ПЕРВОЕ ВЫСШЕЕ ТЕХНИЧЕСКОЕ УЧЕБНОЕ ЗАВЕДЕНИЕ РОССИИ



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОРНЫЙ УНИВЕРСИТЕТ**

**ИМПЕРАТРИЦЫ ЕКАТЕРИНЫ II»**

## Кафедра автоматизации технологических процессов и производств

# Лабораторная работа №1

**ВЫПОЛНИЛИ**  АПН-22 Федотова Н.А.

(шифр группы) (подпись) (ФИО)

**ОЦЕНКА**: \_\_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ПРОВЕРИЛ** доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Федорова Э.Р.

(подпись) (ФИО)

Санкт-Петербург

2024

**Цель работы:** связать программу, написанную для ПЛК в Unity Pro XL c OPC-стандартом для того, чтобы в реальном времени следить и иметь возможность изменить переменные в программе, не выходя из режима симуляции и не загружая проект в ПЛК после изменения любой переменной.

**Ход работы**

Перед созданием проекта мы сконфигурируем контроллер с учетом того, что передача данных будет производиться на физическом уровне через Ethernet, именно поэтому был выбран модуль ПЛК Modicon P34 2020 рис. 1.

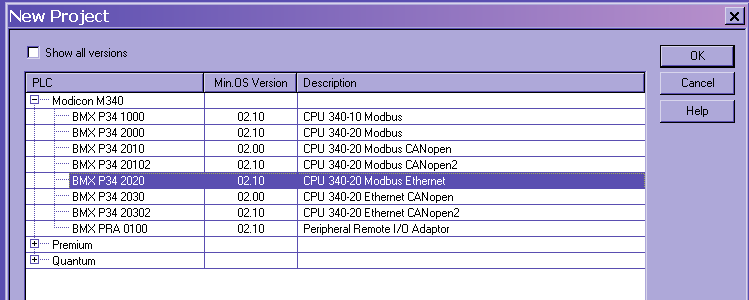


Рисунок 1 – Выбор модуля контроллера

Далее создадим проект, в котором сделаем новую секцию.

Так как структура программы у меня является однозадачной, то в ней будет выполняться только одна основная задача – MAST. Секции, в которых пишется код, соответственно, буду создавать в этой задаче (рис.2).



Рисунок 2 – Создание новой секции

После нажатия на “New Section” появляется окно, в котором мы называем нашу программу, выбираем язык, на котором мы будем её писать, т.е. FBD (рис.3).

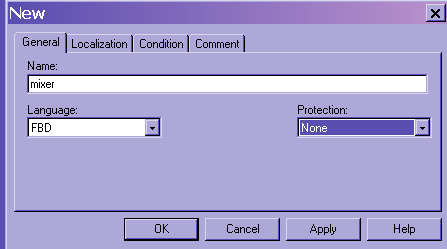


Рисунок 3 – Создание новой секции mixer

Итак, необходимо смоделировать процесс непрерывного смешивания 2-х потоков.

Известно, что в бак объемом V л, подаются 2 потока с расходами Q1 л/мин, Q2 л/мин и концентрациями C1 мг/л, C2 мг/л соответственно (Рисунок 3). На выходе бака-реактора концентрация меняется с заданным запаздыванием delay. Исходные данные для примера приведены в таблице по варианту:

Таблица 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| V, л | Q1, л/мин | Q2, л/мин | C1, л/мин | C2, л/мин | delay, с |
| 300 | 105 | 40 | 8 | 28 | 4,5 |

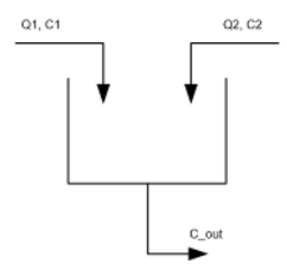


Рисунок 4 – Схема бака-реактора

Изменение выходной концентрации можно описать следующим рекуррентным уравнением:

, где

– начальное значение концентрации на выходе реактора, рассчитываемое по формуле:

Реализуем уравнение с помощью готовых блоков из стандартной библиотеки Unity Pro XL. В данной схеме операции сложения, вычитания, умножения и деления осуществляется соответственно блоками ADD, SUB, MUL и DIV, которые взяты из библиотеки Baselib/Mathematics. Блоки INTEGRATOR и DTIME, которые используются для реализации интегрирования и запаздывания соответственно, взяты из библиотеки CONT CTL.

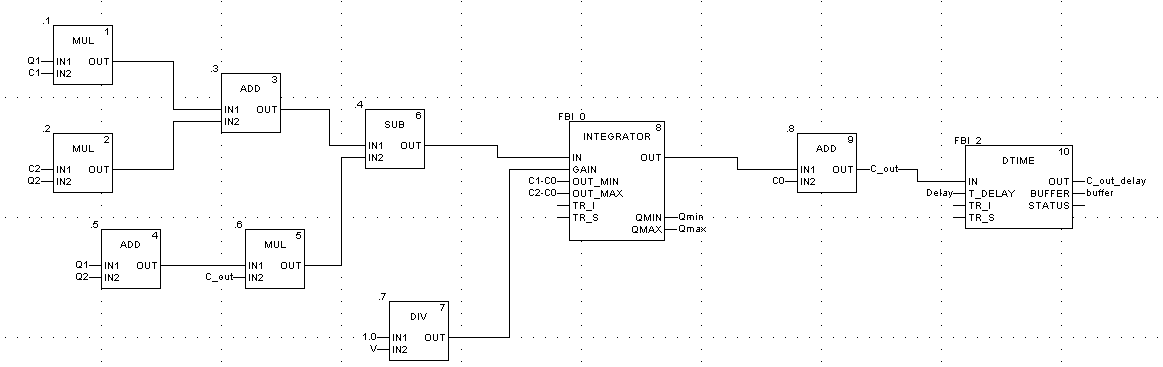


Рисунок 5– Собранная схема без регулятора

Для нормальной работы блока DTIME требуется создать массив данных, в котором будет храниться значение входного сигнала. Объявление массива производится в окне Data Editor, задается имя переменной, например buffer, и задается тип массив. Размер массива определяется формулой:

Где N – размер массива, τ – время запаздывания, Δt – время дискретизации (время цикла). Зайдем в свойства секции и установим любое время цикла, я выбрала его равным 200 мс (рис.5):

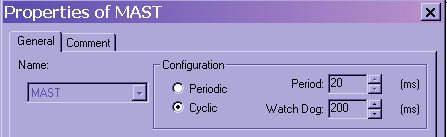


Рисунок 5 – Время дискретизации

Рассчитаем необходимый размер массива:

В итоге были созданы необходимые переменные, которые приведены на рисунке 6.



Рисунок 6 – Таблица созданных переменных

Введем в систему ПИ-регулятор, который представлен в среде моделирования функциональным блоком PI\_B, осуществляющим реализацию смешанного ПИ-алгоритма. Блок PI\_B можно применять для решения большинства классических задач управления. Блок отличается относительной легкостью использования и экономичностью системных ресурсов.

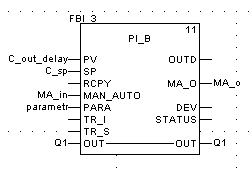


Рисунок 7 – Блок ПИ-регулятор

Передаточная функция блока:

Данная функция меняется в зависимости от применяемого типа алгоритма: абсолютного или инкрементного.

При постоянной времени изодрома равной нулю (ti = 0) применяется абсолютная форма функции:

При постоянной времени изодрома неравной нулю (ti > 0) применяется инкрементная форма функции:

величина пропорциональной составляющей

величина интегральной составляющей

значение, полученное при текущем выполнении блока

значение, полученное на предыдущем выполнении блока

Ниже приведен скриншот с описанием параметров созданного блока PI\_B.

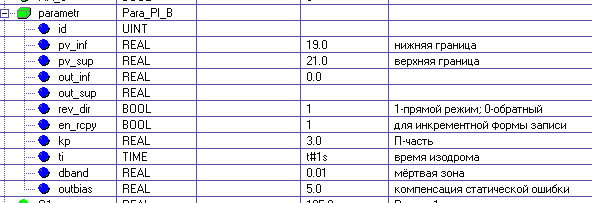


Рисунок 8 – Настройка блока PI\_B

Блок PI\_B может работать в трех режимах: автоматическом, ручном и в режиме слежения.

• В автоматическом режиме TR\_S = 0; MANAUTO = 1.

• В ручном режиме TR\_S = 0; MAN\_AUTO = 0.

• В режиме слежения TR\_S = 1; вход MANAUTO равен 0 или 1.

При инкрементной форме вычисления можно выносить значение интегральной составляющей за пределы блока, при en\_rcpy = 1, это рекомендуется делать, когда фактическое значение отличается от рассчитанного регулятором. Если доступно последнее вычисленное значение, оно должно быть подано на вход RCPY и значение параметра en\_rcpy должно быть равно 1. Тогда при вычислении в блоке выражение примет следующий вид:

Использование зоны нечувствительности позволяет сгладить всплески величины рассогласования внутри этой зоны (а именно обратить их в нуль).

Использование параметра outbias позволяет обеспечить точность срабатывания при отсутствии интегральной составляющей (ti = 0).

Следующим шагом создадим окно оператора, на котором будет отображаться вся необходимая информация: графики sp и pv, значения параметров.

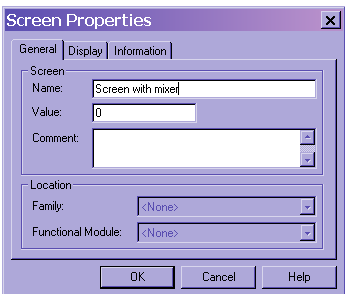
 

Рисунок 9 – Создание окна оператора

На открывшемся полотне будем выводить графики, для этого добавим объект Rectangle.



Рисунок 10 – Панель функционала

Теперь зайдем в свойства созданного объекта и привяжем к нему необходимую переменную, которую нужно выводить, т.е. конечную концентрацию (C\_out\_delay). Линию графика целевого параметра сделаю зеленым цветом.

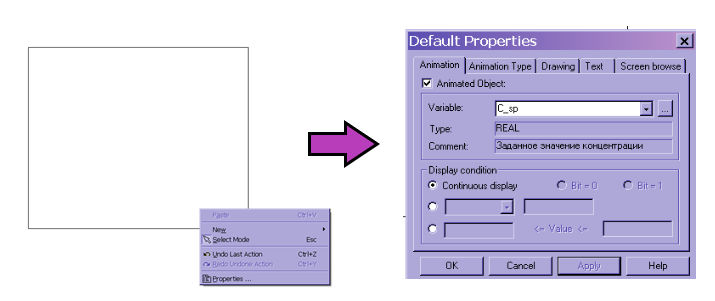


Рисунок 11 – Настройка окна графика

На вкладке Animation Type установим тип анимации Trend diagram и нажмем >.

Sampling устанавливает период дискретизации, то есть как часто появляется новая точка на графике. Definition – разрешение графика, количество пикселей на 1 точку. В зависимости от геометрических размеров объекта, периода дискретизации и разрешения устанавливается различное значение Display period (отображаемый период). Это значение соответствует периоду сигнала, которое полностью отображается на экране в один момент времени. Thresholds (пределы) ограничивают график снизу и сверху.

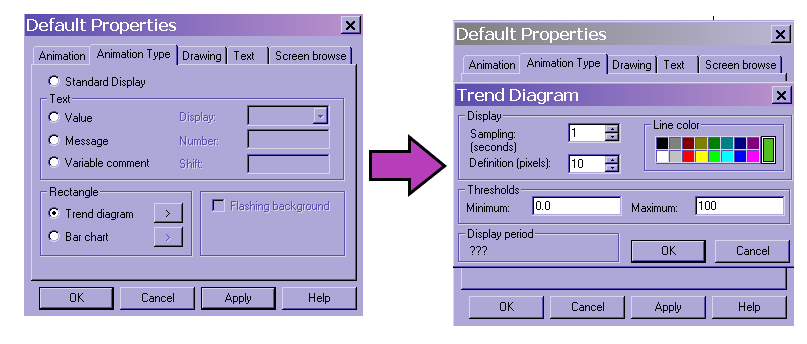


Рисунок 12 – Настройка графика

Построим аналогичным образом график для pv, который будет выделен красным цветом, а также выведу на график 5% зону – линии синего цвета.

Помимо графиков, выведем на окно оператора кнопки, на которых будут отображаться текущие значения параметров, а также создам лампочку, которая будет показывать включен или выключен ручной режим.

Проверим работоспособность созданных графиков. Первым шагом нажмем Simulation Mode.



Рисунок 13 – Симуляция модуля

Далее проверим соединение, после чего подключимся к ПЛК:

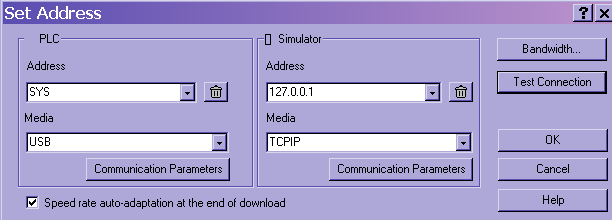
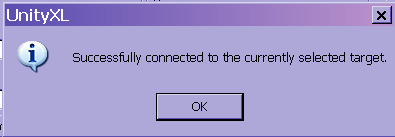


Рисунок 14 – Проверка адреса



Рисунок 15 – Подключение

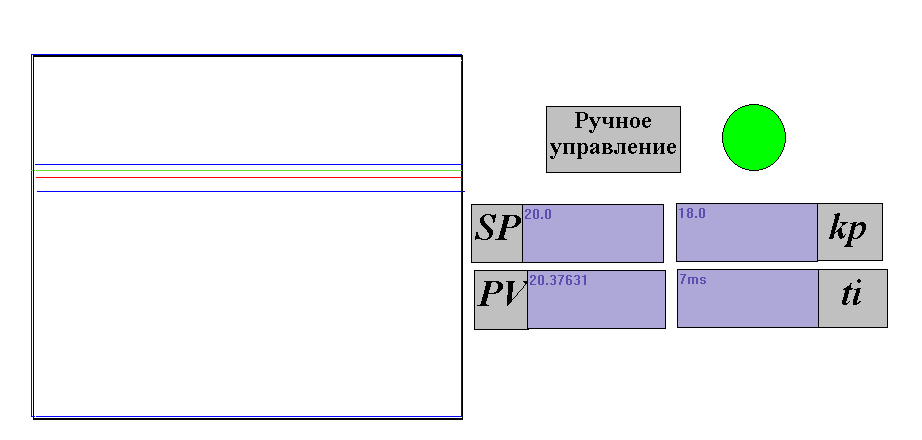


Рисунок 16 – Окно оператора

Для создания Client-Serverной архитектуры на OPC понадобится файл с таблицей переменных данного проекта, поэтому следующим шагом будет создание XVM файла.

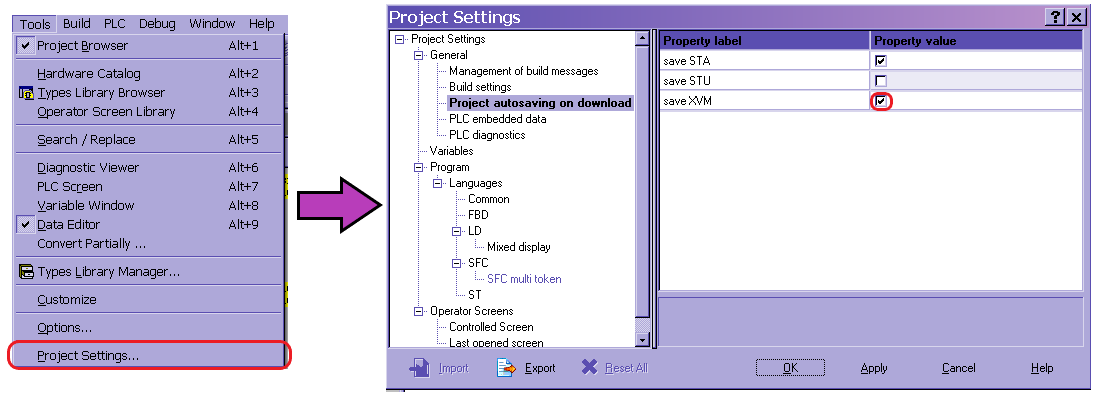


Рисунок 17 – Создание файла XVM

Зададим путь для сохранения файла. Этот же путь пропишем в OFS Configuration Tool:

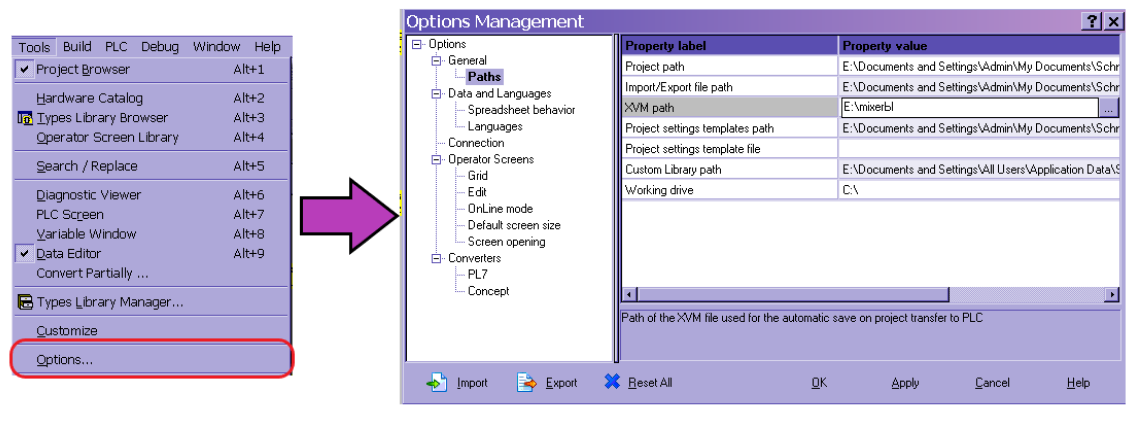


Рисунок 18 – Путь для сохранения созданного файла

Следующим важным шагом будет настройка Client-Serverной архитектуры на OPC от Schneider Electric.

Сначала откроем OFS Configuration Tool

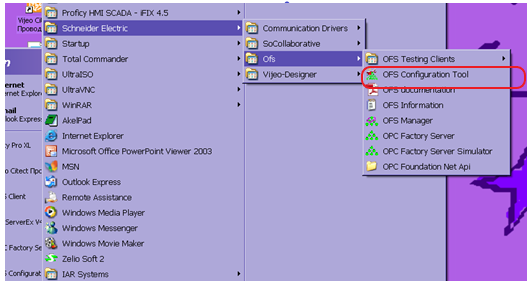


Рисунок 19 – Открытие OFS Configuration Tool

Выберем File -> New Device Alias. Для того, чтобы сконфигурировать новое устройство вы должны обладать следующей информацией:

• Какую сеть вы используете,

• Адрес устройства в этой сети,

• Файл таблицы символов (переменных), в которой содержится описание информации, которую вы хотите получить с устройства.

• Программа-симулятор ПЛК выполняется локально и имеет адрес 127.0.0.1:502 (502 – адрес порта)

• В разделе Device Address указываем протокол TCP/IP и, собственно, адрес: 127.0.0.1 (рис.21).

• В качестве адреса таблицы символов, указываем папку, куда был сохранен XVM-файл (рис.20):

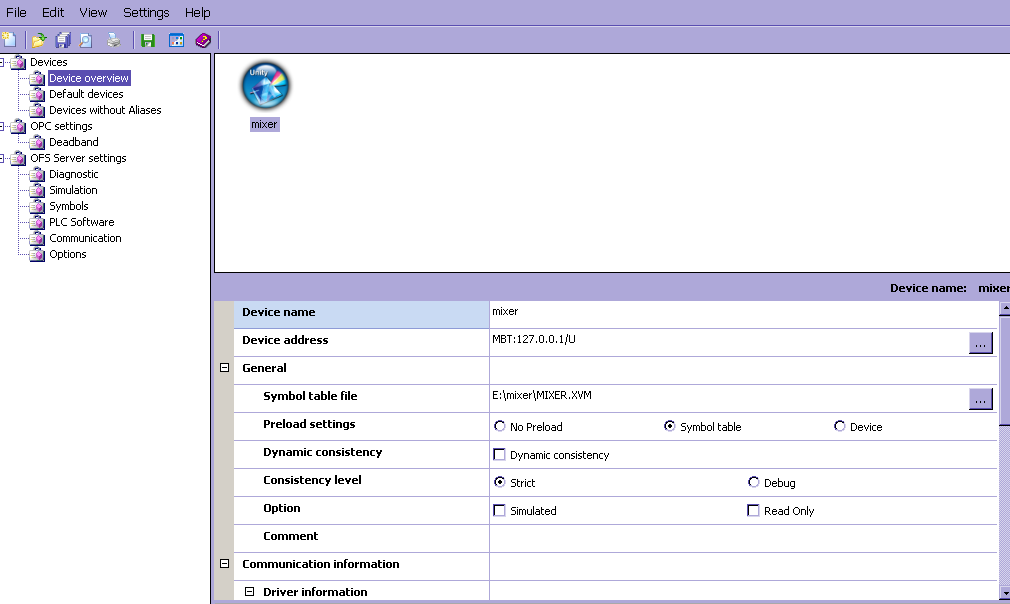


Рисунок 20 – Конфигурация нового устройства

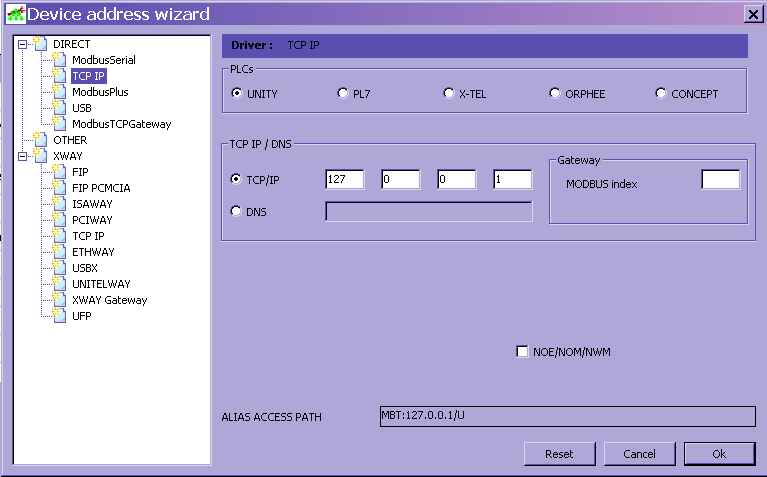


Рисунок 21 – Задание адреса устройства

При создании ОРС сервера (рис.22) появится окно, предупреждающее, что это демоверсия и работать в ней возможно 72 часа (рис.23).

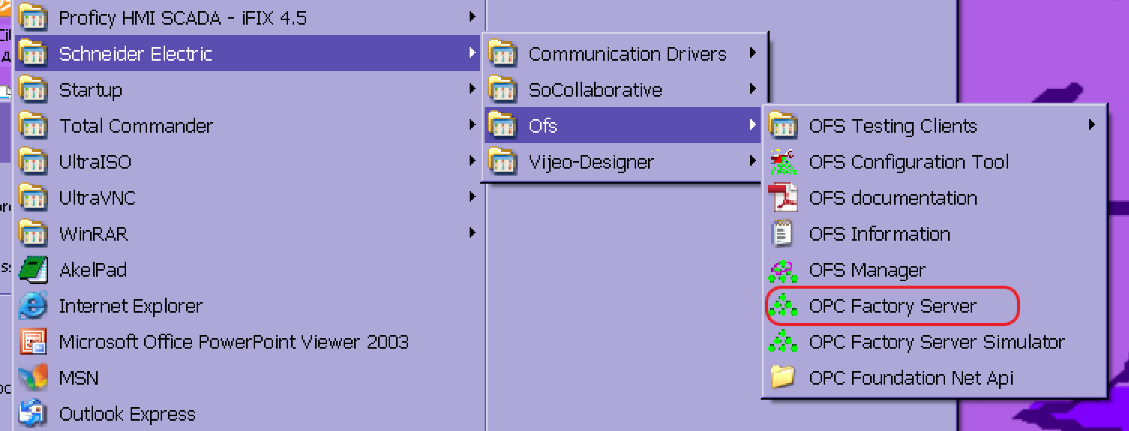


Рисунок 23 – Создание ОРС сервера

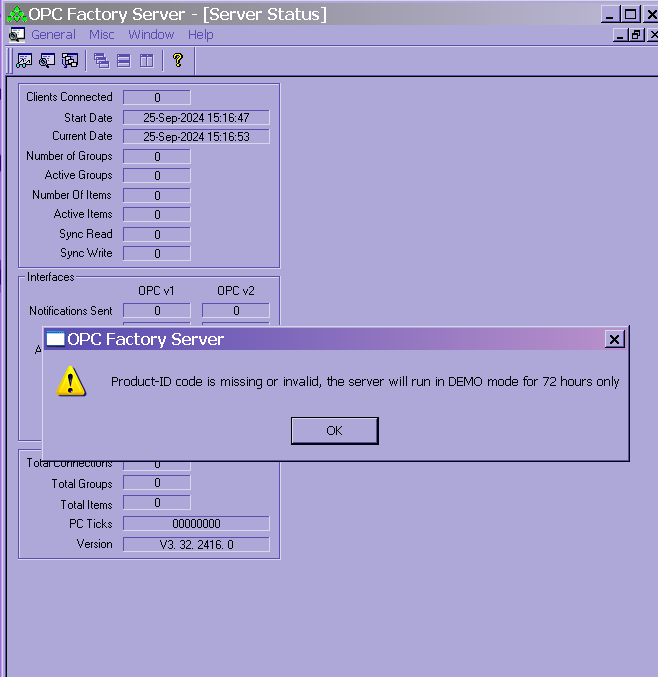


Рисунок 24 – Предупреждение о демоверсии

Далее для тестирования работы сервера запустим OFS Сlient (рис.25), который входит в пакет установки OPC Factory Server. Создадим новый проект. Из появившегося списка серверов выберем ранее настроенный сервер (рис.26).

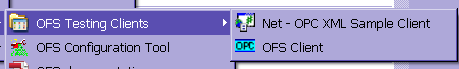


Рисунок 25 – Открытие OFS Сlient

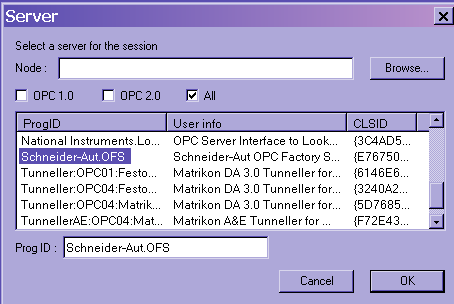


Рисунок 26 – Выбор созданного сервера

В подсмотрщике сервера OFS (OFS Client) нажмем File-New и создадим новую группу ТЭГов (рис.27), которые необходимо проверить на считывание и запись.

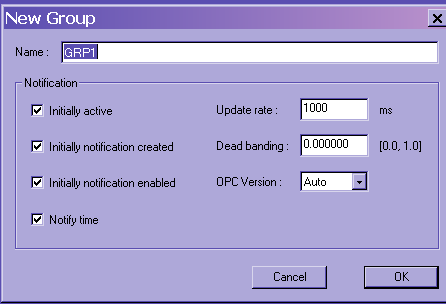
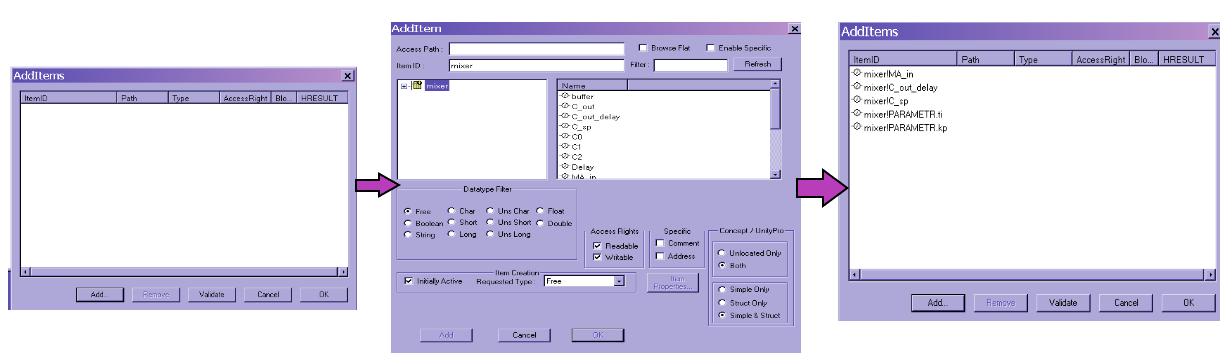


Рисунок 27 – Создание группы ТЭГов

В созданную группу добавим переменные (рис.28):





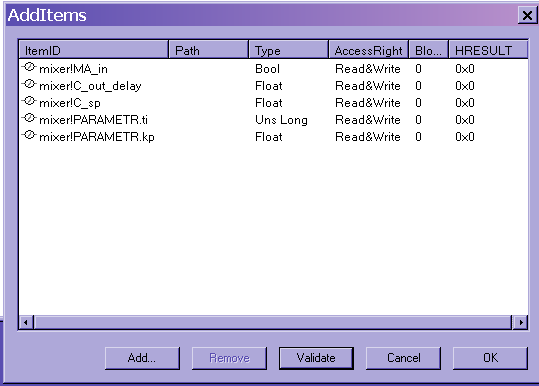
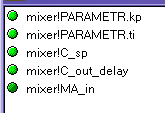


Рисунок 28 – Создание переменных

Если теперь посмотреть на OPC Factory Server, то можно увидеть, что появились активные точки (соединение установлено):

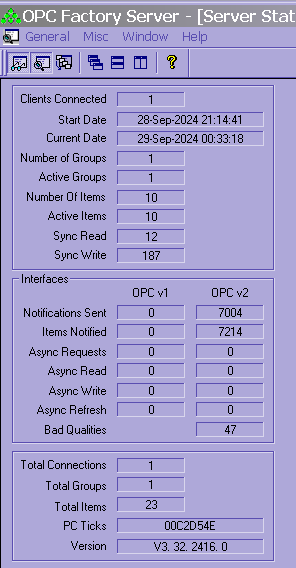


Рисунок 29 – Проверка соединения через OPC Factory Server

Теперь проверим, есть ли связь между программой в ПЛК и настроенным OPC‑сервером. Параллельно проверке буду ручным способом настраивать ПИ-регулятор, изменяя значения переменных kp и ti в OFSClientе. Настраивать регулятор буду исходя из анализа графиков, построенных ранее в окне оператора.

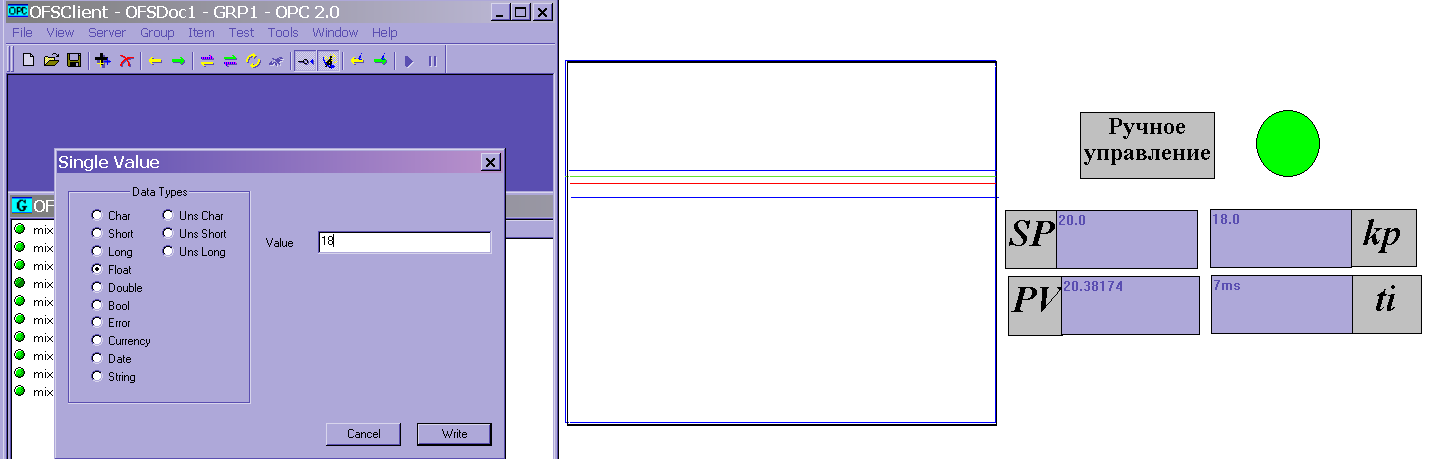


Рисунок 30 – Проверка соединения между созданным сервером и написанной программой ПЛК