему отображения параметров, передаваемых OPC сервером, включая величины синусоидального сигнала и время регистрации OPC сервером соответствующих величин сигнала (см. Рис. 4 и Рис. 6).

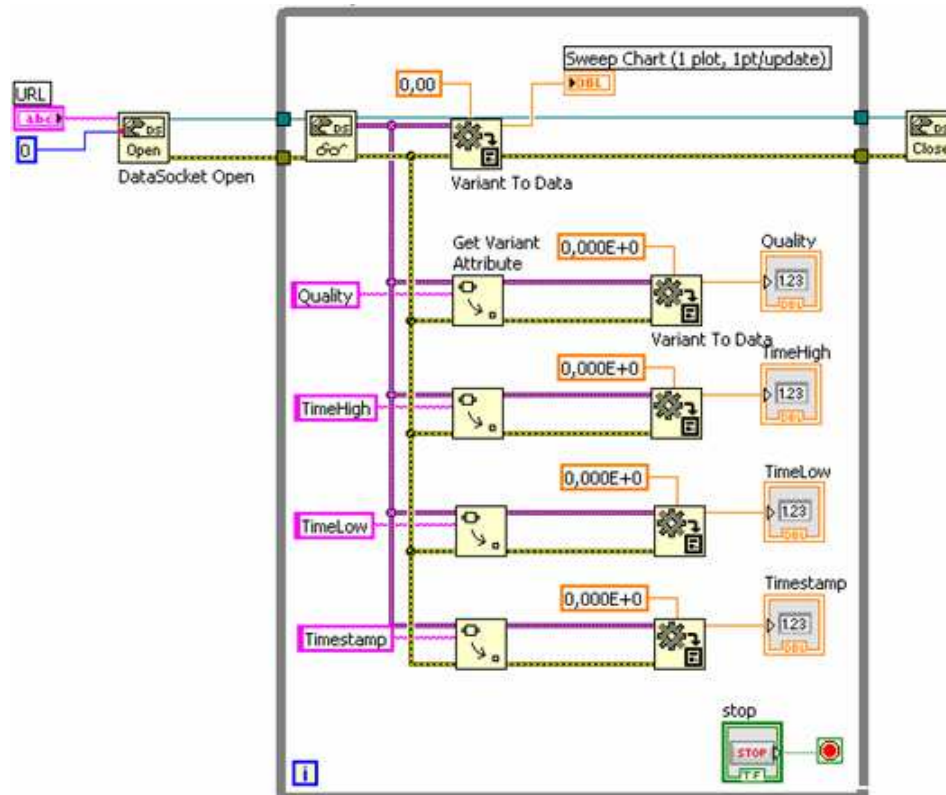


Рис. 4. Блок-диаграмма клиента LabView.

3. Проверьте работоспособность схемы, сравните с результатами Рис. 5 и Рис. 6.

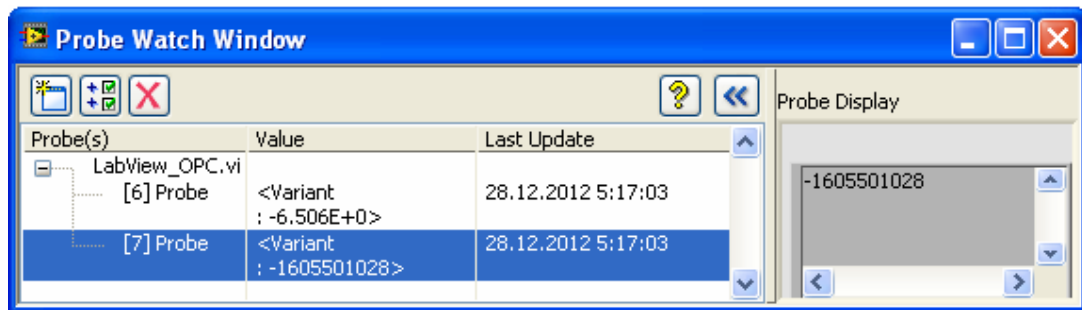


Рис. 5. Отображение переменной клиента LabView в точке “Probe”.

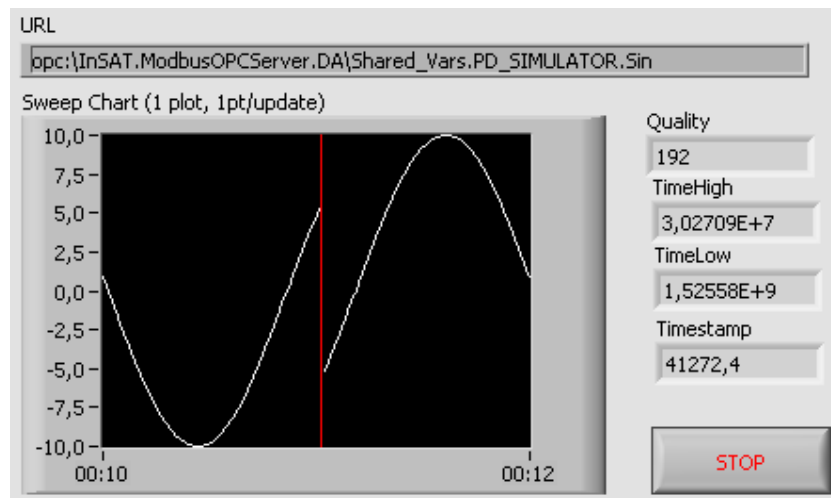
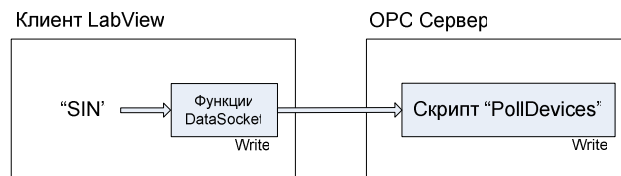


Рис. 6. Интерфейс клиента LabView.

**Задание 2.** Передача переменной LabView в OPC сервер.



Используя показанные скрипт OPC сервера и блок-диаграммы LabView соберите и проверьте работоспособность генератора синусоидального сигнала в LabView. Переменная OPC сервера `opc:\InSAT.ModbusOPCServer.DA\Shared_Vars.PD_SIMULATOR.PollDevice` должна отслеживать состояние сигнала LabView.

```

01 local tag
02 -- инициализация
03 function OnInit()
04 end
05 -- деинициализация
06 function OnClose()
07 end
08 -- обработка
09 function OnWrite()
10   tag = server.ReadCurrentTag();
11   if tag then
12     server.WriteCurrentTag(false, OPC_QUALITY_GOOD);
13     server.PollCurrentDevice();
14   end
15 end

```

Рис. 7. Скрипт OPC переменной.

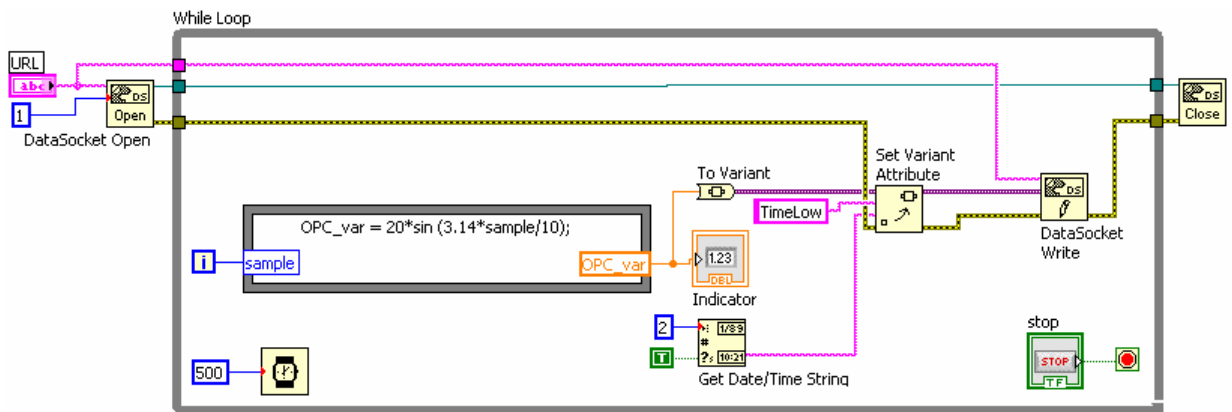
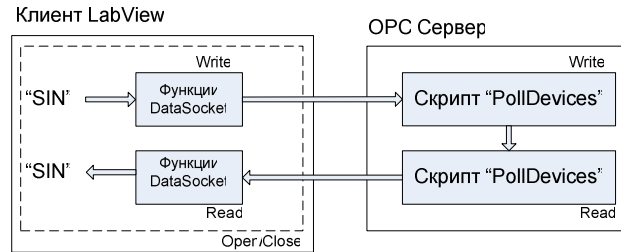


Рис. 8. Блок-диаграмма клиента LabView.

Идентификатор	Регион	Адрес	Значение	Качество	Время (UTC)	Тип в сер...	Тип в уст...	Доступ
Shared_Vars.PD_SI...	SERV...	0	-19.3681	OK	2013-01-0...	float	--	ReadWrite

Рис. 9. Состояния OPC сервера и LabView.

### Задание 3. Запись и считывание переменной LabView через OPC сервер.



Используя показанные ниже состояние и скрипты OPC сервера и блок-диаграммы LabView соберите канал записи синусоидального сигнала LabView в ОС сервер и считывание записанных данных обратно в LabView. Переменная **opc:\InSAT.ModbusOPCServer.DA\Shared\_Vars.PD\_SIMULATOR.PollDevice** должна отслеживать значение передаваемых данных.

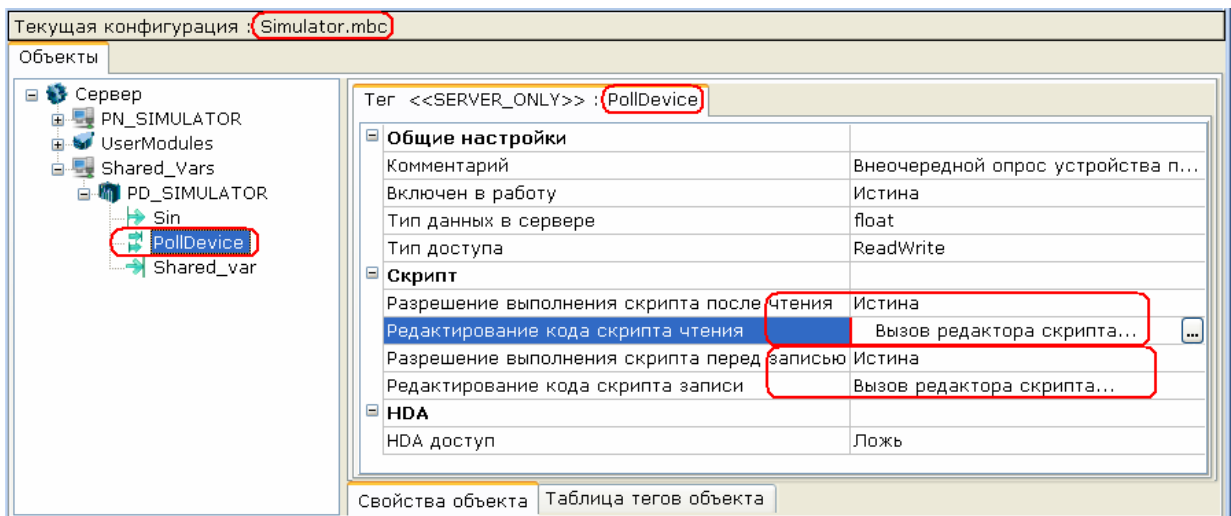


Рис. 10. Переменная PollDevice.

Примечание. Редактор скрипта OPC сервера вызывается кнопкой .

```

01  -- инициализация
02  function OnInit()
03      server.WriteCurrentTag(false, OPC_QUALITY_GOOD);
04  end
05  -- деинициализация
06  function OnClose()
07  end
08  -- обработка
09  function OnRead()
10  end
    
```

```

01 local tag
02 -- инициализация
03 function OnInit()
04 end
05 -- деинициализация
06 function OnClose()
07 end
08 -- обработка
09 function OnWrite()
10 tag = server.ReadCurrentTag();
11 if tag then
12     server.WriteCurrentTag(false, OPC_QUALITY_GOOD);
13     server.PollCurrentDevice();
14 end
15 end

```

Рис. 11. Скрипты OPC переменной.

Стартовая конфигурация : Simulator.mbc

Объекты

- Сервер
  - PN\_SIMULATOR
  - UserModules
  - Shared\_Vars
    - PD\_SIMULATOR
      - Sin
      - PollDevice
      - Shared\_var

Теги

Идентификатор	Регион	Адрес	Значение	Качество	Время (UTC)	Тип в сер...	Тип в уст...	Доступ
Shared_Vars.PD_SI...	SERV...	0	11.17638	OK	2013-01-0...	float	--	ReadWr

Сообщения | Запросы | Сообщения скриптов

Режим вывода: Запущен Фильтр: PollDevice

```

04-01-2013 21:34:43.296 PollDevice:Запись в Shared_Vars.PD_SIMULATOR.PollDevice значение 11.17639
04-01-2013 21:34:43.109 PollDevice:Запись в Shared_Vars.PD_SIMULATOR.PollDevice значение 5.50715
04-01-2013 21:34:42.906 PollDevice:Запись в Shared_Vars.PD_SIMULATOR.PollDevice значение -0.70062
04-01-2013 21:34:42.703 PollDevice:Запись в Shared_Vars.PD_SIMULATOR.PollDevice значение -6.83989
04-01-2013 21:34:42.500 PollDevice:Запись в Shared_Vars.PD_SIMULATOR.PollDevice значение -12.31029

```

Рис. 12. Работа OPC сервера.

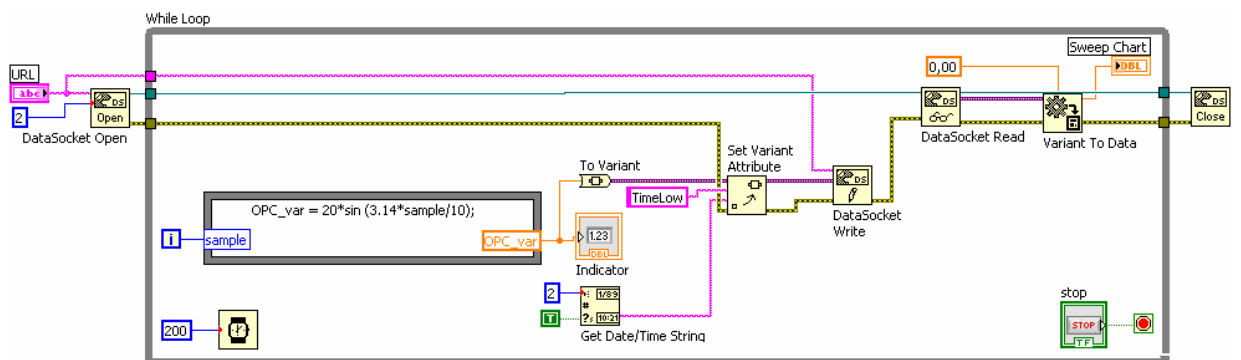
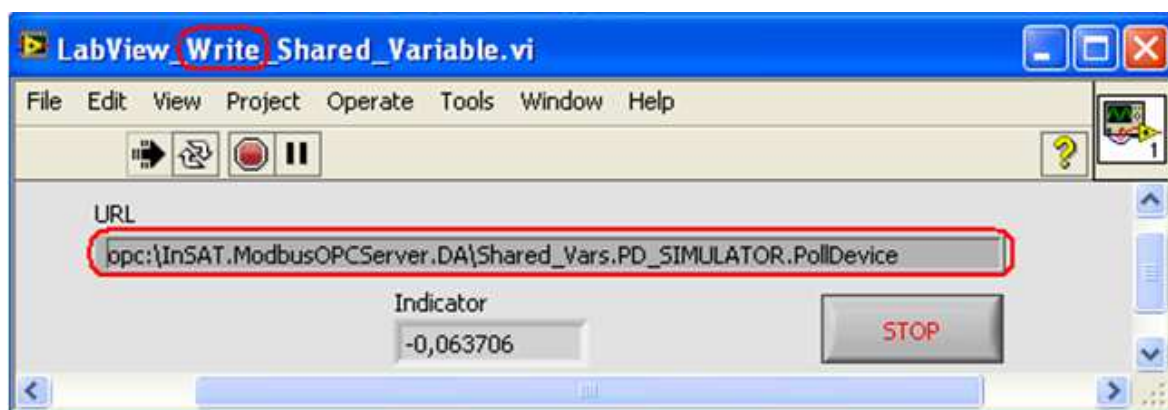
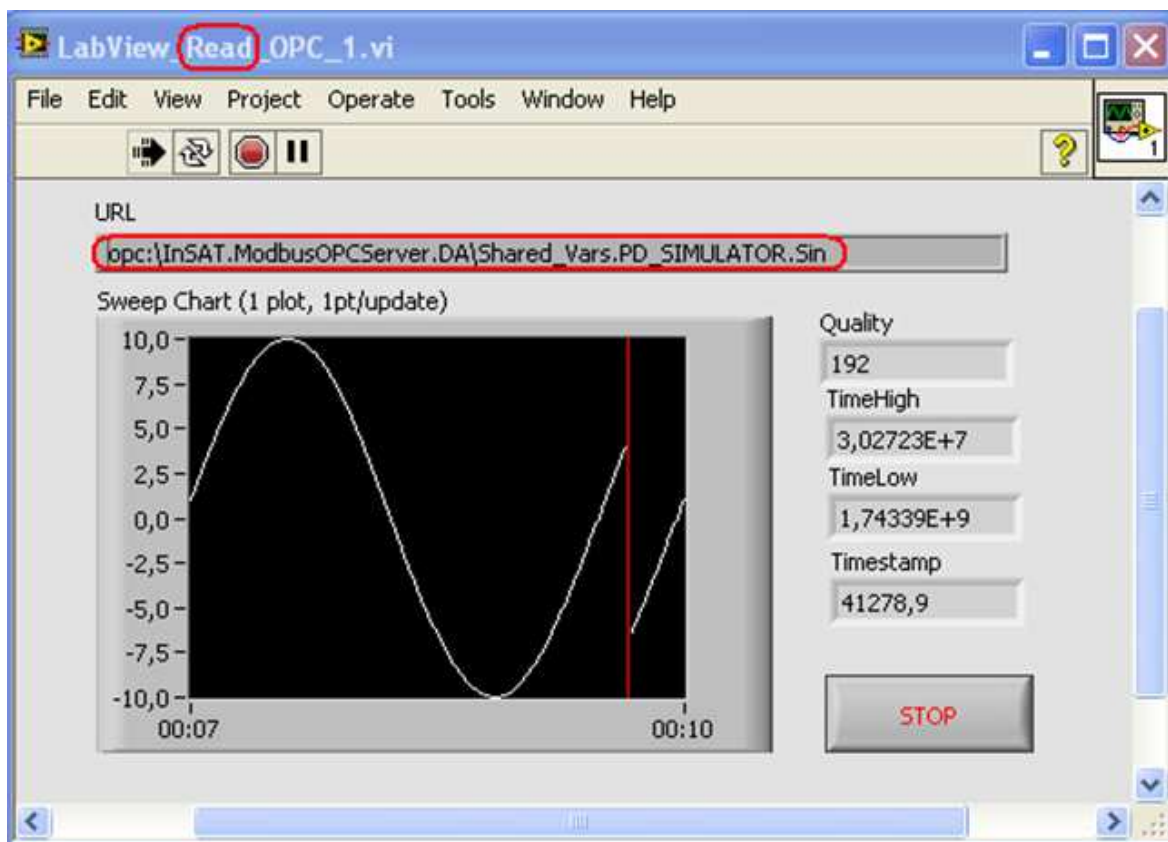


Рис. 13. Блок-диаграмма клиента LabView





**Рис. 14.** Пример совместной работы двух процессов LabView клиента и InSAT Modbus OPC сервера. Один процесс считывает OPC данные, другой – вычисляет и передает данные OPC серверу. Сервер отображает значения переменных и состояние канала обмена данными.

**Задание 4.** Запись и считывание удвоенного значения переменной LabView через OPC сервер.

Доработайте предыдущее задание так, чтобы OPC сервер увеличивал в два раза принимаемую от LabView переменную прежде чем вернуть ее обратно в LabView.

### Задание 5. Соединение MatLAB - OPC и LabView - OPC сервер.

1. Соберите каналы передачи данных для следующих виртуальных устройств.

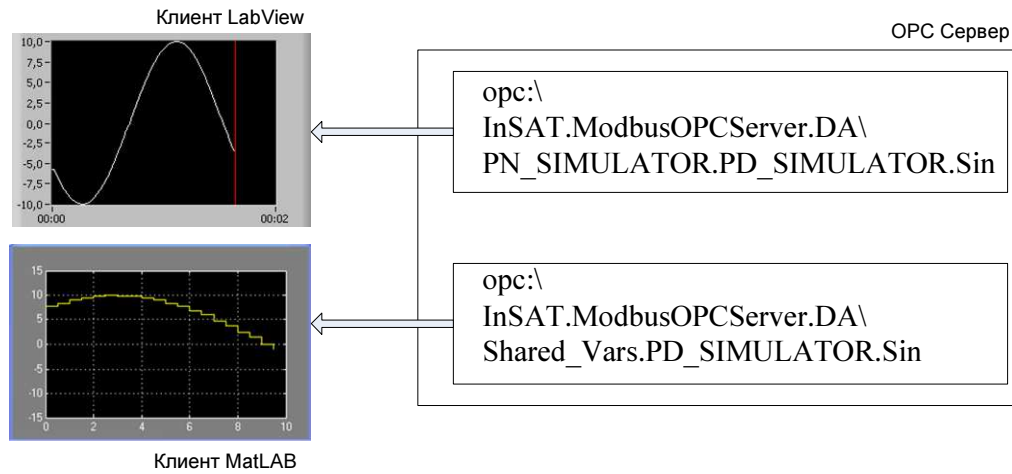


Рис. 15. Структурная схема OPC соединений.

2. Снимите основные характеристики канала.

### Задание 6. Соединение MatLAB - OPC - LabView.

1. Соедините MatLAB и LabView через общий тег OPC сервера.

MatLAB OPC клиент -

opc:\InSAT.ModbusOPCServer.DA\Shared\_Vars.PD\_SIMULATOR.PollDevice

LabView OPC клиент -

opc:\InSAT.ModbusOPCServer.DA\Shared\_Vars.PD\_SIMULATOR.PollDevice

2. Выясните, почему такая связь не работает?

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как задается режим реального времени при работе с OPC серверами?
2. Перечислите все параметры тега принимаемого LabView в первом задании.
3. Как получить краткую справку по объектам блок диаграммы в LabView?
4. Почему нельзя обеспечить устойчивую циклическую (while) запись и циклическое (while) чтение одной и той же OPC переменной двумя LabView процессами – записи и чтения переменной?

### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Help LabView.
2. Dr. Bob Davidov. Компьютерные технологии управления в технических системах <http://portalnp.ru/author/bobdavidov>