

Изображение лабораторного стенда показано на рис. 1. Стенд представляет собой два бака с жидкостью, которая перекачивается насосом. Так же на стенде имеются различные датчики: давления, расхода, температуры, уровня; клапаны, нагревательный элемент.

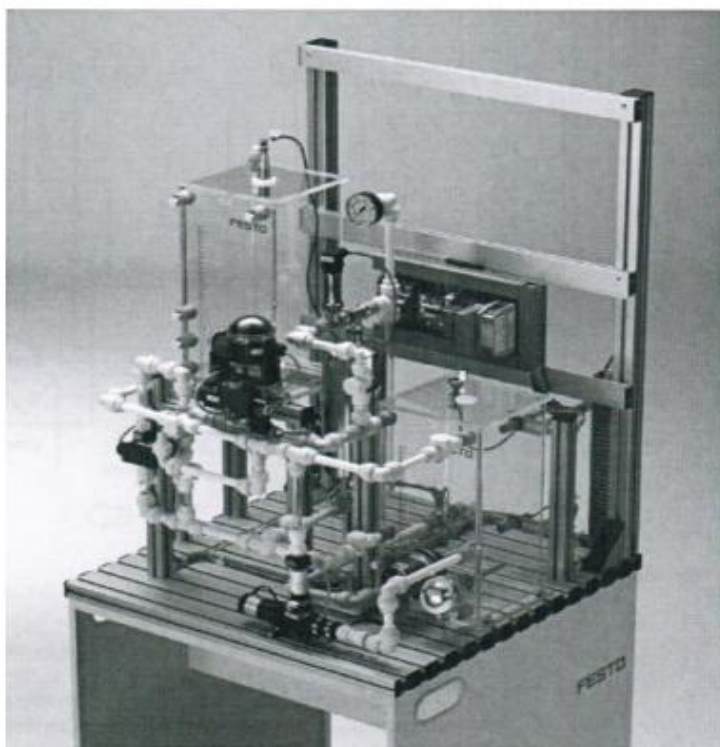


Рис. 1. Изображение лабораторного стенда.

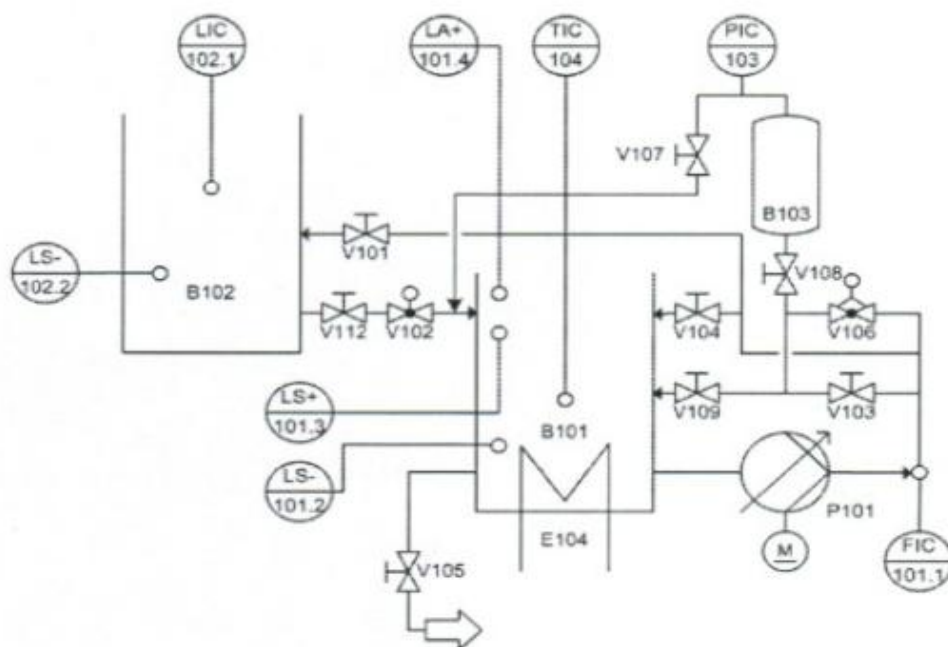


Рис. 2. Функциональная схема лабораторного стенда.

Лабораторный стенд оснащен 4 интегрированными системами. Как показывает схема технологического процесса, отдельные системы управления могут быть активированы путем переключения ручных клапанов. Гибкая система трубо-

проводов позволяет осуществлять быстрое изменение схемы технологического процесса или интеграцию дополнительных компонентов.

Основными компонентами КРС являются:

Ультразвуковой датчик уровня (LIC 102)

Расходомер с частотным сигналом (FIC 101)

Датчик давления электронный (PIC 103)

Датчик давления механический

Термометр сопротивления PT100 (TIC 104)

Два емкостных датчика уровня (LS+ 101, LS- 101)

Поплавкового порогового датчика в верхнем резервуаре (LS- 102)

Поплавкового датчика переполнения в нижнем резервуаре (LA+ 101)

Центробежный насос (P101)

Мотор-контроллер для центробежного насоса

Пропорциональный клапан (V106) с электронным управлением

Дискретный нагревательный элемент (E104) с управлением от реле

Шаровой клапан (V102) с пневматическим приводом

Терминал ввода/вывода

Преобразователи сигналов: ток, частота, сопротивление в напряжение
PLC

Панель контроля

Система трубок, включая 4 прозрачных сегмента

Резервуар давления

2 резервуара для жидкости

Ручные клапаны для изменения конфигурации

Ручной клапан для слива (V105)

Монтажная панель

Краткое описание основных элементов и систем лабораторного стенда:

Центробежный насос



Рис 3. Центробежный насос.

Питание: 24 В постоянного тока
Сигнал: 0-10 В постоянного тока
Мощность: 26 Ватт
Соединения: 20мм (3/4")

Пропорциональный клапан

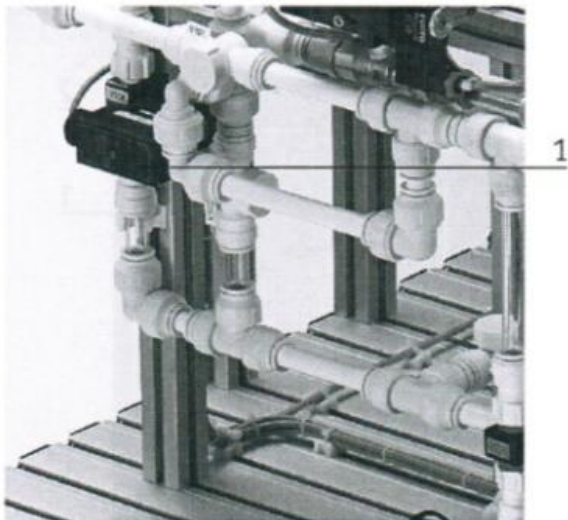


Рис 4. Пропорциональный клапан

Пропорциональный клапан с электронным управлением регулирует поток жидкости проходящей через него изменением степени закрытия от 0 до 100%.

Питание: 24 В постоянного тока
Сигнал: 0-10 В постоянного тока
Соединения: G 1/4

Шаровой клапан с пневматическим приводом

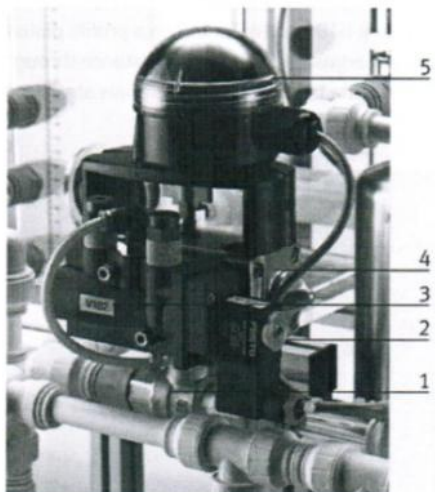


Рис 5. Шаровой клапан с пневматическим приводом

Шаровой клапан с пневматическим приводом V102 пропускает либо перекрывает поток жидкости, перетекающей из верхнего резервуара в нижний резервуар.

Данный элемент включает в себя:

Шаровой клапан

Соленоид

5/2 распределитель

Поворотный пневматический привод

Сигнальный элемент

Питание: 0-250 В переменного тока

Сигнал: 0-30 В постоянного тока

Нагревательный элемент с релейным управлением



Рис 6. Нагревательный элемент с управлением от реле

Ультразвуковой датчик уровня



Рис 7. Ультразвуковой уровнемер

Принцип его действия основан на генерации ультразвуковых волн, и последующем приеме отраженных от поверхности раздела волн и измерения сдвига их фаз.

Питание элемента: 230 В переменного тока

Сигнал на реле: 24 В постоянного тока

Соединения: G ¼

Контроль уровня жидкости в резервуарах производится с помощью:

Емкостных датчиков B113 и B114

Поплавкового датчика S111

Поплавкового датчика S112

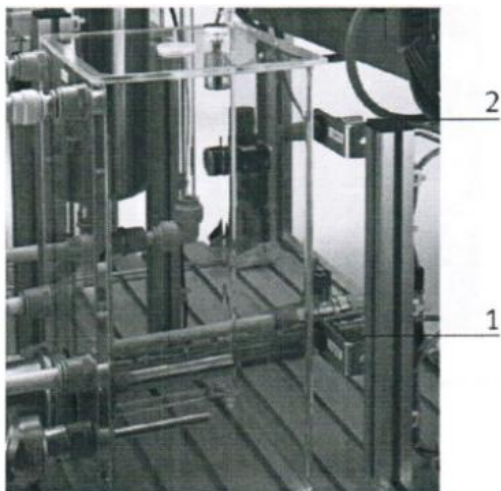


Рис 8. Контроль уровня жидкости в нижнем резервуаре с помощью двух емкостных датчиков B113(1) и B114(2)

С помощью данных емкостных датчиков задается минимально и максимально допустимые уровни жидкости в нижнем резервуаре, для того чтобы предотвратить его переполнение либо попадание воздуха в центробежный насос.



Рис 9. Защита от переполнения верхнего резервуара с помощью поплавкового датчика S111(1)

В случае достижения уровнем жидкости в верхнем резервуаре критического значения поплавки всплывает и замыкает контакт, тем самым подавая сигнал на контроллер либо EasyPort. Программу следует настроить так, чтобы при срабатывании поплавкового датчика S111 выключался центробежный насос P101 и открывался шаровой клапан V102.



Рис. 10. Поплавковый датчик

Увеличение или уменьшение уровня жидкости может отслеживаться, начиная с определенного уровня, который определяется с помощью поплавкового датчика. Данный датчик может быть монтирован в нескольких местах.

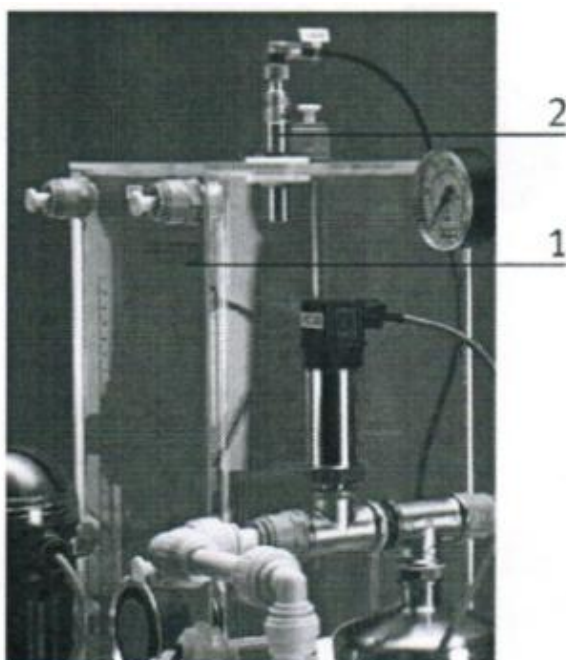


Рис 11. Верхний резервуар (1) и ультразвуковой уровнемер (2)

Насос P101 посылает жидкость из нижнего резервуара B101 в верхний резервуар B102(1) по трубопроводу. Жидкость, под действием силы тяжести перетекает из верхнего резервуара в нижний по другому пути. Расход на входе в верхний резервуар зависит от производительности насоса, на выходе от уровня столба жидкости. При изменении производительности насоса оба расхода через некоторое время становятся одинаковыми, и уровень устанавливается на каком-то определенном значении. Каждому конкретному значению производительности насоса соответствует свое конкретное значение уровня.

Уровень в верхнем резервуаре измеряется ультразвуковым датчиком В101(2). Так же имеется возможность измерять расход.



Рис 12. Расходомер В102 (2)

Наладка канала связи по средствам OPC-технологии.

Основой технологии OPC являются следующие понятия: сервер и клиент; интерфейс; интеграция.

Сервер - это программа или библиотека, исполняющаяся на персональном компьютере, к которому непосредственно подключена встроенная система. OPC-сервер могут использовать производители оборудования автоматизации или OEM (Original Equipment Manufacturer - поставщик комплексного оборудования).

Предполагается, что тот, кто создаёт, например, плату сбора данных, снабжает её не только драйвером, но и реализует OPC-сервер, работающий с этой платой через драйвер или даже напрямую. Тем самым OEM-производитель предоставляет стандартный доступ к своей плате.

Список возможных изготовителей OPC-серверов неограничен. OPC-сервером можно снабдить контроллер, плату ввода/вывода, адаптер полевой шины, программу пересчёта, генератор случайных чисел, что угодно, лишь бы это могло поставлять или принимать данные. Но всё-таки здесь речь идёт, в первую очередь, о программном обеспечении для более низкого уровня в системах автоматизации.

Клиент - это программа, которой нужен доступ к встроенной системе. Схема взаимодействия клиентов и серверов показана на рис.13. У каждого сервера может быть несколько клиентов, а клиент может обмениваться данными с несколькими серверами.

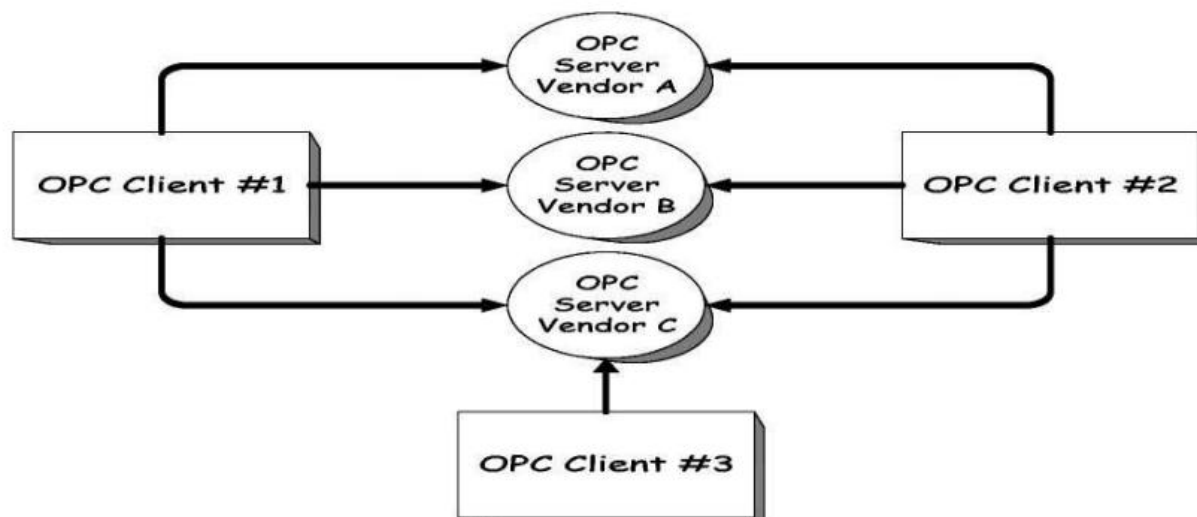


Рис. 13. Взаимодействие OPC-серверов и OPC-клиентов

OPC-сервер предоставляет данные, **OPC-клиент** потребляет. Этим задаётся вторая категория пользователей спецификаций OPC. И к ней относятся в первую очередь те, кто реализует программное обеспечение более высокого уровня. Например, поставщик SCADA-пакета. Или чего-то близкого по назначению. **Основной единицей данных в OPC является переменная (Item, ТЕГ, ТЭГ, переменная).**

Переменная может быть любого типа, допустимого в OLE: различные целые и вещественные типы, логический тип, строковый, дата, валюта, тип VARIANT и так далее. Кроме того, переменная может быть массивом.

Каждая переменная обладает свойствами. Различаются обязательные свойства, рекомендуемые и пользовательские.

Обязательными свойствами обязана обладать каждая переменная. Это, во-первых, текущее значение переменной, тип переменной и права доступа (чтение и/или запись). Во-вторых, очень важные свойства - качество переменной и метка времени. Технология OPC ориентирована на работу с оборудованием, а оборудование может давать сбой, так что корректное значение переменной не всегда известно OPC-серверу, о чём и уведомляется клиент через качество (хорошее/плохое/неопределённое и дополнительная информация). Метка времени сообщает о том, когда переменная получила данное значение и/или качество. Ещё одним обязательным свойством является частота опроса переменной OPC-сервером. Но не все OPC-серверы работают в режиме опроса оборудования. Поэтому существуют серверы, не реализующие это свойство. Последним из обязательных свойств является описание переменной. Это строковое значение, содержащее информацию для пользователя о том, зачем нужна эта переменная.

Дополнительные свойства являются необязательными для реализации в OPCсервере. Это, например, диапазон изменения (выход за границы диапазона должен специальным образом обрабатываться клиентом) и единица измерения. Есть перечень рекомендуемых свойств, но можно добавить и свои собственные.

Существует три основных способа получения OPC-клиентом данных от OPCсервера: синхронное чтение, асинхронное чтение и подписка. При синхронном чтении клиент посылает серверу запрос со списком интересующих его переменных и ждёт, когда сервер его выполнит. При асинхронном чтении клиент посылает серверу запрос, а сам продолжает работать. Когда сервер выполнил запрос, клиент получает уведомление. И, наконец, в случае подписки клиент передаёт серверу список интересующих его переменных, а сервер затем регулярно присылает клиенту информацию об изменившихся переменных из этого списка. Запись данных ничем не отличается от чтения, за исключением того, что нет записи по подписке.

Технология OPC содержит как обязательные к реализации интерфейсы, так и необязательные. В каждом интерфейсе также есть обязательные и необязательные функции. В соответствии с моделью COM набор функций каждого интерфейса фиксирован. Необязательность функций означает, что при вызове они возвращают код ошибки, говорящий о том, что функция не реализована. Например, каждая единица данных Data Access - сервера идентифицируются уникальным именем. В минимальном объеме реализации сервер не предоставляет информацию о названиях своих данных. В более полной реализации есть специальный интерфейс для получения списка имен и других свойств данных (тип, поддержка временных меток и т.п.). Имена могут быть произвольными, а могут иметь иерархическую структуру, например MD3846.DIN3, MB846.DOUT2, MA444.AIN8. Есть интерфейсы, позволяющие раскрывать эту структуру имен, переходить от одного к другому, но они необязательны. В минимальной реализации клиент может получить доступ к данным OPC-сервера, только если он знает названия данных из других источников. OPC-клиент предоставляет OPC-серверу интерфейс IAdviseSink. В минимальном объеме он способен принимать и передавать данные. Поддержание более высокоуровневых механизмов требует их реализации, как в сервере, так и в клиенте.

Могут быть реализованы такие возможности, как совместная работа нескольких клиентов, имеющих общие данные, уведомление о завершении работы сервера, получение дополнительной информации о клиентах и т.п.

Разработать OPC-приложения можно на основе любой среды программирования, позволяющей использовать технологию COM. В том числе можно исполь-

зовать офисные приложения, например MS Word, в которые встроен язык Basic for Application или VB Script. В этом случае происходит интеграция OPC-приложений в офисные приложения, что довольно удобно.

Например, можно получить последовательность данных от встроенной системы, показать ее в таблице Excel, произвести необходимые преобразования и построить график, пользуясь исключительно средствами Excel.

Большинство интерфейсов OPC продублировано для интегрированных приложений.

Поддержка интеграции в различные среды является необязательной. Использование технологии OPC предоставляет возможности совершенно нового уровня по сравнению с возможностями инструментальных средств, не придерживающихся стандартов такого уровня. Приложения могут быть написаны задолго до момента готовности встроенной системы, так как интерфейсы OPC определены и зафиксированы. Достоинством применения технологии OPC является также легкая и незаметная модернизация приложений, что является следствием применения технологии COM.

Серверы и клиенты могут находиться на разных рабочих станциях, как в пределах локальной сети, так и по всему миру, а для связи использовать как IPC (межпроцессорное взаимодействие, при необходимости прочитайте самостоятельно) в рамках одного компьютера, так и обмениваться через Интернет. Применение одного инструмента формирует стандартные или часто применяемые приемы работы с ним. Ряд клиентов может быть использован для работы с большим числом встроенных систем.

Технология OPC позволяет применять сетевую поддержку, интеграцию и другие сервисы, предоставляемые операционной системой. Технология OPC является мощным и универсальным средством, но это же и является причиной его сложности.

Универсальность не эквивалентна удобству, поддержка стандарта OPC увеличивает сложность программного обеспечения, для его разработки требуются более квалифицированные сотрудники. Основным недостатком этой технологии является то, что в силу своей универсальности она не может предоставить достаточно высокоуровневый интерфейс. Возможность работы с большим числом встроенных систем ведет к тому, что множество параметров обмена данными задается непосредственно в вызовах функций. Например, добавление обрабатываемой клиентом ячейки данных требует 12 параметров, а ее чтение - 7 параметров. Понятно, что для разных систем их значения будут различными, но

значительное число параметров придает громоздкость и осложняет применение этой технологии.

Появилось множество фирм-разработчиков программного обеспечения, специализирующихся только на разработке систем, поддерживающих технологию OPC. Однако, поддержание этой технологии в системах, при проектировании которых это изначально не было учтено, проблематично.

Существуют встроенные системы, в которых применение OPC неоправданно, так что технология OPC на текущем уровне развития пока не может позиционироваться как универсальный инструмент для организации взаимодействия встроенных систем и программного обеспечения на персональном компьютере.

Чаще всего OPC-технологии **не** используют для обмена данными в системах автоматизации низкого уровня, из-за низкой скорости обмена данными, и обычно используется в системах более высокого уровня. Но в данном случае OPC-сервер обеспечивает удовлетворительную скорость обмена данными, а его несомненные плюсы по универсальности работы с различным оборудованием значительно упрощают работу.

На рис. 14. представлена схема измерительной системы с использованием OPC-технологий. Аналогичная схема использована в описываемой далее работе для снятия данных измерительной аппаратуры (датчики расхода, давления, уровня, температуры, и т.д.) и записи на исполнительные механизмы (насос, клапаны, нагревательный элемент).

Все сигналы с датчиков и входы исполнительных механизмов подключены к ЦАП/АЦП. Он, в свою очередь подключен, к ПК через по последовательному интерфейсу RS-232. Для работы с этим устройством на ПК установлен OPC-сервер, к которому подключен OPC-клиент, управляющий всей установкой.

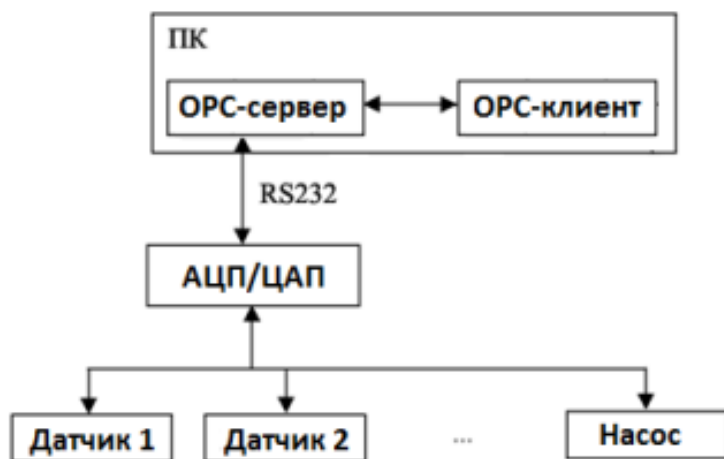


Рис. 14. Схема измерительной системы

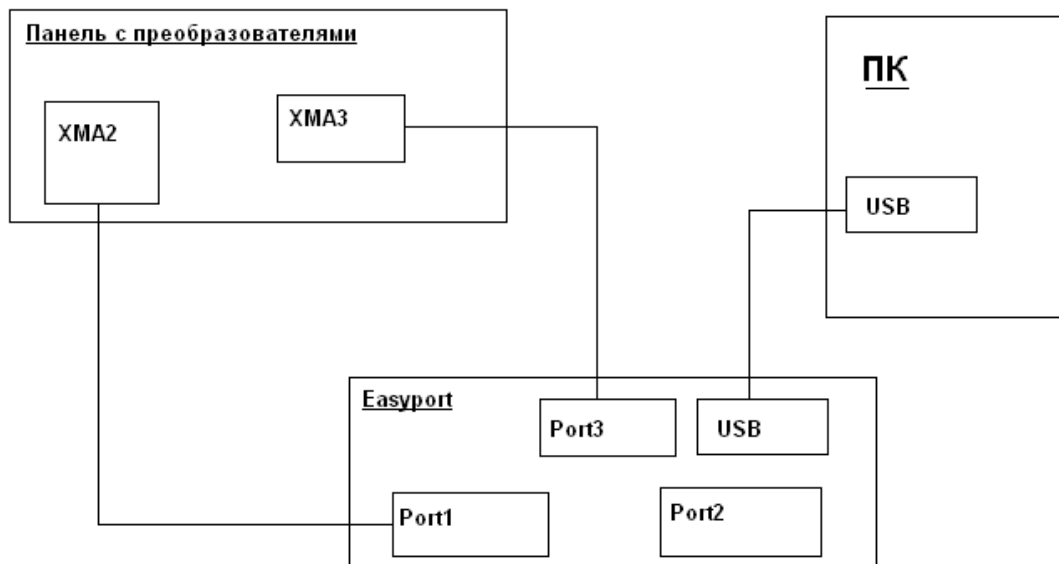


Рис. 14.б Та же схема подключений, только иначе прорисованная. У каких-то установок коробка Изипорт по USB, у каких-то по последовательному интерфейсу RS232.

Для работы с данной установкой компанией “FESTO” был разработан OPC-сервер FESTO Didactic EzOPC. Программа для управления лабораторным стендом, написанная при помощи программного продукта Matlab, является OPC-клиентом. Для работы реализован экран оператора в SCADA – системе.

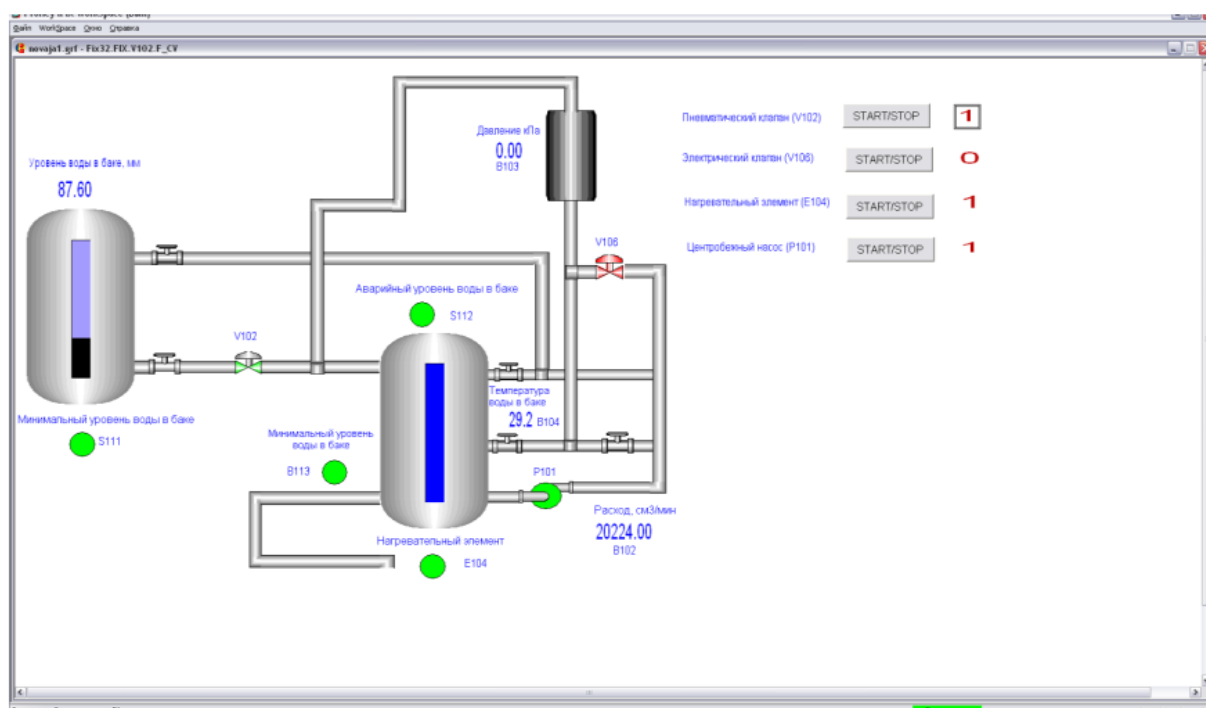


Рис. 15. Экран оператора в SCADA – системе

В исследуемой системе основная функция автоматизации - функция контроля расхода.

Используемая система – контролируемая система с саморегуляцией. Она не имеет задержки по времени. Контур контроля в комбинации с насосом создает легко управляемую систему.

Существуют два способа управления:

– контроль расхода посредством насоса P101 как контролируемая система.

Управляющее воздействие – напряжение насоса, который устанавливает расход перекачиваемой жидкости;

– контроль расхода посредством пропорционального клапана V106 как контролируемого элемента. Управляющее воздействие – напряжение на катушке клапана, которая устанавливает положение поршня клапана. Насос P101, при этом, качает с постоянной скоростью.

1. Выяснили используемые сигналы для данной установки.

Полный перечень используемых сигналов, их идентификаторов и назначений представлен в таблице 1.

Таблица 1.

Перечень входных и выходных сигналов модуля

Идентификатор сигнала	Идентификатор блока	Назначение
Дискретные сигналы		
Входные		
EasyPort1.Bits.IputPort1Bit00	–	Цифровой датчик расхода
EasyPort1.Bits.IputPort1Bit01	S111	Поплавковый датчик переполнения в нижнем резервуаре
EasyPort1.Bits.IputPort1Bit02	S112	Поплавковый пороговый датчик в верхнем резервуаре
EasyPort1.Bits.IputPort1Bit03	B113	Ёмкостной датчик уровня (нижний)
EasyPort1.Bits.IputPort1Bit04	B114	Ёмкостной датчик уровня (верхний)
EasyPort1.Bits.IputPort1Bit05	S116	Клапан закрыт
EasyPort1.Bits.IputPort1Bit06	S116	Клапан открыт
Выходные		
EasyPort1.Bits.OutputPort1Bit00	V102	Шаровой клапан с пневматическим приводом
EasyPort1.Bits.OutputPort1Bit01	E104	Нагревательный элемент с управлением от реле
EasyPort1.Bits.OutputPort1Bit02	–	Переключение режима работы насоса цифровой(0)/аналоговый(1)
EasyPort1.Bits.OutputPort1Bit03	P101	Центробежный насос (в цифровом режиме)
EasyPort1.Bits.OutputPort1Bit04	V106	Пропорциональный клапан с электронным управлением
Аналоговые сигналы		
Входные		
EasyPort1.AnalogIn0	B101	Ультразвуковой датчик уровня
EasyPort1.AnalogIn1	B102	Расходомер с частотным сигналом
EasyPort1.AnalogIn2	B103	Датчик давления электронный
EasyPort1.AnalogIn3	B104	Термометр сопротивления PT100
Выходные		
EasyPort1.AnalogOut0	–	Задание температуры – 0 – 10 В
EasyPort1.AnalogOut1	–	Задание расхода при аналоговом режиме работы насоса – 0 – 10 В

Настройка серверной части начиналась к подключению к серверу посредством vnc-viewer.

В качестве логина использовалось имя установки, в данном случае *opc04* (либо можно сделать это через ip-адрес самой установки), пароль: *manage*.

```
Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

E:\Documents and Settings\Admin>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection 3:

    Connection-specific DNS Suffix  . : localdomain
    IP Address. . . . . : 192.168.211.132
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.211.2

E:\Documents and Settings\Admin>ping 198.162.2.54

Pinging 198.162.2.54 with 32 bytes of data:

Reply from 198.162.2.54: bytes=32 time=1ms TTL=128
Reply from 198.162.2.54: bytes=32 time<1ms TTL=128
Reply from 198.162.2.54: bytes=32 time<1ms TTL=128
Reply from 198.162.2.54: bytes=32 time<1ms TTL=128

Ping statistics for 198.162.2.54:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

E:\Documents and Settings\Admin>
```

Рис. 13. Проверка связи клиента с сервером

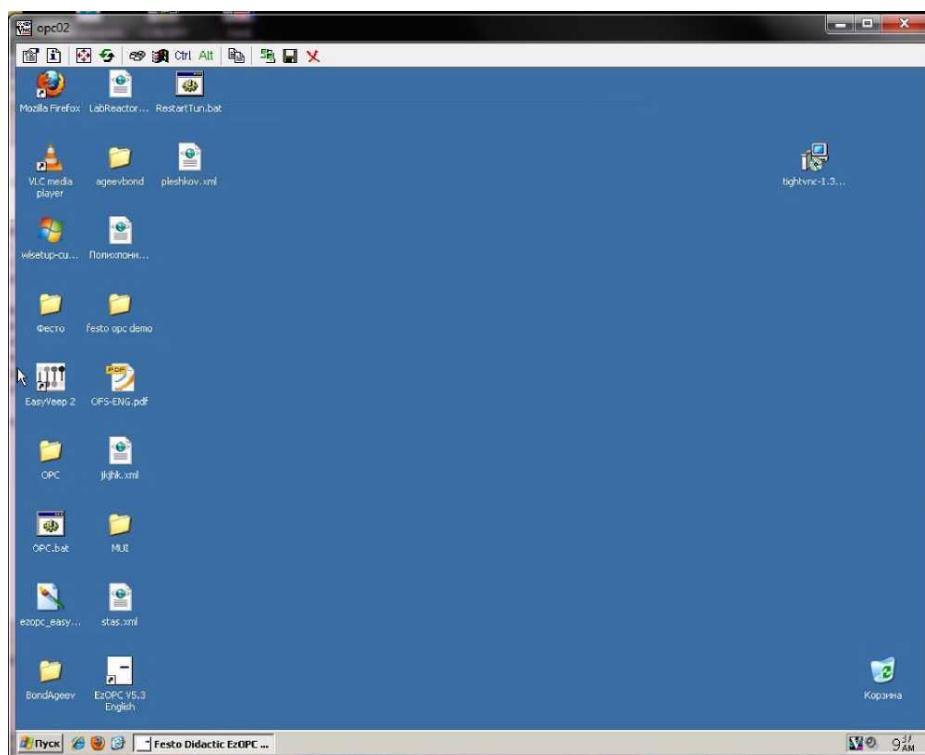


Рисунок 136 - Рабочий экран установки

Проверяли связь с easyport посредством утилиты EzOPC

Сигналы датчиков и исполнительных механизмов передаются серверу модулем EasyPort, подключаемым через интерфейс USB. Проверяем через Диспетчер устройств наличие подключения по USB или последовательному интерфейсу RS232.

3. Проверяем подключение по EasyPort. На данной вкладке настроек OPC-сервера возможно задать цепочку компонентов, между которыми будет организована связь. Также, здесь отображена общая информация о драйвере устройств, их текущем состоянии, а также информация о состоянии OPC-сервера.

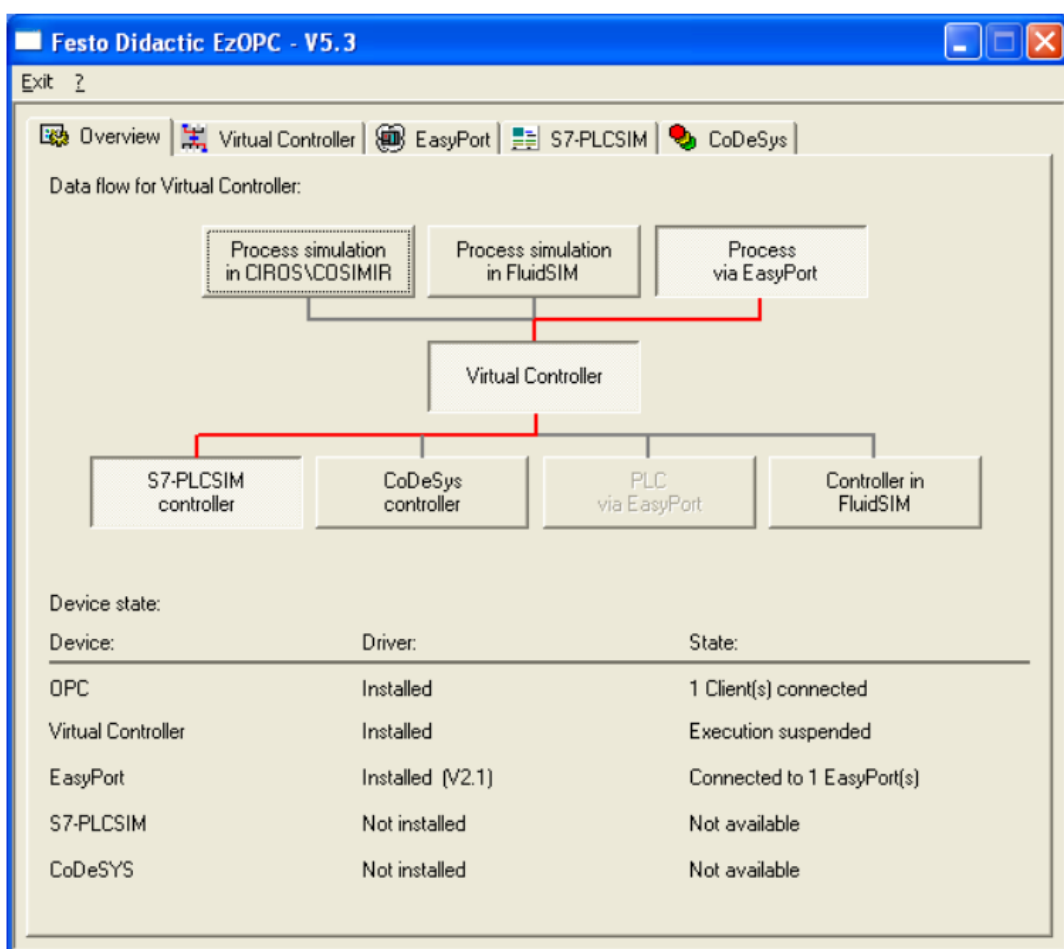


Рис. 15. Вид интерфейса вкладки EasyPort сервера EzOPC

На вкладке EasyPort предоставлена информация о модулях, подключенных и функционирующих на данный момент.

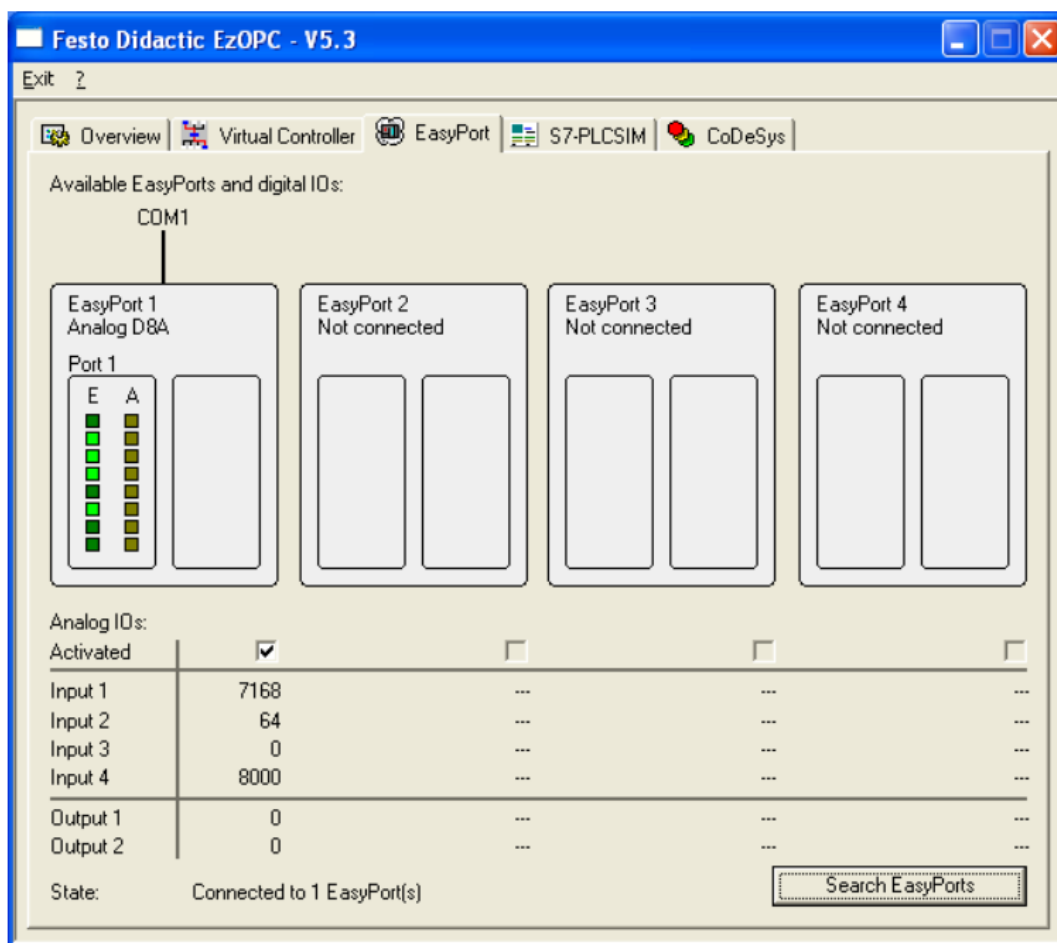


Рис. 16. Проверка связи по EasyPort

В случае отсутствия сигнала проверяли соединение на маршрутизаторе, рабочей станции и самом easyport.

4. Установка и настройка серверной части OPC Tunneller осуществляется на сервере сбора данных, который также является OPC-сервером. Утилитой Server-Side Gateway Configuration Tool (далее SSC) предусмотрена настройка таких функций коммуникационного сервера как шифрование данных, параметров персонализации, контроля доступа, а также параметров, непосредственно отвечающих за связь между объектами.

Производим настройку параметров персонализации и параметров контроля доступа. В Matrikon OPC Tunneller предусмотрены параметры, позволяющие ограничить доступ к OPC-серверу, предоставляя его определенному кругу удаленных пользователей. При этом, клиенты могут быть идентифицированы по таким свойствам как IP-адрес, Имя домена\Имя пользователя, Host Name, или произвольной комбинации этих свойств. Каждому сконфигурированному уда-

ленному пользователю назначается список OPC-серверов, к которым он может получить доступ.

Если кратко, то: Encryption отвечает за шифрование, в чем нет необходимости при выполнении лабораторных работ. Impersonation определяет виды доступа подключающимся устройствам. Access lists - определяет устройства который могут подключиться по IP адресу.

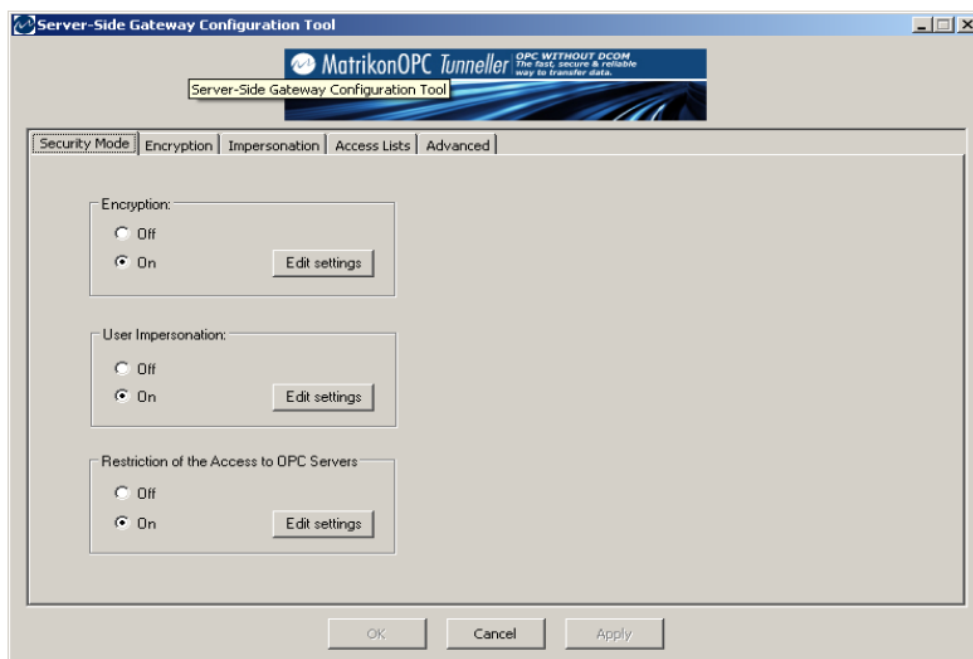


Рис. 17. Настройка параметров доступа

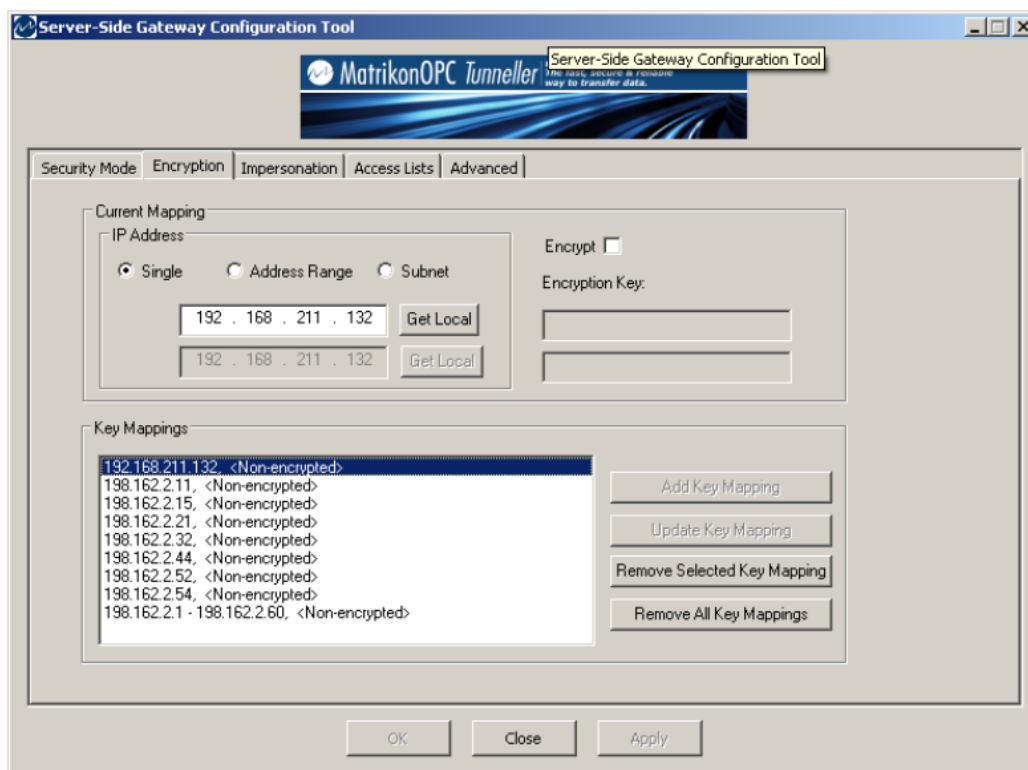


Рис. 18. Настройка параметров шифрования SSC

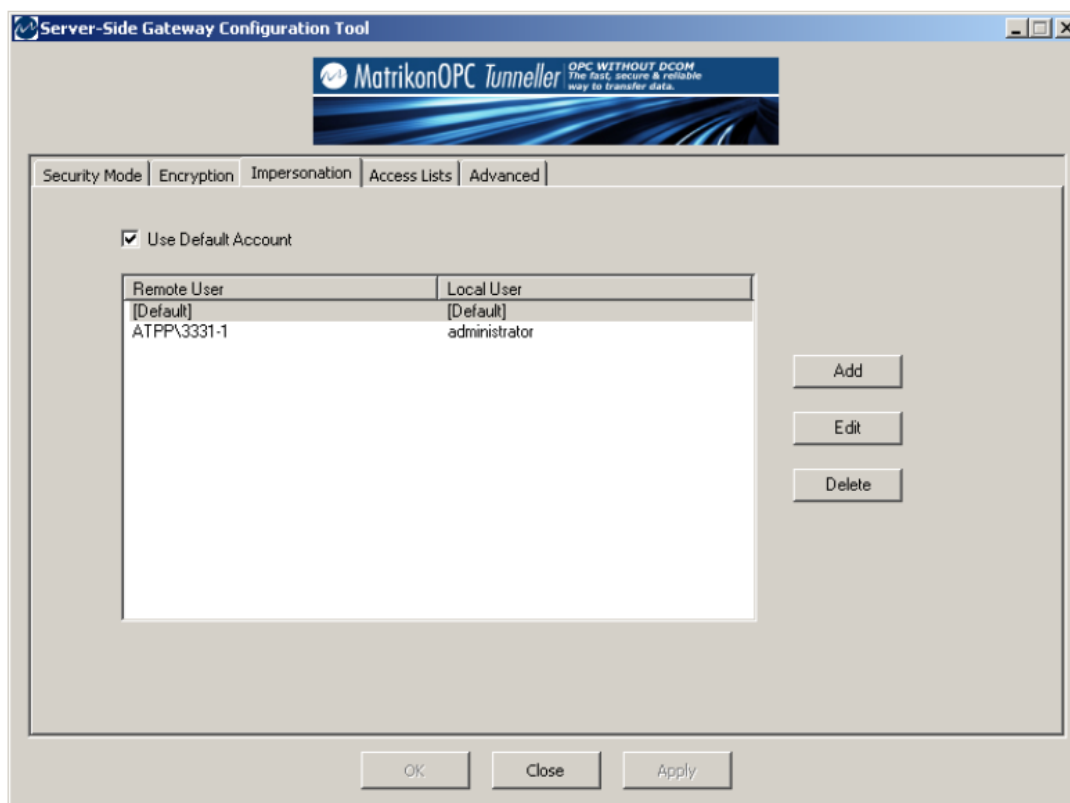


Рис. 19. Настройка параметров персонализации

Для возможности удаленно подключиться к рабочей станции посредством клиента в Access lists прописывают ip адреса самого компьютера и виртуальной машины, если она присутствует и используется. Следует отметить, что также в Access lists проверяется сервер OPC, для версий свыше 5 используют Matrikon.OPC.Simulation.1, при версии ниже четвертой появляется Matrikon.OPC.Simulation.2.

Ethernet adapter Подключение по локальной сети 3:

DNE-суффикс подключения :atpplab
 1Ри4-адрес.....: 198.162.2.21
 Маска подсети :255.255.240.0
 Основной шлюз : 198.162.2.1

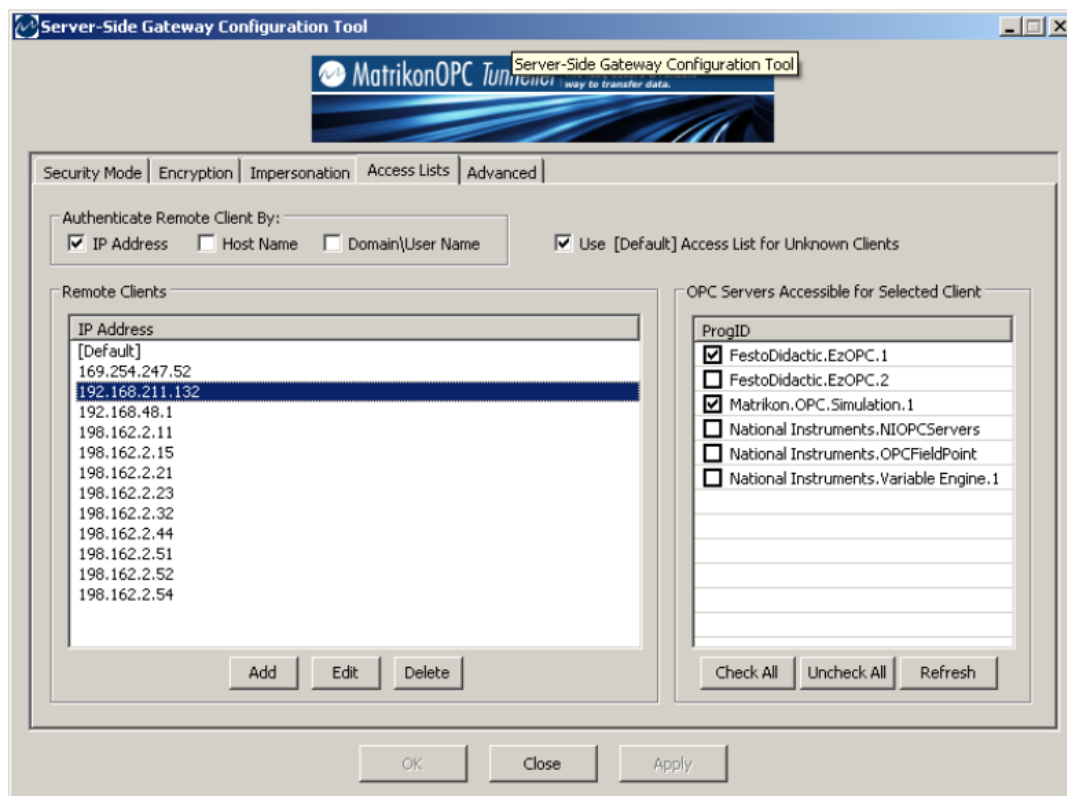


Рис.20. Настройка параметров доступа

Дополнительные параметры могут быть настроены на вкладке Advanced меню SSC.

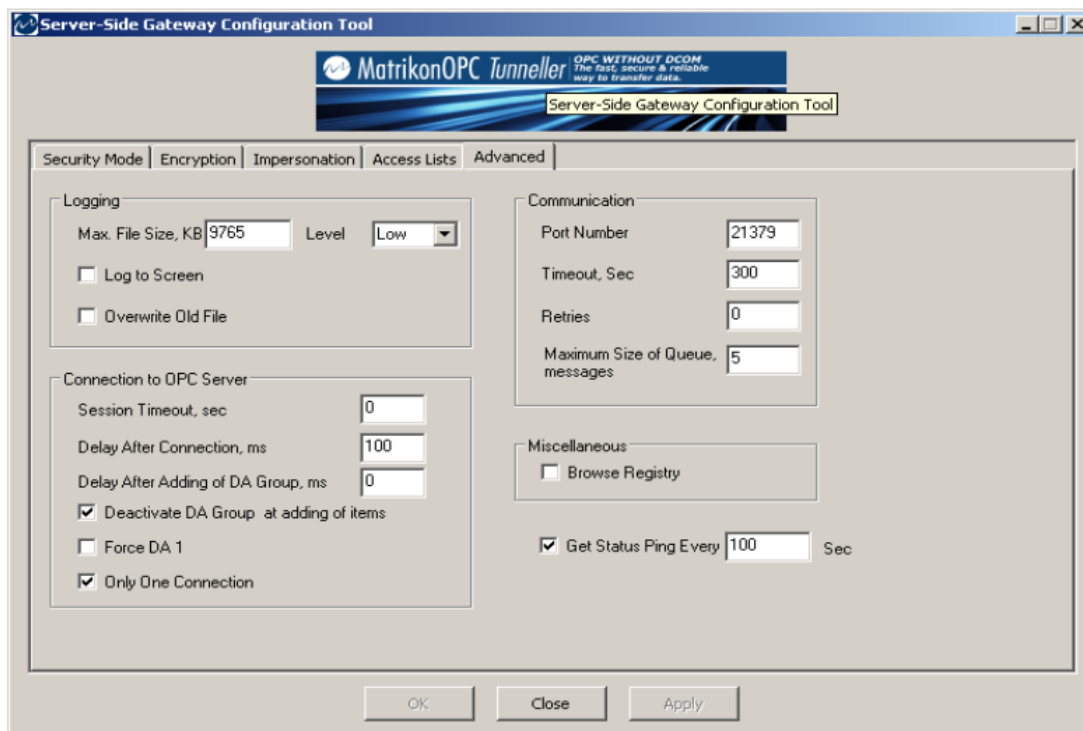


Рис. 21. Меню настройки дополнительных параметров

После настройки серверной части можно переходить к настройке OPC со стороны Клиента или уже на этом этапе проверить связь Сервера с OPC туннелем. Для этого используется тестировщик/просмотрщик. Далее представлена краткая информация:

После внесения ip адреса переходили в MatriconOPC Explorer, где из библиотеки вытаскивали все необходимые аналоговые и дискретные входа-выхода для выбранного порта №1. Проверка соединения устанавливается посредством статуса «Good».

Item ID	Access Path	Value	Quality	Timestamp	Status
EasyPort1.AnalogIn0		10639	Good, non-specific	02/13/2018 10:51:26.198...	Active
EasyPort1.AnalogIn1		0	Good, non-specific	02/13/2018 10:46:29.697...	Active
EasyPort1.AnalogIn2		0	Good, non-specific	02/13/2018 10:46:29.697...	Active
EasyPort1.AnalogIn3		1	Good, non-specific	02/13/2018 10:51:26.197...	Active
EasyPort1.AnalogOut0		9750	Good, non-specific	02/13/2018 10:47:16.063...	Active
EasyPort1.AnalogOut1		0	Good, non-specific	02/13/2018 10:22:46.383...	Active
EasyPort1.Bits.InputPort1Br00		False	Good, non-specific	02/13/2018 10:22:46.383...	Active
EasyPort1.Bits.InputPort1Br01		True	Good, non-specific	02/13/2018 10:22:46.383...	Active
EasyPort1.Bits.InputPort1Br02		True	Good, non-specific	02/13/2018 10:22:46.383...	Active
EasyPort1.Bits.InputPort1Br03		False	Good, non-specific	02/13/2018 10:22:46.383...	Active
EasyPort1.Bits.InputPort1Br04		False	Good, non-specific	02/13/2018 10:22:46.383...	Active
EasyPort1.Bits.InputPort1Br05		False	Good, non-specific	02/13/2018 10:22:46.383...	Active
EasyPort1.Bits.InputPort1Br06		False	Good, non-specific	02/13/2018 10:22:46.383...	Active
EasyPort1.Bits.InputPort1Br07		True	Good, non-specific	02/13/2018 10:22:46.383...	Active
EasyPort1.Bits.OutputPort1Br00		True	Good, non-specific	02/13/2018 10:43:39.854...	Active
EasyPort1.Bits.OutputPort1Br01		True	Good, non-specific	02/13/2018 10:49:33.717...	Active
EasyPort1.Bits.OutputPort1Br02		False	Good, non-specific	02/13/2018 10:49:33.717...	Active
EasyPort1.Bits.OutputPort1Br03		True	Good, non-specific	02/13/2018 10:44:47.571...	Active
EasyPort1.Bits.OutputPort1Br04		False	Good, non-specific	02/13/2018 10:22:46.383...	Active
EasyPort1.Bits.OutputPort1Br05		False	Good, non-specific	02/13/2018 10:22:46.383...	Active
EasyPort1.Bits.OutputPort1Br06		False	Good, non-specific	02/13/2018 10:22:46.383...	Active
EasyPort1.Bits.OutputPort1Br07		False	Good, non-specific	02/13/2018 10:43:39.854...	Active

Рисунок 2 - Выбранные тэги

Уже на этом шаге можно проверять, какие биты за что отвечают непосредственно на самой установке. Например, *Дискретные выходы: 00 – насос; 01 – разлив; 02 – конвейер. Аналоговый выход: Out0 - задание уставки (объем воды, мл. $V=15000\sim5$ мВ/В)*

5. Далее речь идет о настройке Клиентской части. Если коротко: Настройка клиентской части осуществляли через виртуальную машину, при этом последовательность такая же, как и в настройке серверной части, только для работы использовали один ip-адрес установки орс04 - 192.168.2.44 (т.е. указывали для Клиента только адрес Серверной части).

Более подробно: Реализация клиентской части программного обеспечения производится на удаленном компьютере, с которым необходимо установление соединения и обмен данными.

При организации нового подключения конфигурируются параметры связи, которые будут использоваться для данного конкретного клиента. Идентификация сервера, к которому производится подключение, может осуществляться по имени компьютера или IP-адресу.

Номер порта TCP должен соответствовать параметру, настраиваемому в SSC. Таймаут обработки (Processing Timeout) определяет промежуток времени, в течение которого ожидается получение ответа на запросы CSC. При отсутствии ответа или истечении времени ожидания происходит генерирование сообщения об ошибке. Таймаут связи определяет промежуток времени, в течение которого сетевой механизм отправляет запросы и ожидает получения ответа от SSC. По истечении данного времени и отсутствии отклика SSC связь считается разорванной.

Создадим новое подключение.

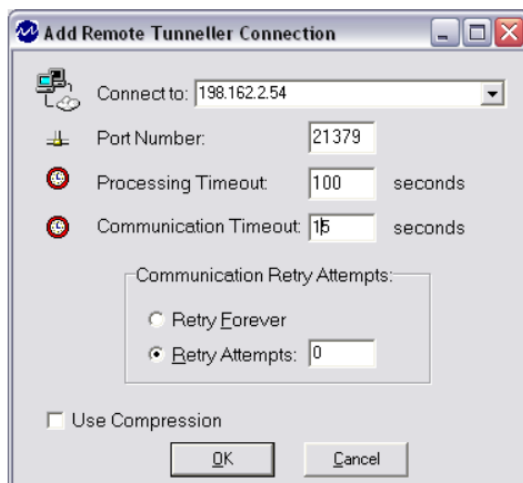


Рис. 22. Создание нового подключения

При работе канала связи в режиме шифрования данных для каждого клиента CSC должны быть настроены ключи безопасности, соответствующие конфигурации серверного компонента. Для корректного функционирования должен быть указан IP-адрес сервера, и задан ключ шифрования, соответствующий ключу, присвоенному в настройках серверного компонента.

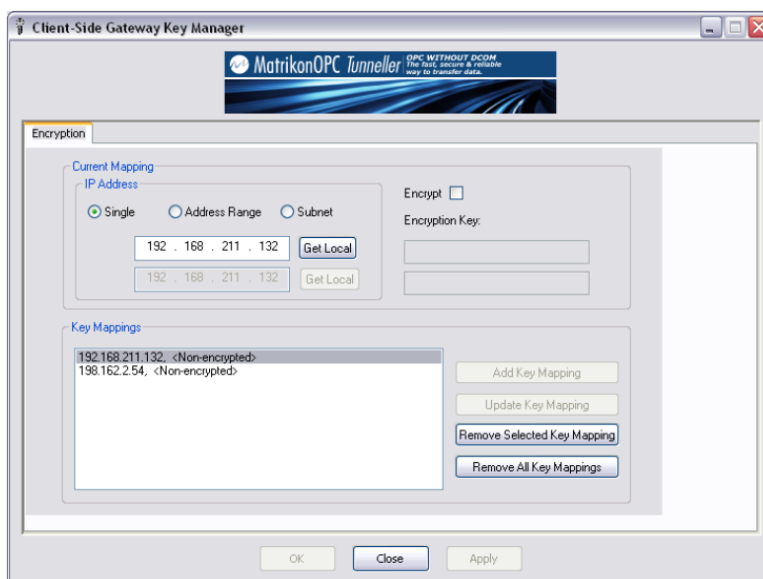


Рис. 23. Настройки параметров шифрования клиента OPC Tunneller

В меню клиентского компонента OPC Tunneller отображается список OPC-серверов компьютера, к которому произведено подключение. Идентификаторы серверов (ProgID) при обработке в OPC Tunneller представляются в виде «Tunneller:Имя компьютера:Стандартный ID сервера».

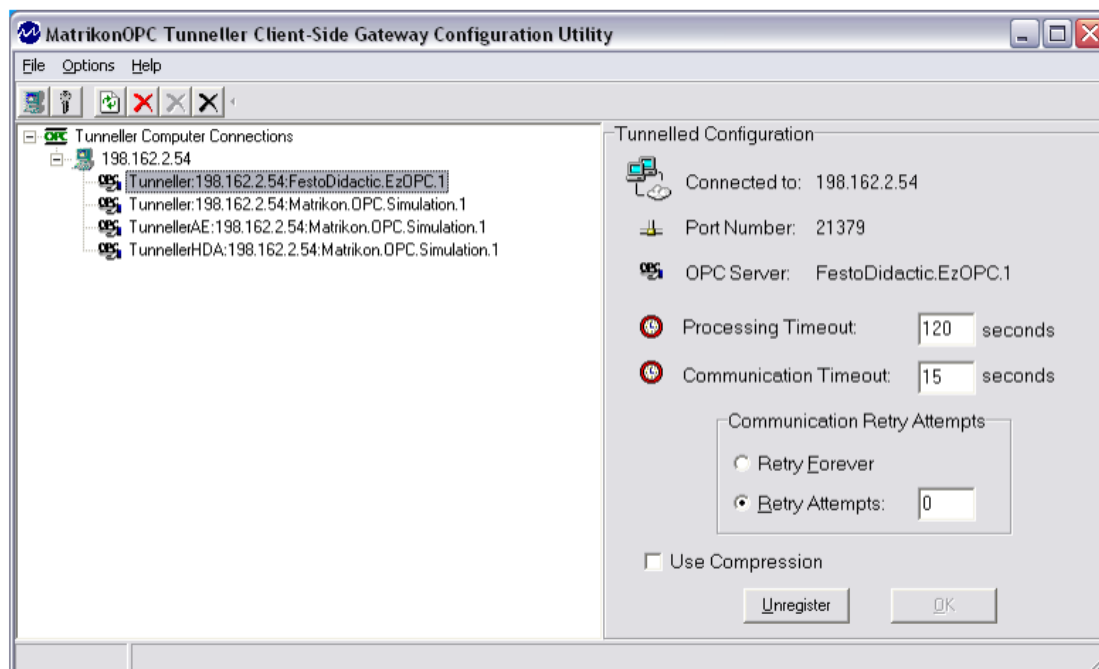


Рис. 24. Вид интерфейса CSC

Проверку связи, установленной с туннелированным сервером можно осуществить с помощью утилиты OPC Explorer. При подключении к серверу в сводке EzOPC отобразится информация о наличии связи с клиентом.

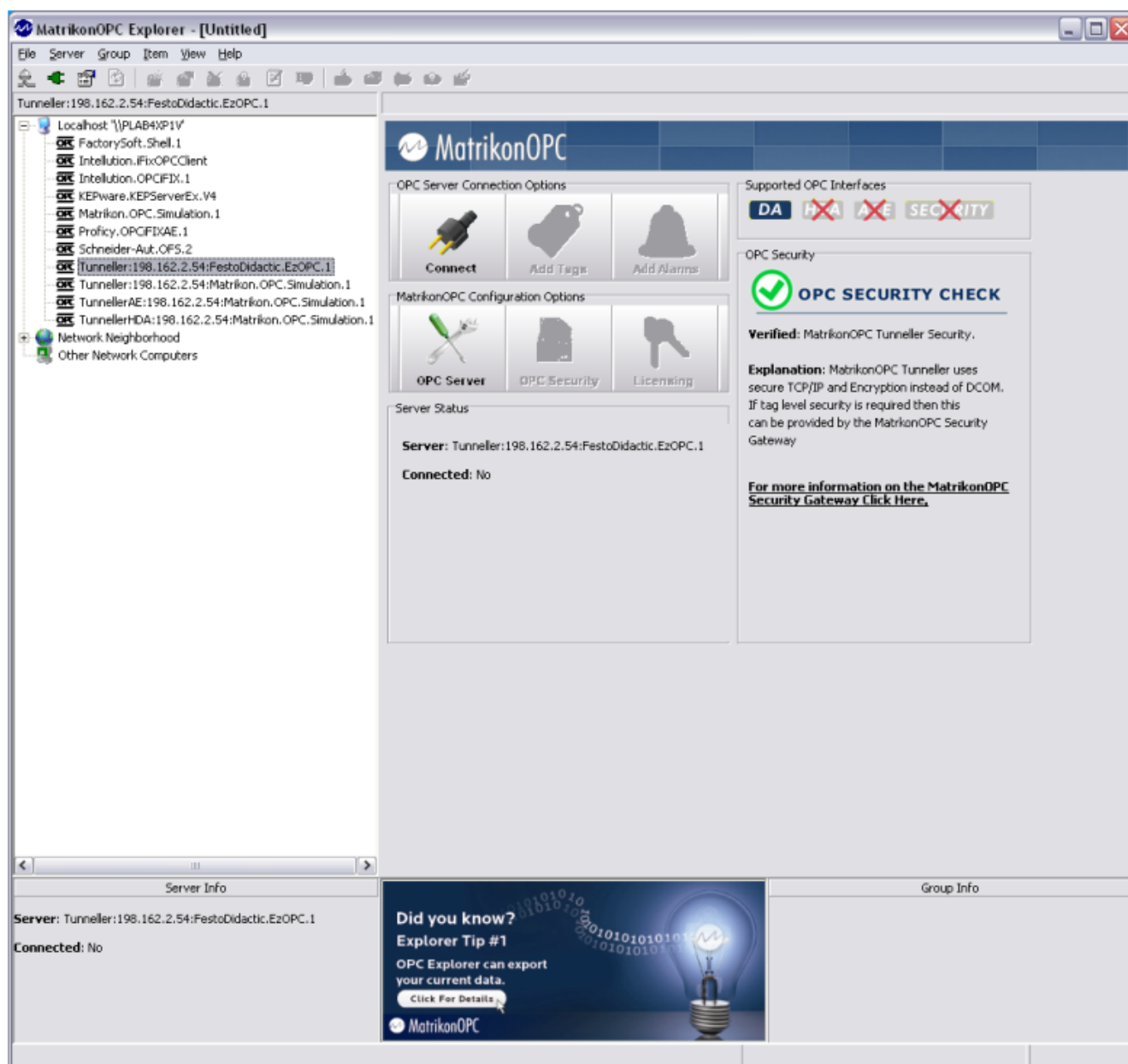


Рис. 25. Список доступных локальных OPC-серверов

В MatrikonOPC Explorer произведем подключение к OPC-серверу с COM ID FestoDidactic.EzOPC.1. После подключения появляется возможность создать группу и добавить в нее элементы данных.

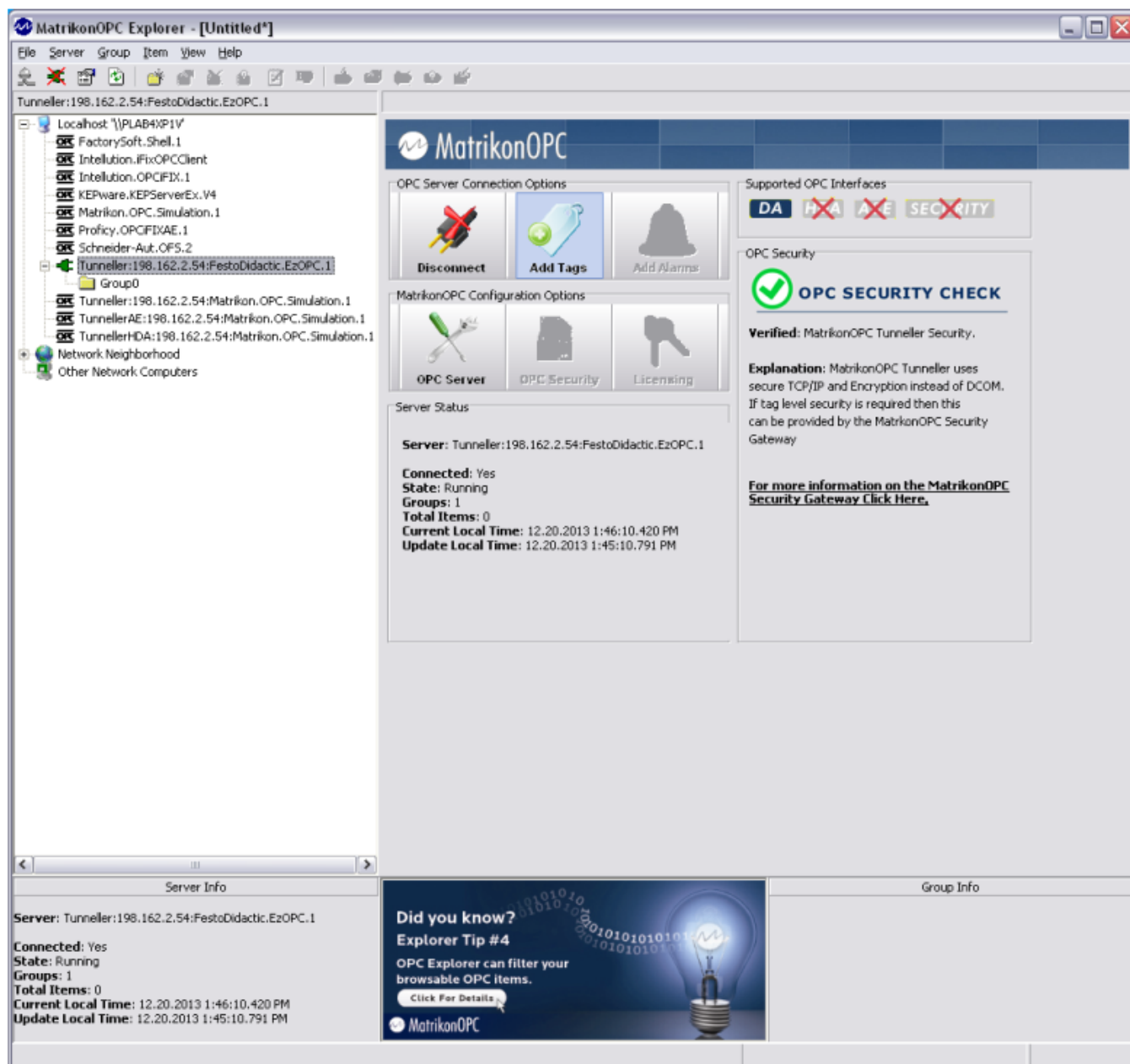
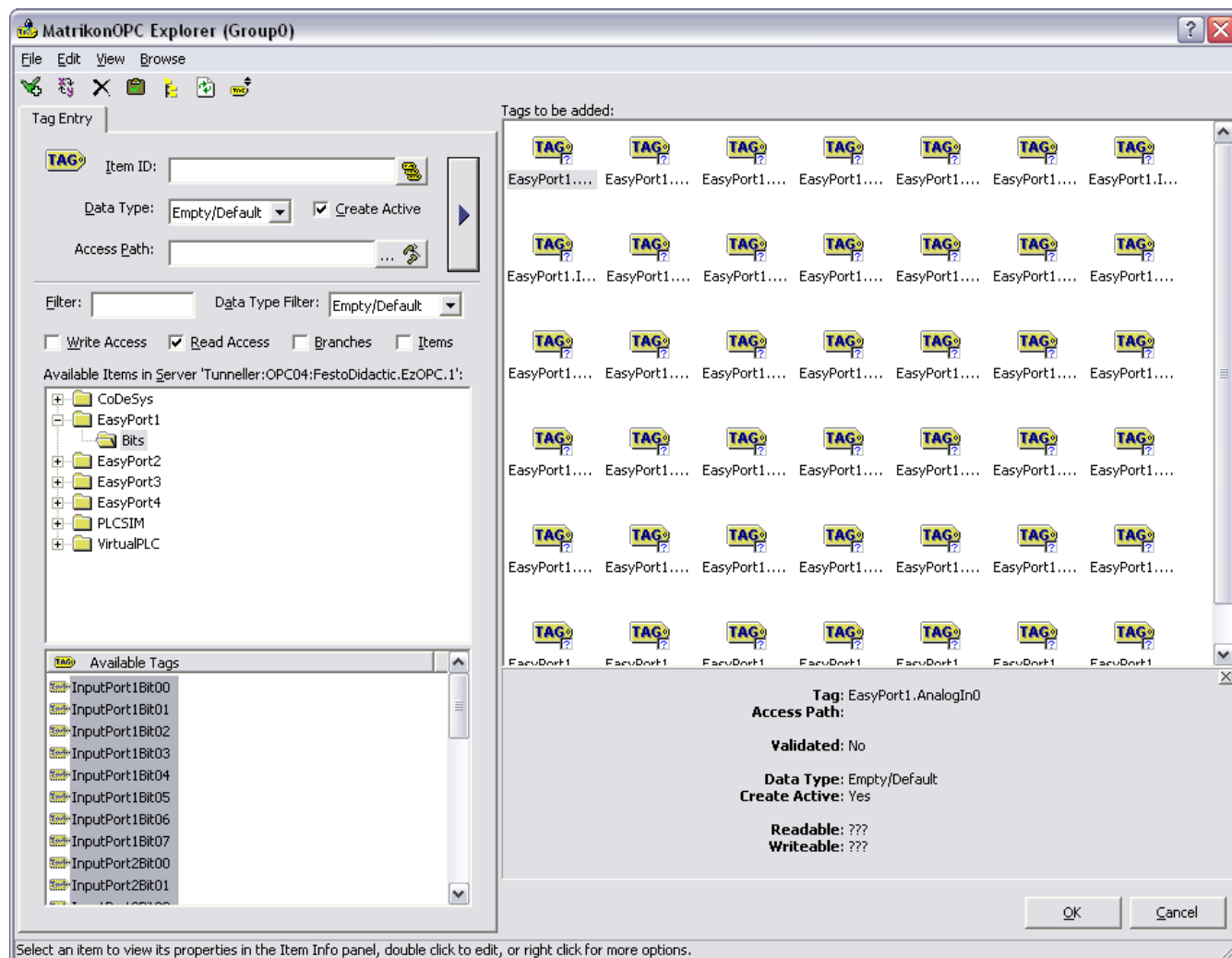


Рис. 26. Добавление элементов в группу

После подключения необходимо выяснить, каким битам принадлежат соответствующие дискретные входы/выходы, а также выяснить аналоговые входы/выходы. Для этого выбираем соответствующий подключению порт EasyPort1. Добавляем дискретные и аналоговые входы/выходы:



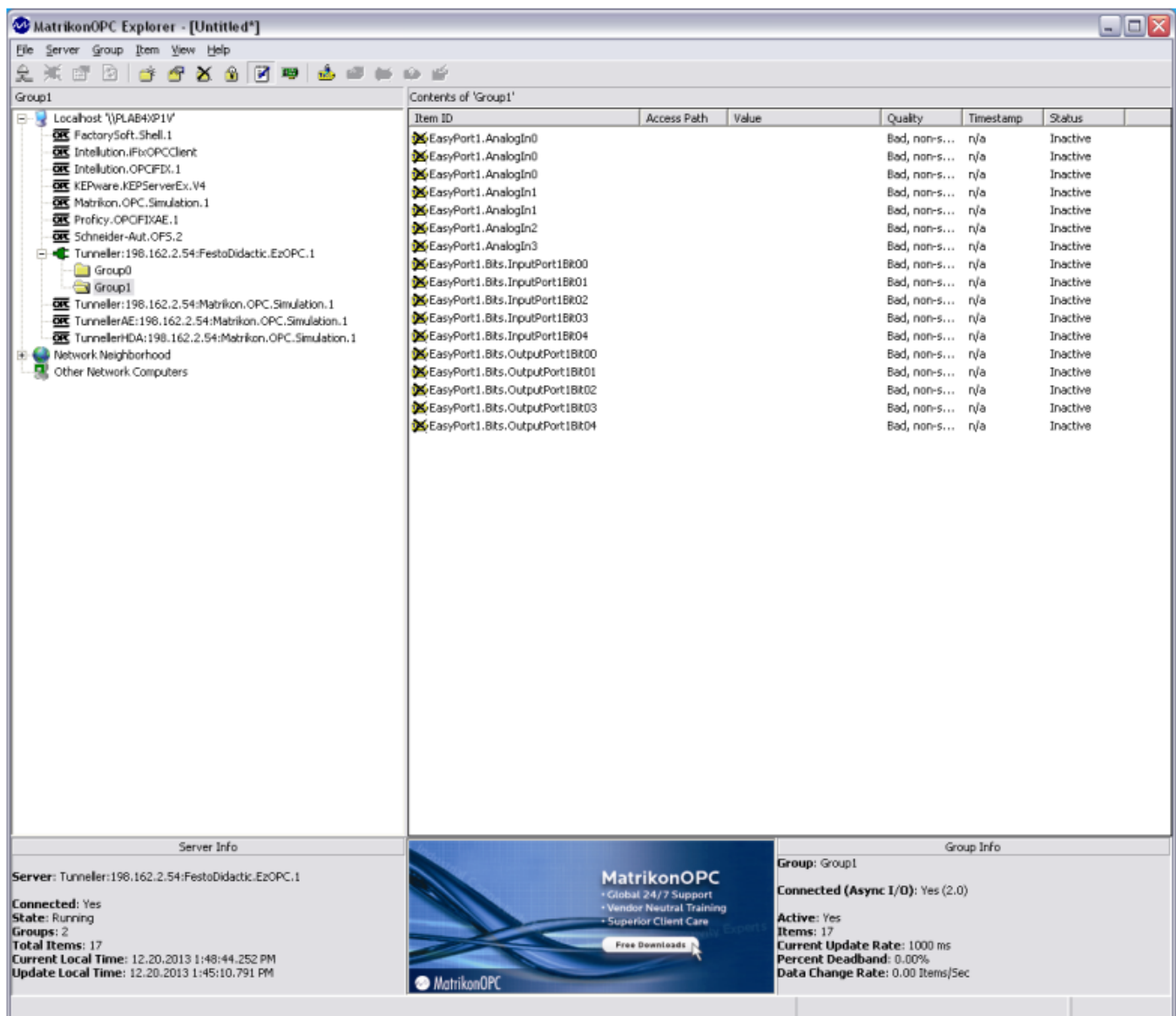


Рис. 27. Элементы группы Group1

Помимо идентификатора элемента отображаются параметры метки качества и времени, текущее значение элемента данных в зависимости от типа, а также статус активности. Активируем элементы группы Group1.

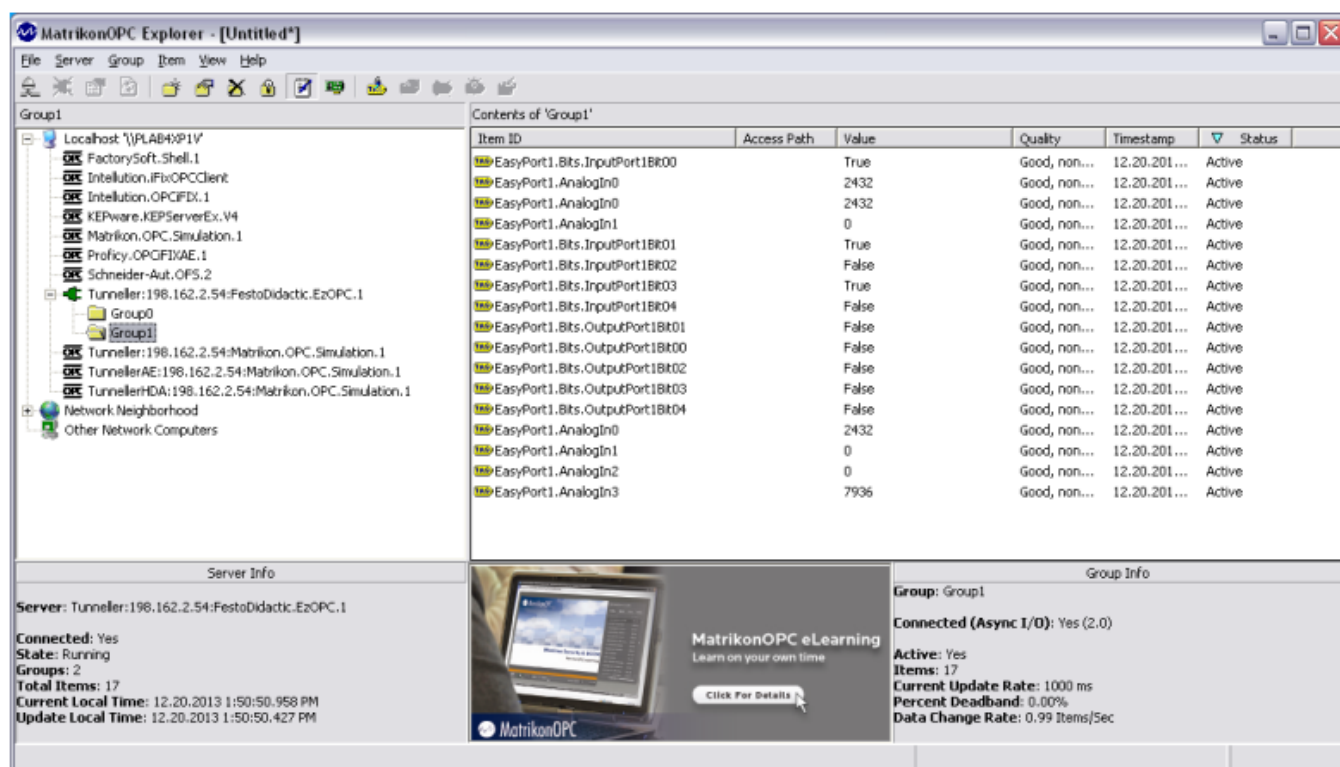


Рис. 28. Элементы группы Group1

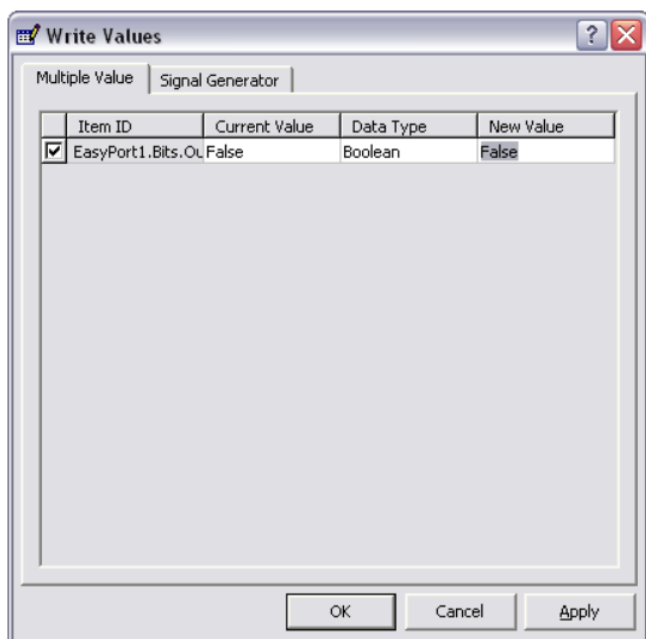


Рис. 29. Проверка связи с технологическим оборудованием

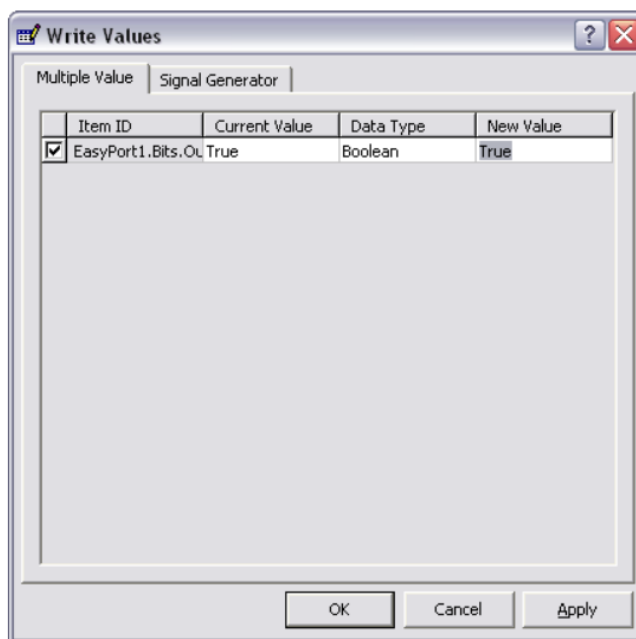


Рис. 30. Проверка связи с технологическим оборудованием

6. Работа в программной среде Matlab. Для того, чтобы получить возможность использовать в работе значения параметров с технологического оборудования, необходимо произвести конфигурацию связи между сервером и средой Matlab, которая в данном случае выступает в роли клиента. Устанавливаем связь с установкой. Добавляем блок OPC Config Real-Time.

OPC Config
Real-Time
OPC Configuration

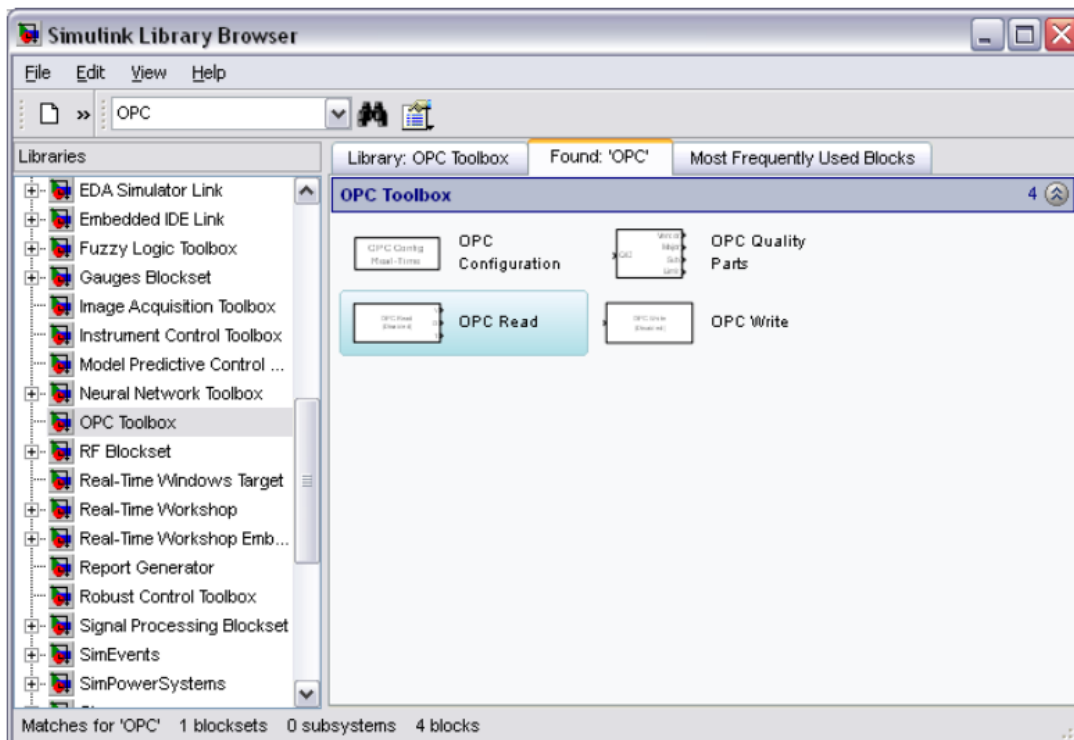


Рис. 31. Блоки в среде Matlab для установки связи с технологическим оборудованием

Настраиваем подключение OPC-клиента к серверу:

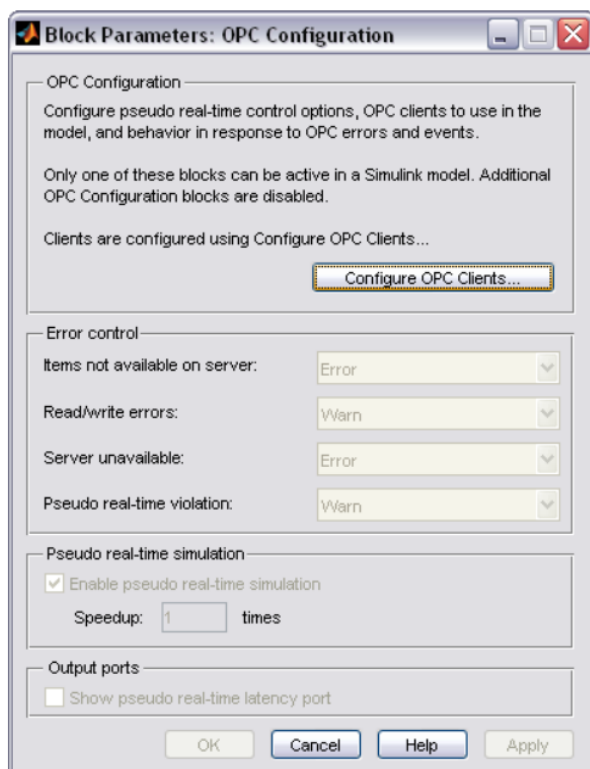


Рис. 32. Подключение OPC-клиента Matlab к серверу

Выбираем OPC-сервер.

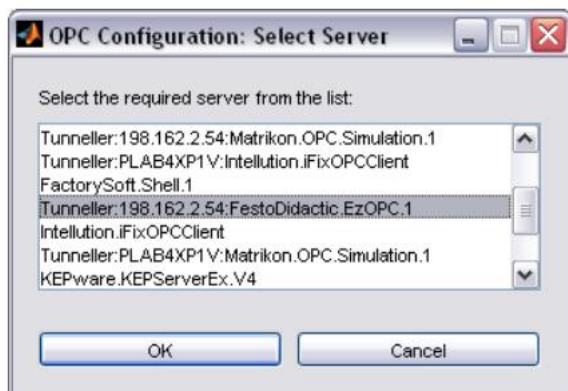


Рис. 33. Выбор сервера

Настраиваем параметры сервера.

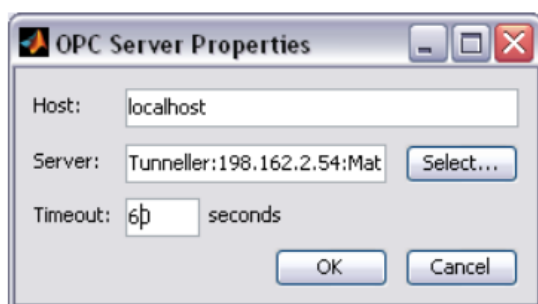


Рис. 34. Настройка параметров сервера

Подключаем OPC-клиента Matlab к серверу.

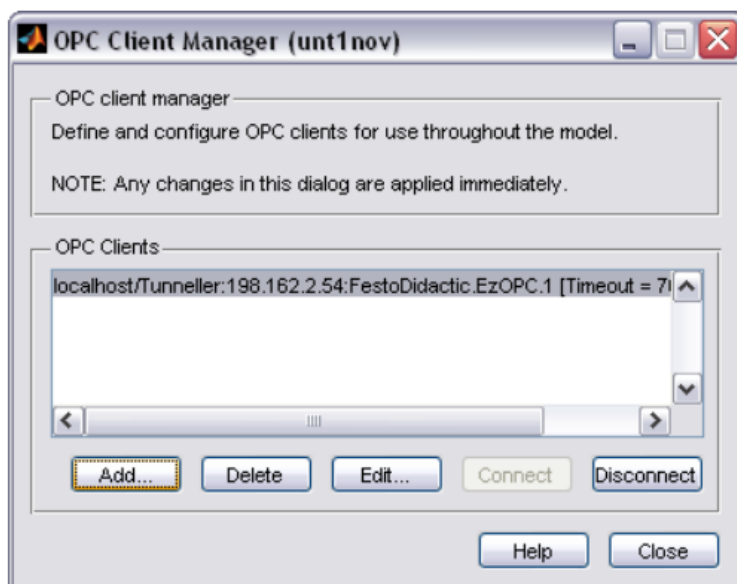
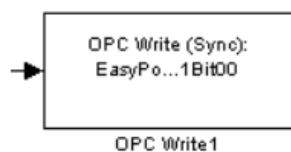


Рис. 35. Подключение OPC-клиента Matlab к серверу

7. Добавляем блоки чтения/записи дискретных и аналогового выходов



Добавляем тег, который необходимо записывать в данный блок.

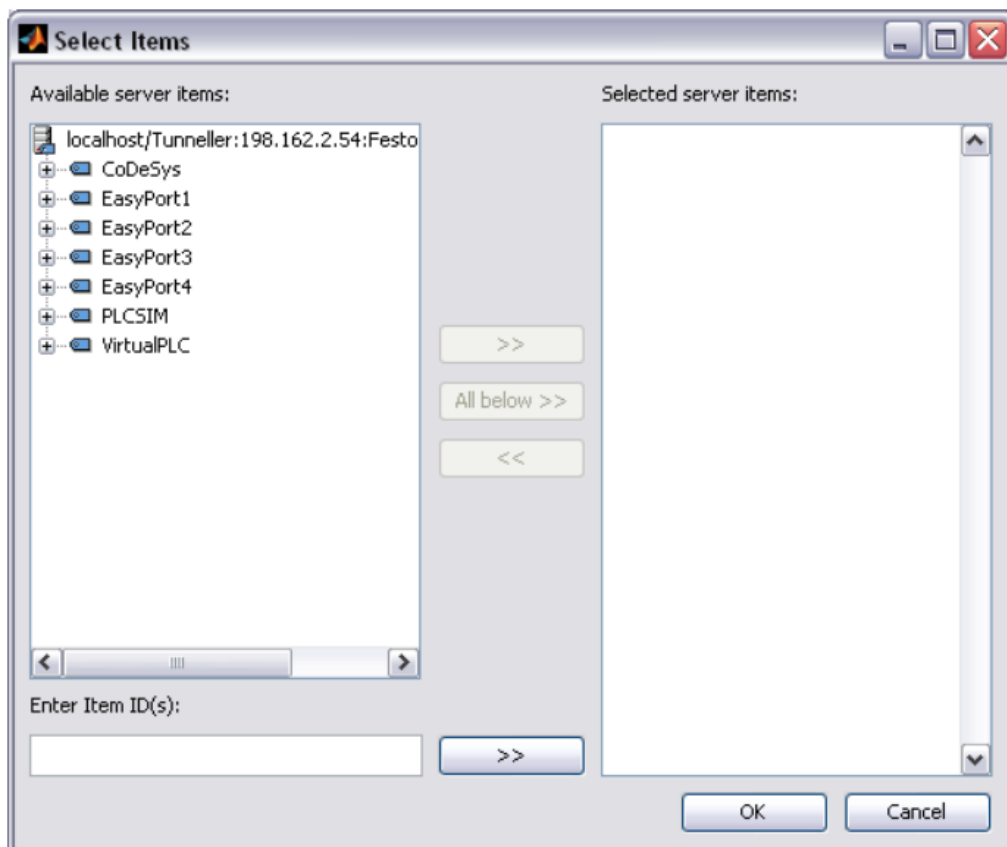


Рис. 36. Добавление необходимых элементов данных

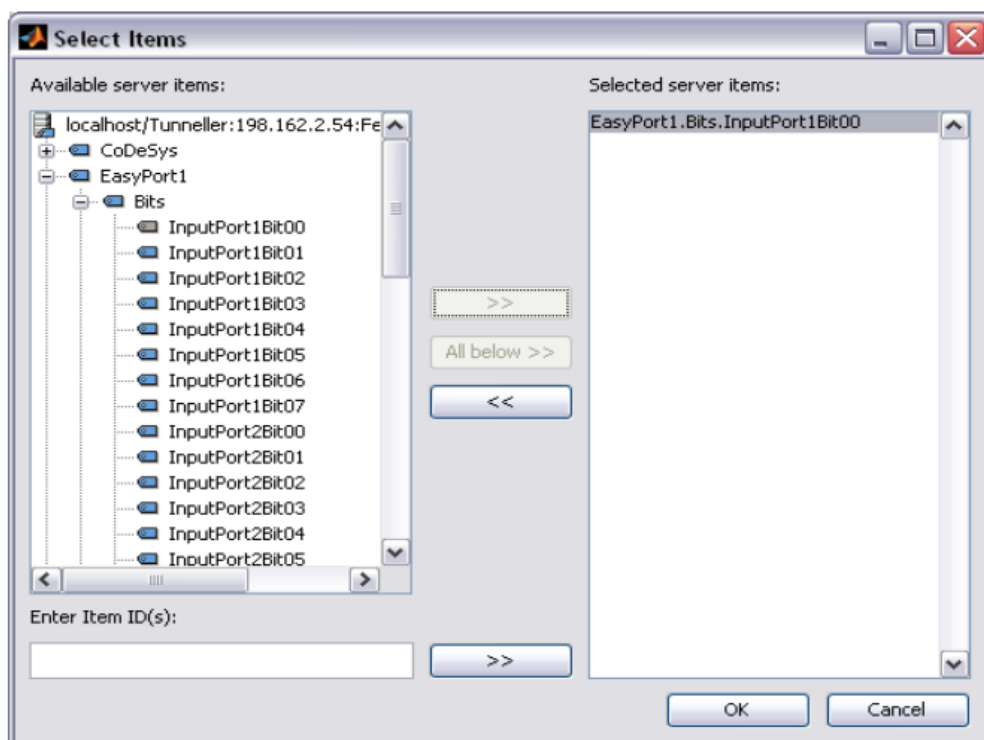


Рис. 37. Добавление необходимых элементов данных

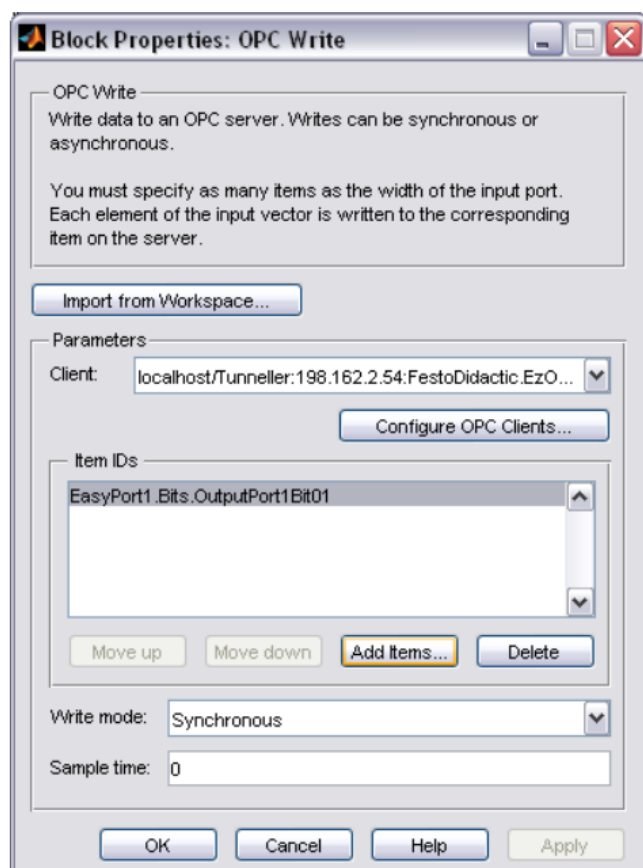
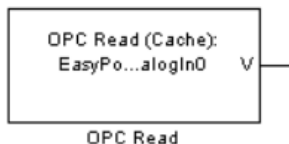


Рис. 38. Добавление необходимых элементов данных

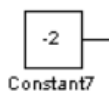
Для считывания информации добавляем следующий блок.



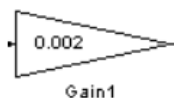
Добавление тега производим по методике, описанной выше.

8. Для регулирования уровня собираем схему в программной среде MATLAB, включающую в себя ПИД-регулятор, который требует настройки, то есть подбираются необходимые коэффициенты пропорциональной, интегральной и дифференциальной частей.

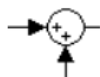
В схеме используются следующие блоки:



Блок Constant используется для задания различных величин-констант.



Блок Gain используется для умножения на вводимую величину, в нашем случае для перевода физической величины (расхода, уровня) в кодовый сигнал и обратно.



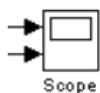
Блок Sum предназначен для суммирования (вычитания) сигналов/величин.



Блок Mux комбинирует свои входы в единственную векторную продукцию. Вход может быть векторный сигнал или скаляр. Все входы должны иметь тот же самый тип данных и числовой тип.

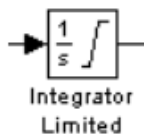


Блок Display предназначен для вывода значения сигнала/параметра.



Блок Scope предназначен для построения графика.

Для схемы ПИД-регулятора используются следующие блоки:

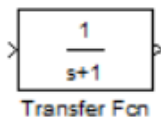


Блок Integrator Limited выводит интеграл своего ввода в шаге текущего времени.

Вывод блока ограничен верхним и нижним пределами.



Блок Saturation используется как фильтр по заданному диапазону.



Блок Transfer Fcn используется для моделирования передаточной функции.



Блок вычисления производной Derivative выполняет численное дифференцирование входного сигнала.

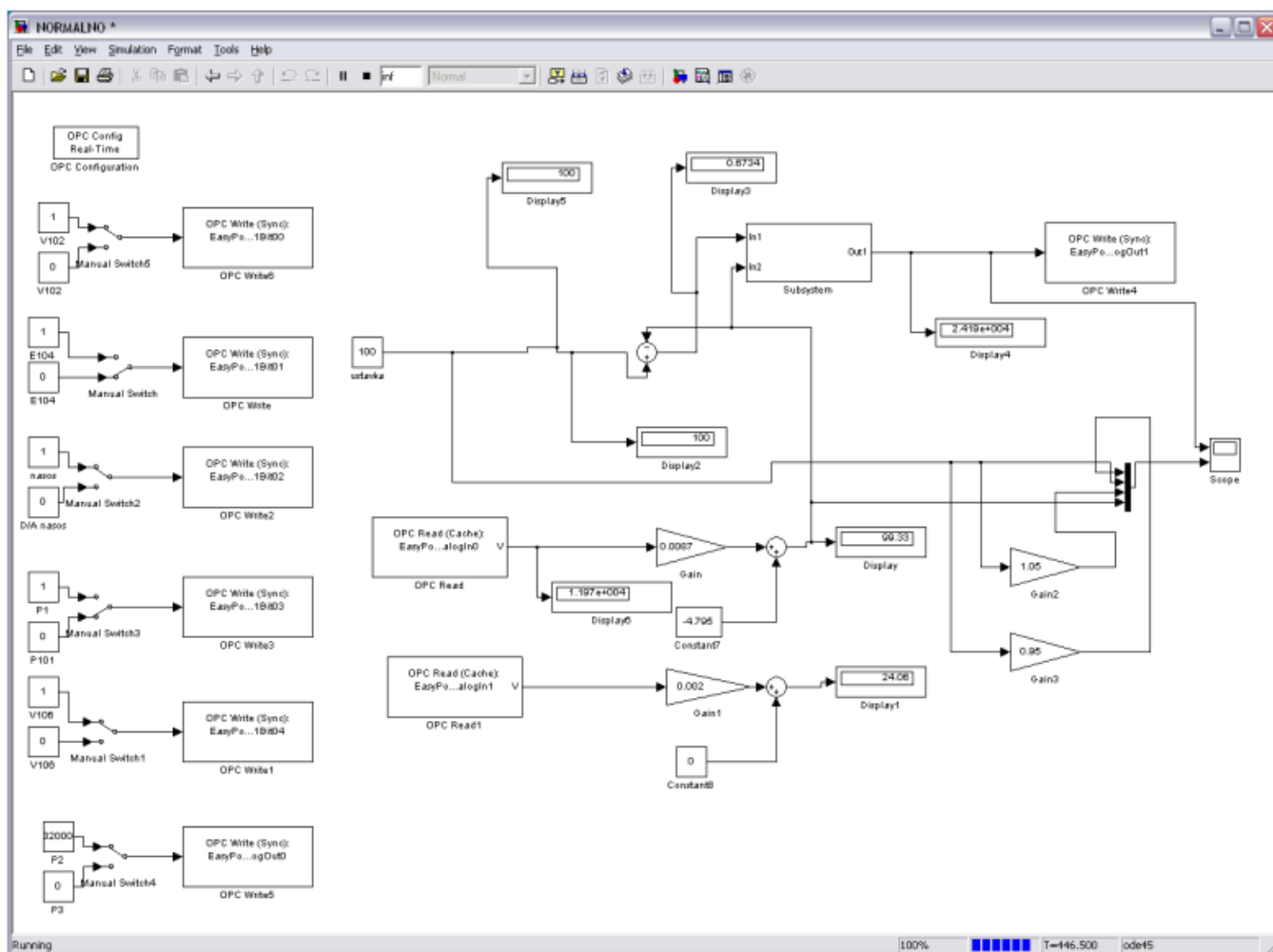


Рис. 39. Схема для регулирования уровня

В данной работе будем использовать ПИД-регулятор для регулирования уровня. ПИД-регулятор – самый сбалансированный из всех регуляторов, построенных на основе типовых звеньев. Пропорциональная составляющая вырабатывает сигнал, который противодействует отклонению регулируемой величины в данный момент времени. Интегральная составляющая накапливает результирующее значение, нивелируя, таким образом, недостаток П-регулятора – наличие статической ошибки. ПИД-регулятор обладает Д-составляющей, которая как бы прогнозирует отклонение от задания и следит за скоростью отклонения, поэтому является самой быстрой в данном алгоритме. По сути, это является преимуществом и недостатком одновременно.

Передаточная функция ПИД-регулятора имеет вид:

$$W_p(p) = K_p \left(1 + \frac{1}{T_i p} + T_d p \right)$$

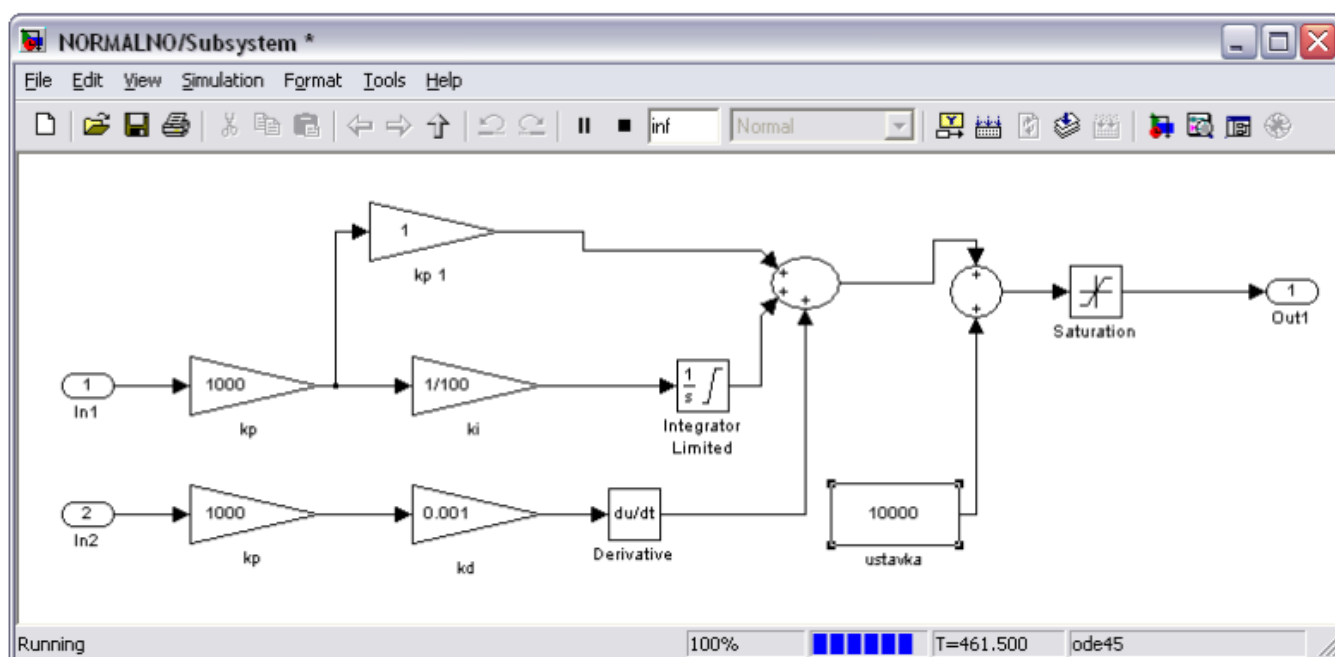


Рис. 40. ПИД-регулятор

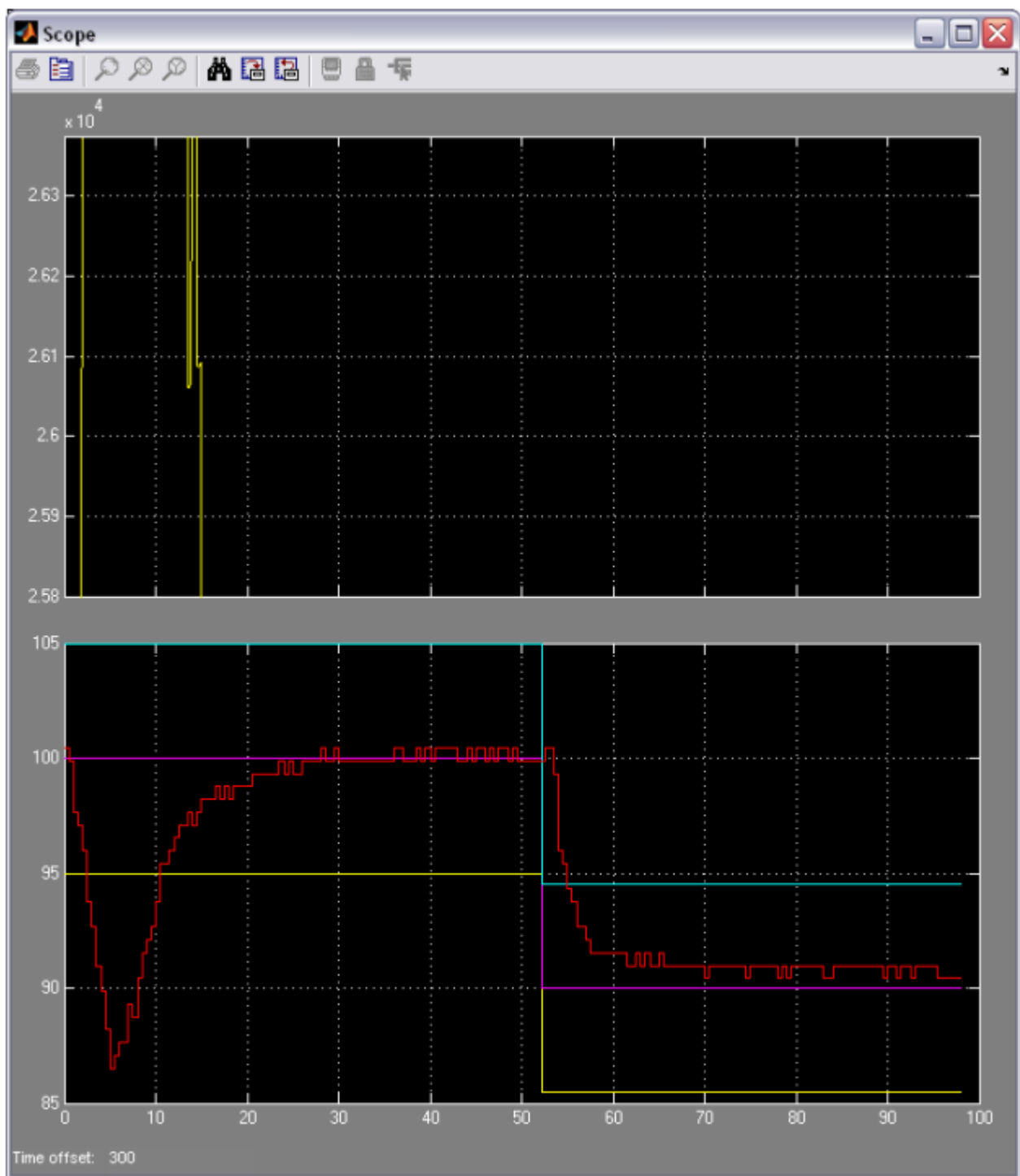


Рис. 41. График переходного процесса

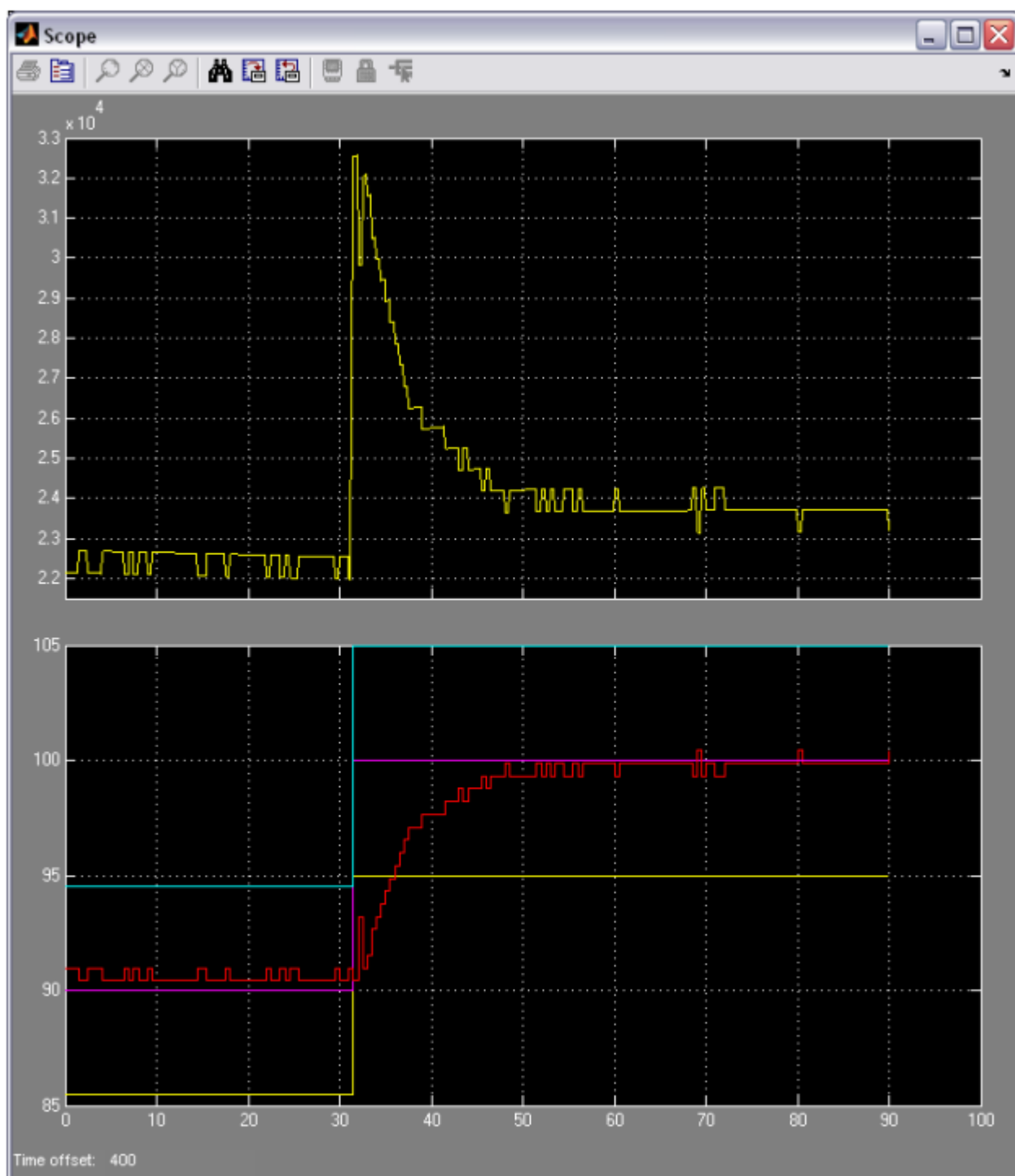
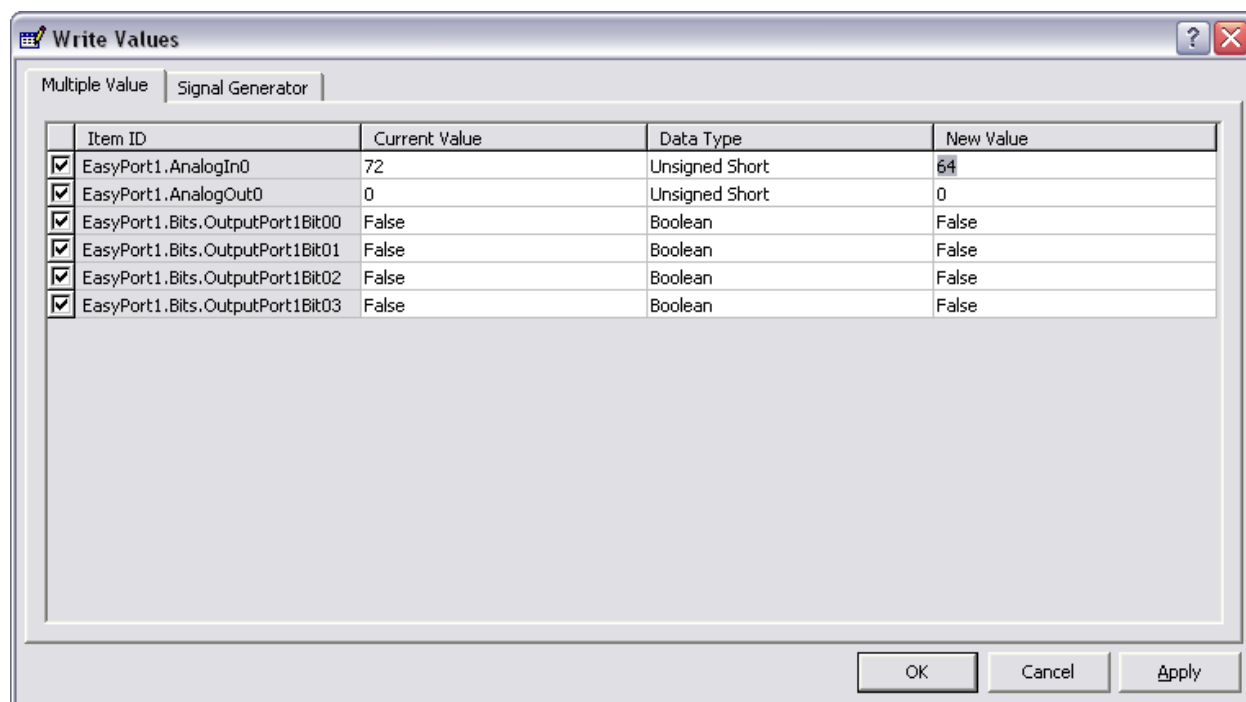


Рис. 42. График переходного процесса

Возможно вам (обращение к обучающимся, которые делали лабы по ТИП) более знакома система с двумя баками и разливкой в бутылочки от Фесто, где основные ТЕГи для управления процессом как непрерывным, так и периодическим:



AnalogIn0 – аналоговый вход, отображает уровень жидкости (в нашем случае, воды) в коде (0...32766)

AnalogOut0 – аналоговый выход, управляющий мощностью насоса путем подачи на него соответствующего напряжения.

OutputPort1Bit00 – дискретный выход, который включает/выключает центробежный насос.

OutputPort1Bit01 – дискретный выход, открывающий/закрывающий дозирующий клапан.

OutputPort1Bit02 - дискретный выход, запускающий/останавливающий конвейерную ленту.

OutputPort1Bit03 – дискретный выход, меняющий положение «пальцев» сепаратора.

Схема в MATLAB

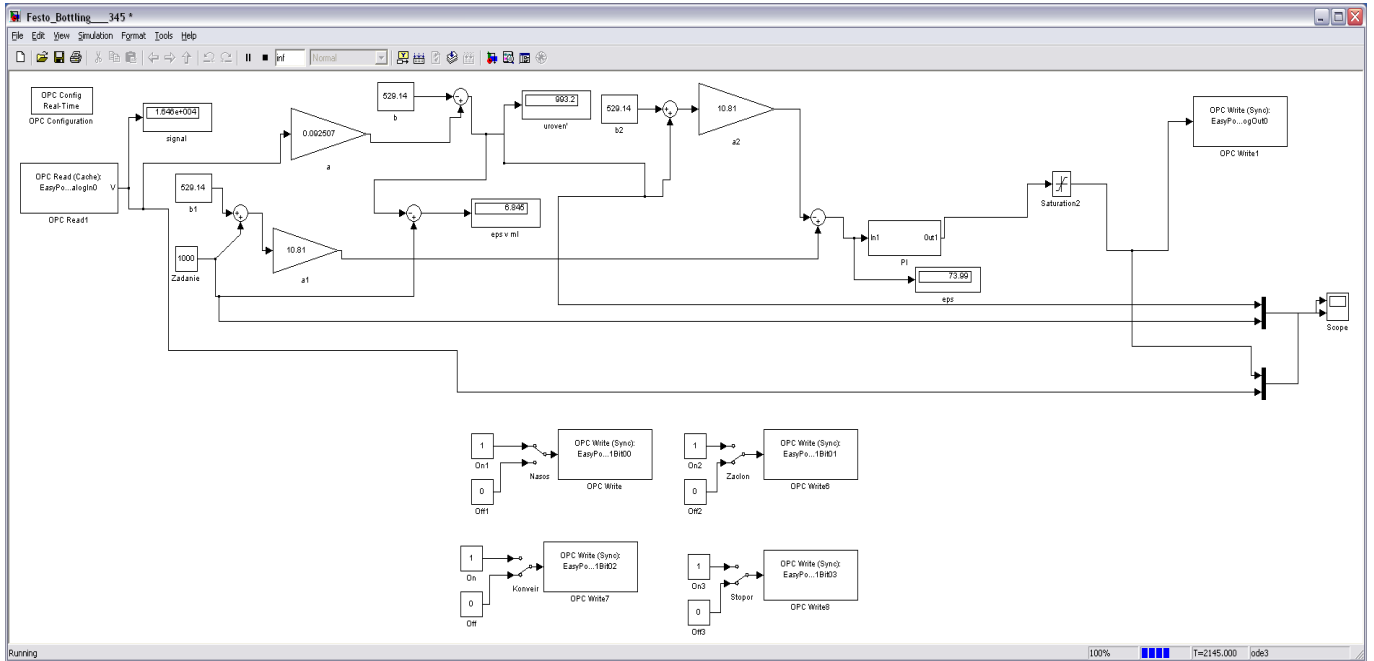
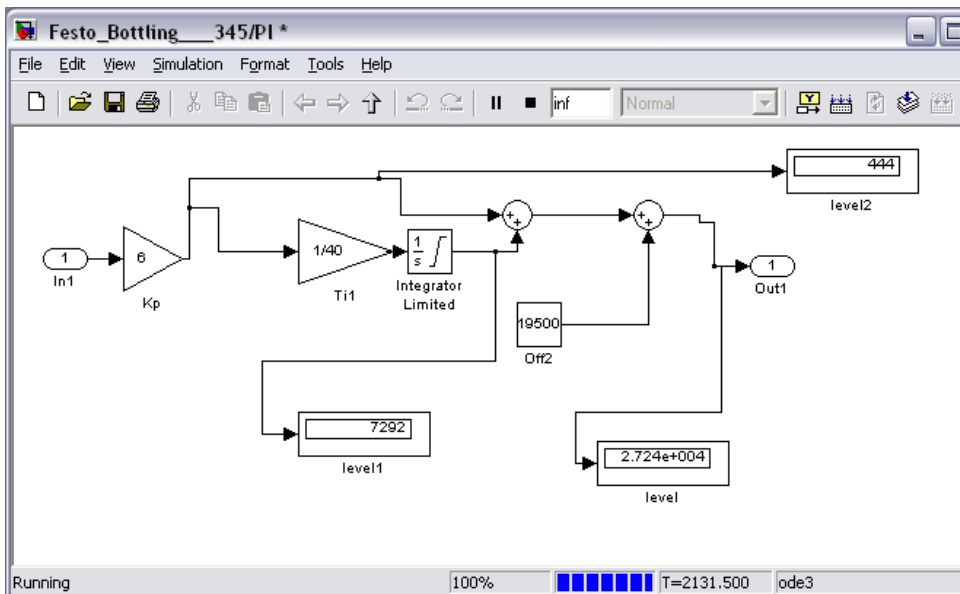


Схема ПИ-регулятора



Выясним коэффициенты перевода их кодовой системы (0...32726) в нашу систему исчисления, то есть в мл. Значения уровня в миллилитрах зависит от значения в коде по следующему закону:

$$y = ax + b, \text{ где}$$

x – значение уровня в коде,

y – в миллилитрах,

а и b – коэффициенты, которые мы ищем

Экспериментальным путем получим следующие соотношения:

2000 миллилитрам соответствует 27340 (значение в коде)

1000 миллилитрам – 16530

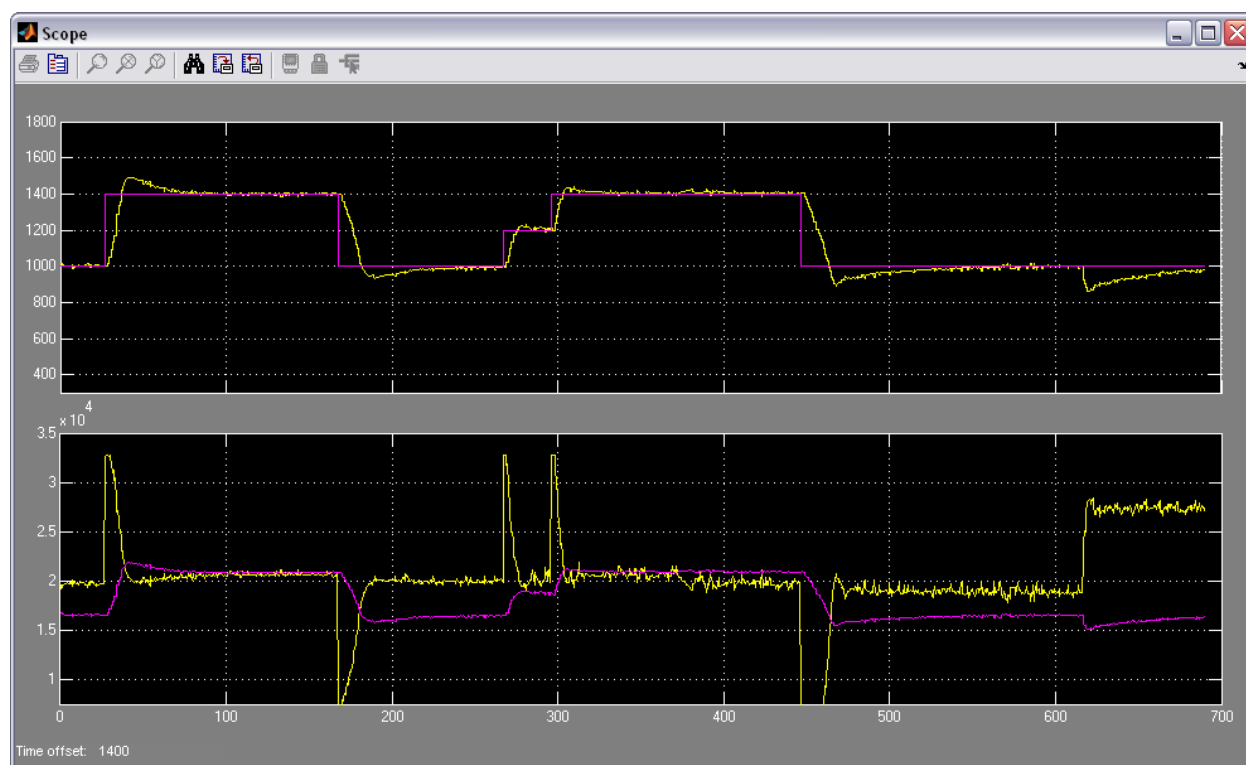
Составляем систему:

$$\begin{cases} a * 27340 + b = 2000 \\ a * 16530 + b = 1000 \\ a = 0,092507 \\ b = -529,14 \end{cases}$$

т. е. для этой установки $a = 0,092507$; $b = -529,14$

Далее настраиваем ПИ регулятор ($K_p=6$, $T_i=1/40$).

Как итог получаем следующий график переходного процесса:



Начиная с $\text{time}=450$ задаём первое возмущающее воздействие (добавили жидкость в основной бак), следующее возмущающее воздействие – открытие слива и, наконец, после установления процесса, снова закрываем слив.

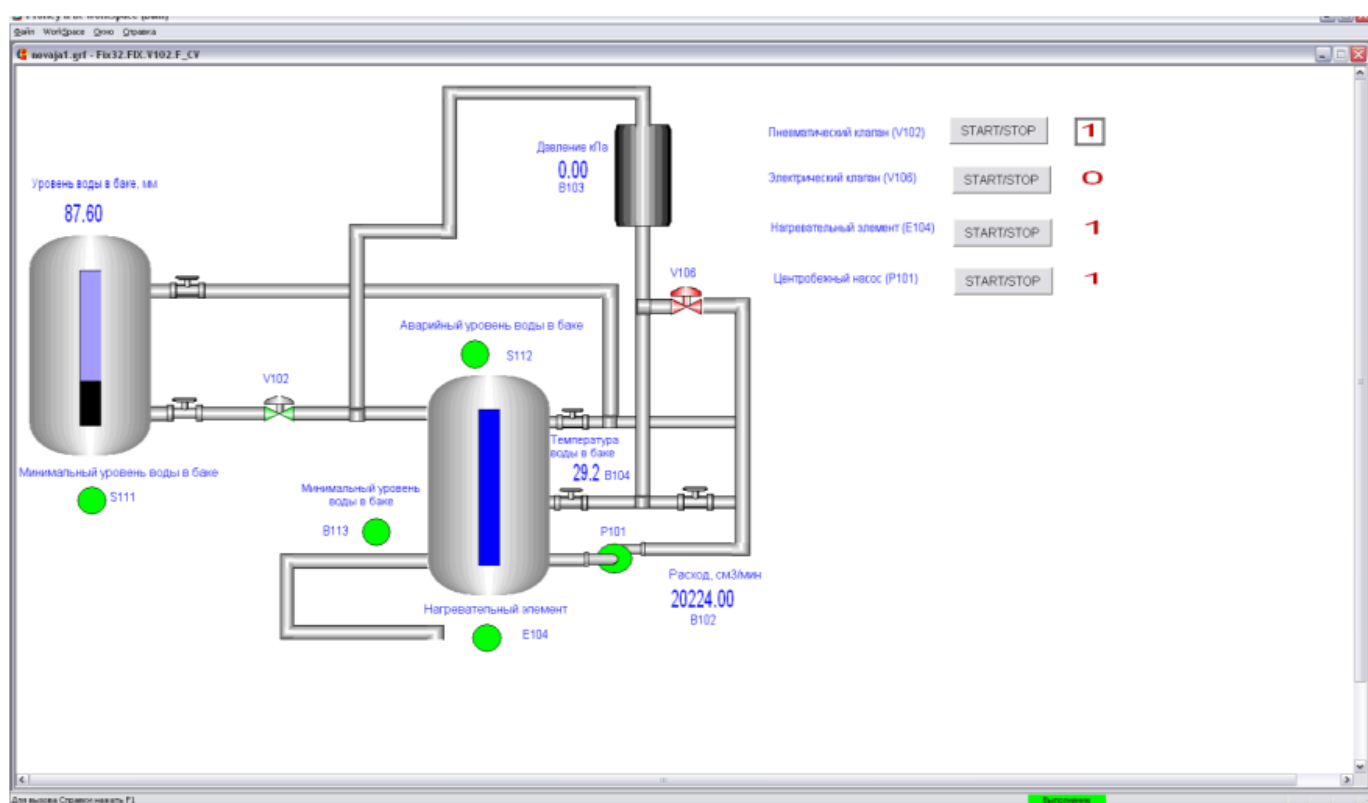


Рис. 15. Экран оператора в SCADA – системе