

Методическое пособие
«Freelance 2016. Быстрый старт»
(работа с эмулятором контроллера)

Автор к.т.н. Шаровин И.М.

Москва 2019г.

Оглавление

1.	ВВЕДЕНИЕ	4
2.	УСТАНОВКА ПО FREELANCE НА ПК	5
3.	ЦЕЛЬ РАБОТЫ	8
4.	СОЗДАНИЕ ПРОЕКТА	9
4.1.	СОЗДАНИЕ НОВОГО ПРОЕКТА.....	9
4.2.	ПРОВЕРКА ПРОЕКТА НА НАЛИЧИЕ ОШИБОК	13
4.3.	НАСТРОЙКА ЭМУЛЯТОРА КОНТРОЛЛЕРА.....	14
4.4.	РЕАЛИЗАЦИЯ ПРОСТОГО ПРИМЕРА (ПРОВЕРКА РАБОТОСПОСОБНОСТИ ПРОЕКТА).....	17
5.	СОЗДАНИЕ ФУНКЦИОНАЛЬНЫХ БЛОКОВ И ПРОГРАММЫ ПОЛЬЗОВАТЕЛЯ.....	21
5.1.	ФУНКЦИОНАЛЬНЫЙ БЛОК АПЕРИОДИЧЕСКОГО ЗВЕНА.....	21
5.2.	ФУНКЦИОНАЛЬНЫЙ БЛОК РЕГУЛИРУЮЩЕГО КЛАПАНА	23
5.3.	ПРОГРАММА ПОЛЬЗОВАТЕЛЯ	24
6.	СОЗДАНИЕ ИНТЕРФЕЙСА ОПЕРАТОРА (ВИЗУАЛИЗАЦИЯ ПРОЦЕССА).....	32
6.1.	СОЗДАНИЕ МНЕМΟΣХЕМЫ.....	32
6.2.	ЗАГРУЗКА МНЕМΟΣХЕМЫ ВО FREELANCE OPERATIONS	35
7.	ВМЕСТО ПОСЛЕСЛОВИЯ.....	36

Перечень используемых сокращений в пособии:

АСУТП – автоматизированная система управления технологическим процессом

ПИ – пропорционально-интегральный регулятор

ПК – персональный компьютер

ПО – программное обеспечение

ПТК – программно-технический комплекс

САУ – система автоматического управления

ФБ – функциональный блок

FBD – Function Block Diagram (язык программирования блоками)

ST – Structured Text (язык программирования структурированный текст, паскалеподобный)

1. Введение

Программно-технический комплект (ПТК) ABB Freelance достаточно распространён в качестве системы управления на энергетических объектах России (системы где количество сигналов не превышает 1200 шт. на один контроллер).

В настоящее время ПТК Freelance в разных версиях (v.9.1; v.9.2, v.2000, v.2013, v.2016) внедрён на более чем 30 объектах генерации. Среди них:

- Артёмовская ТЭЦ
 - Владивостокская ТЭЦ-2
 - Владимирская ТЭЦ-2
 - Железногорская ТЭЦ
 - Иркутская ТЭЦ-9
 - Иркутская ТЭЦ-10
 - Иркутская ТЭЦ-11
 - Курганская ТЭЦ
 - Невинномысская ГРЭС
 - Ново-Зиминская ТЭЦ
 - Ново-Рязанская ТЭЦ
 - ТЭЦ НЛМК (Новолипецкий металлургический комбинат)
 - Южно-Кузбасская ГРЭС

и многие другие

В ходе реализации проектов основной проблемой передачи ПТК в промышленную эксплуатацию является необходимость обучения персонала станции. Обучение проводится подрядной организацией один раз, а людям свойственно со временем забывать очередность создания проекта и нюансы реализации элементов. Существующие инструкции ABB (в прочем как и любого другого производителя ПТК) не описывают процесс создания проекта с нуля. Количество простейших операций для создания проекта для новичка пугающе велико, а время обучения ограничено. Чтобы сделать процесс обучения более гибким, а также помочь людям в освоении программно-технического комплекса было разработано и создано пособие по быстрому старту системы Freelance. Данное пособие демонстрирует создание проекта в эмуляторе контроллера своими руками с нуля. Пособие помогает не только службе эксплуатации ТЭЦ тестировать новые функциональные блоки и программы на эмуляторе, но также обеспечивает быстрое погружение программистов (при приёме на работу) в систему без длительных объяснений.

Десятки людей создали свои проекты с помощью данного пособия. Предлагаю попробовать и вам освоить ПТК ABB Freelance самостоятельно. Для людей неискущённых в создании программного обеспечения, а также для желающих просто посмотреть систему имеется уже реализованный проект (см. информацию на сайте www.sharovin.ru). Проект можно открывать, модернизировать и запускать на эмуляторе контроллера. Если проект в ходе ваших экспериментов перестанет работать, то всегда можно ознакомиться с методическим пособием и вернуть ему работоспособность своими силами (или скачать исходный файл повторно).

2. Установка ПО Freelance на ПК

Любая работа с программным обеспечением начинается с его установки на ОС Windows. Т.к. мы планируем работать с эмулятором контроллера, а библиотеки конфигурационных файлов для инсталляции модулей ввода/вывода нам не нужны, то необходимо проделать ряд манипуляций с диалоговыми окнами при установке ПО на компьютер. Ниже описываются только окна в которых пользователь имеет выбор какую кнопку или ссылку нажать. Окна, где пользователь должен прочитать информацию и нажать кнопку «Далее» или «Завершить» не приводятся.

ПО Freelance делится на 2 основных вида:

1. ПО для разработчика (программиста) – Freelance Engineering
2. ПО для оператора Freelance Operations.

В трудовой деятельности (на производстве) эти 2 пакета никогда не ставятся на один персональный компьютер. При создании проекта в лабораторных условиях необходимо совмещать эти два пакета на одном ПК и Freelance имеет такую возможность. Ниже приведено последовательное описание установки ПО. Особое внимание обращайте на галочки при установке пакетов.

Распаковать архив и в папке нажать на файл запуска установки ПО «AutoPlay.exe»

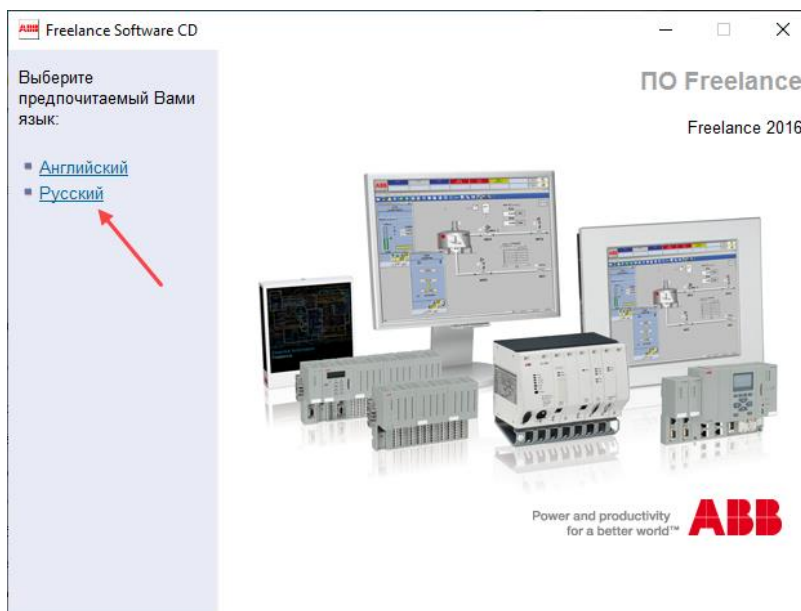


Рис.1 Приветственное окно установки Freelance 2016. Выбор языка

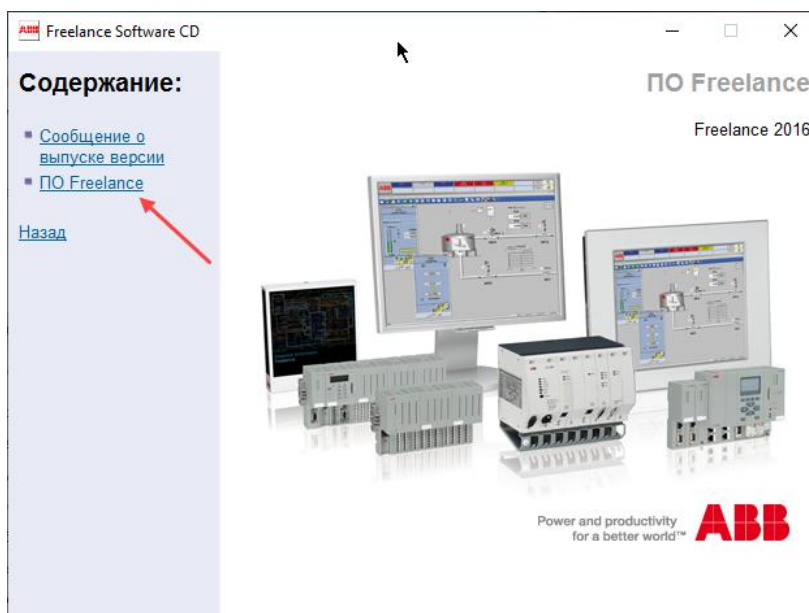


Рис.2 Установка пакета

Особое внимание необходимо уделить следующему окну. Т.к. мы используем компьютер для установки ПО для тестирования (у нас нет лицензии и аппаратных лицензионных ключей), то мы должны поставить галочку «Демо». Если выбрать «Производственный режим», то программа создаст нового пользователя на компьютере и при старте компьютера будет автоматически переходить в созданную учётную запись и запускать приложение оператора. Удалить «Производственный режим» из своего компьютера достаточно проблематично. Будьте внимательны!

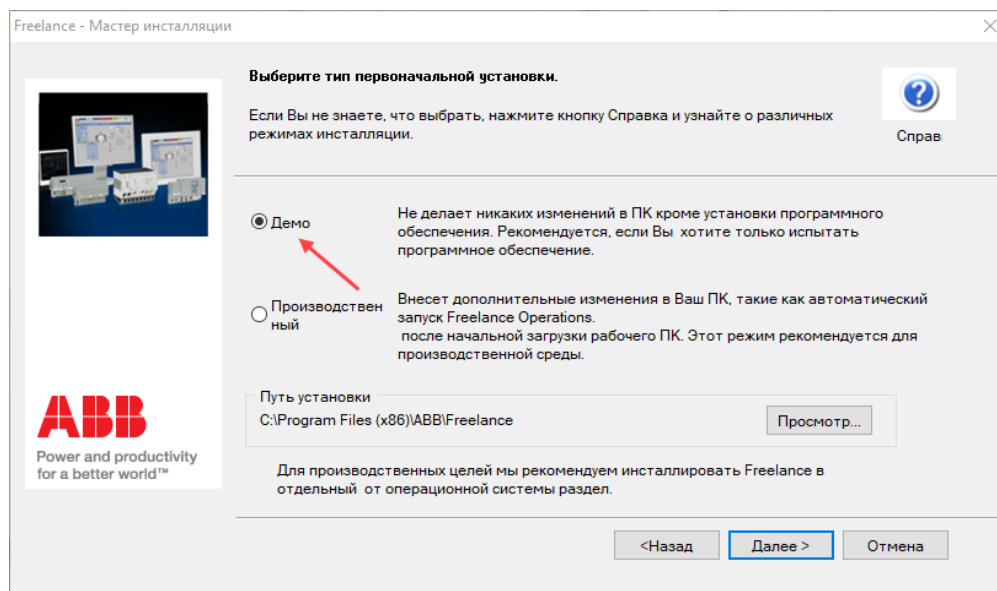


Рис.3 Выбор демо режима работы.

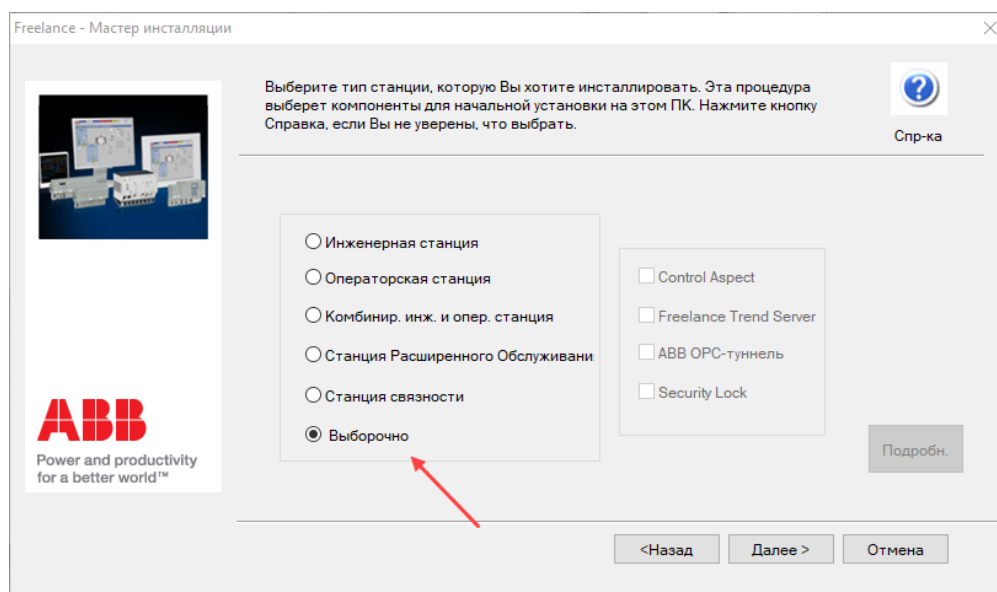


Рис.4 Выборочная установка пакетов на ПК

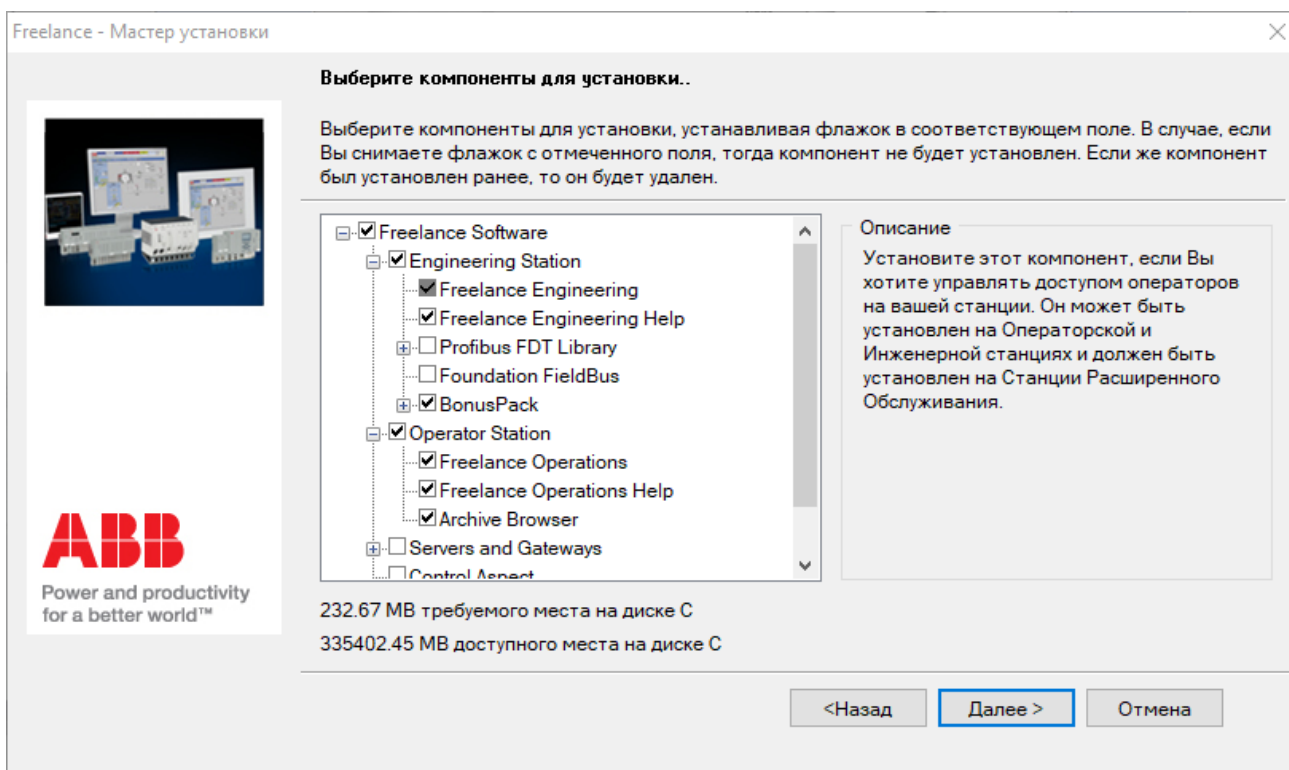


Рис.5 Выбор необходимого к установке ПО (внимание на галочки)

После выбора необходимых пакетов произойдет установка ПО на компьютер и откроется окно «Настройки». В настоящий момент настраивать нам нечего т.к. отсутствует тело проекта. В окне настройки ничего не меняем и нажимаем кнопку «Ок» (рис. 6)

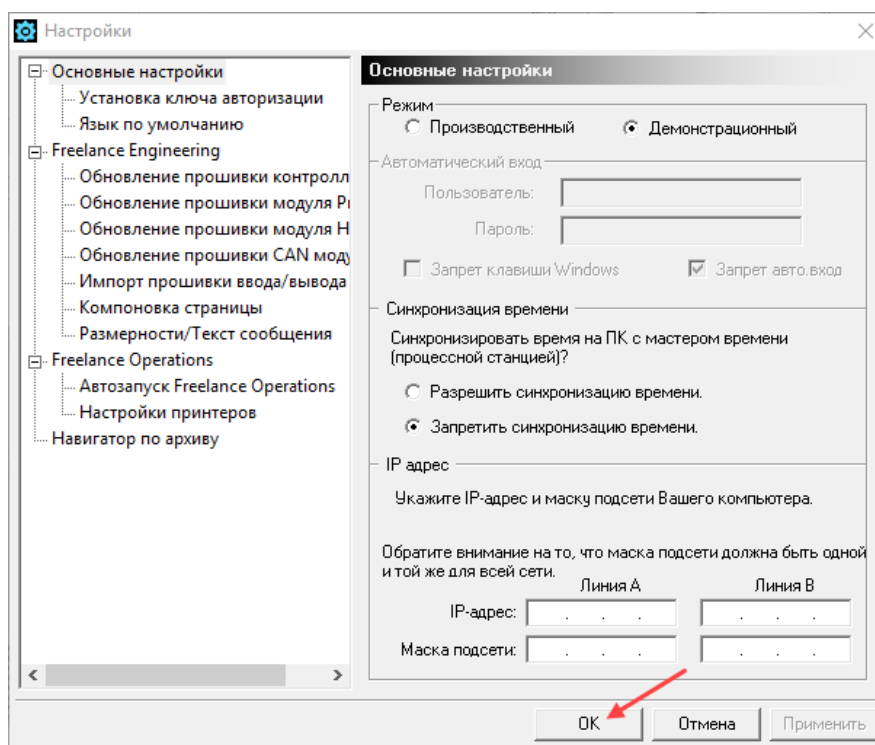


Рис.6 Окно настройки параметров

После установки пакета обязательно перезагрузите компьютер! В противном случае эмулятор контроллера просто не будет работать.

3. Цель работы

Целью данного методического пособия является изучение программно-технического комплекса (ПТК) компании ABB (ООО «АББ») и непосредственное овладение навыками работы в программной среде разработки «Freelance Engineering» и SCADA-системы «Freelance Operations».

Итогом работы должна стать мнемосхема, при помощи которой оператор, т.е. Вы, сможет управлять технологическим объектом (рис.7).

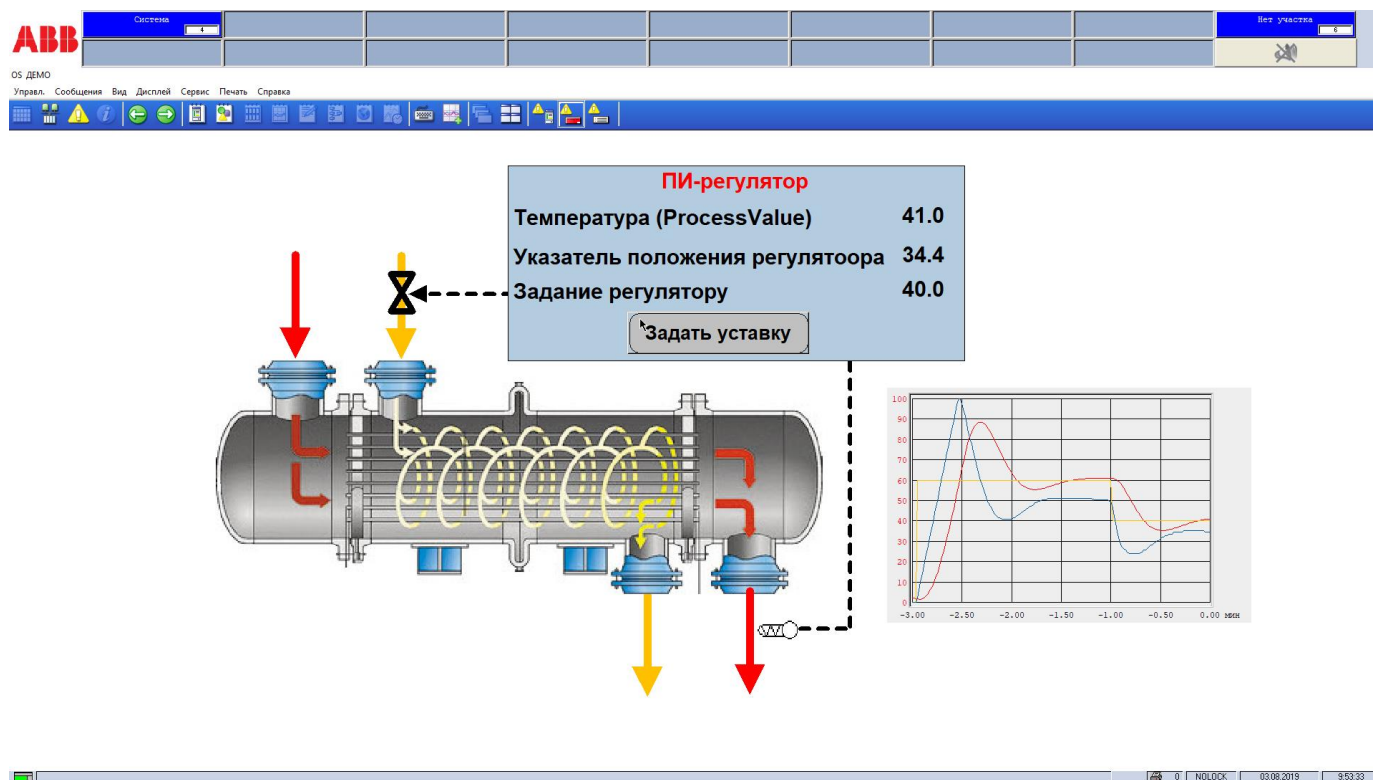


Рис.7 Мнемосхема технологического процесса на операторской станции

В нашем проекте будет применена одноконтурная система автоматического регулирования (САУ) с импульсным регулятором, реверсивным исполнительным механизмом (регулятор расхода пара) и объектом управления.

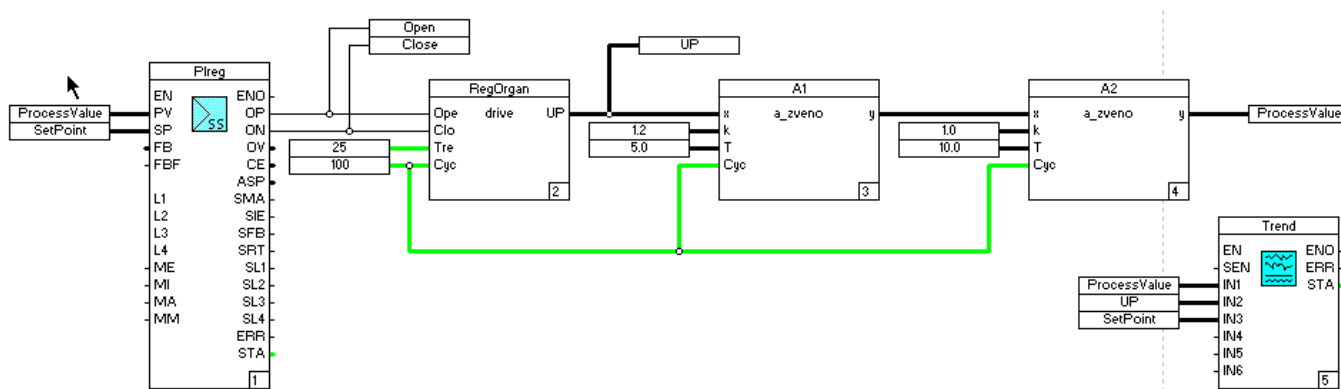


Рис.8 Программа реализации одноконтурной САУ

Для регулирования объекта применим ПИ-регулятор с передаточной функцией вида:

$$W_{рег}(s) = k_p + \frac{k_i}{s},$$

(1)

где k_p - коэффициент передачи регулятора; T_u - постоянная времени интегрирования.

Передаточная функция объекта регулирования представляет последовательное соединение двух апериодических звеньев и звена запаздывания:

$$W_o(s) = \frac{k_o}{(T_1 \cdot s + 1) \cdot (T_2 \cdot s + 1)}, \quad (2)$$

где $k_o = 1.2$, $T_1 = 5 \text{ сек.}$, $T_2 = 10 \text{ сек.}$

Каждый элемент программы и мнемосхемы будет создаваться отдельно и только на завершающем этапе выполнения работы все объекты объединяться в систему автоматического регулирования.

Здесь отдельно стоит отметить, что часто у молодых специалистов происходит путаница в связи с тем, что специалисты употребляют одно и тоже слово в разных значениях. Следует различать контекст слова **РЕГУЛЯТОР**. Регулятором одновременно может считаться вычислительное устройство (говорят П-регулятор, ПИ-регулятор, ПИД-регулятор), также регулирующий орган (регулирующая арматура) или ещё его можно назвать исполнительный механизм. Когда на объекте говорят, что регулятор работает исправно, то имеется ввиду, что связка ПИ-регулятора и регулирующей арматуры взаимодействует так как и должно быть. Со временем данная терминология утрясается и путаницы не происходит.

В нашем случае мы проводим лабораторные испытания и чтобы показать самый распространённый тип ПИ-регулятора импульсного типа (работает командами «Больше», «Меньше» или «Открыть», «Закрыть»), то нам приходится реализовывать регулирующий орган также в программе. Это полезно для эмуляции технологического процесса, а также для отработки математических моделей перед их реализацией непосредственно на объекте.

4. Создание проекта

Первым этапом будет создание одноконтурной САР, после чего созданные переменные будут привязаны к нарисованной мнемосхеме.

4.1. Создание нового проекта

Открыть Freelance Engineering и нажать в панели меню «Проект» - «Новый...» (Рис.9)

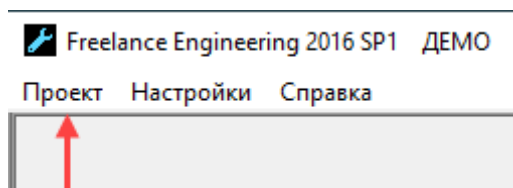


Рис.9 Создать новый проект

В открывшемся окне выбрать директорию в которой будет храниться проект и дать ему название. В нашем случае название тестового проекта будет «test_abb». После чего откроется окно «Конфигурирование: заголовок проекта» (рис.10), заполнить его в соответствии с приведенным рисунком. Сохранить результат нажатием на кнопку «ОК».

Конфигурирование: Заголовок проекта

Имя проекта: test_abb

Руководитель проекта: Шаровин

Номер проекта:

Заказчик проекта:

Номер заказа:

Пароль проекта: Изменить Автом.рез.копиров

Размер проекта: ОК Версия: 27.07.2019 20:46:18

Коммент. к проекту: Тестовый пример для освоения ПТК

Графика верхн. колонтитула Графика нижн. колонтитула

Текст верхн. колонтитула Текст нижн. колонтитула

ОК Отмена

Рис.10 Окно «Конфигурирование: Заголовок проекта»

По умолчанию программа сразу отрывает окно структуры проекта. В данном окне можно создавать проект.

В среде разработки имеется два режима работы – конфигурирование (создание проекта в режиме off-line) и наладка (просмотр текущих значений переменных проекта и возможность изменения их значений в режиме on-line). В версии Freelance 2016 эти значки сменяют друг друга при нахождении в одном из режимов. При конфигурировании системы вы можете увидеть значок «Наладка». В режиме наладки на том же самом месте можно увидеть значок «Конфигурирование» (рис.11, 12). На данном этапе пока настраивать нечего и поэтому надо работать в режиме конфигурирования, т.е. видеть перед собой меню как на рис.11.

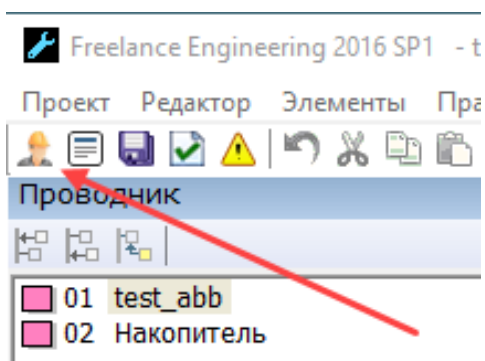


Рис.11 Кнопка «Наладка»

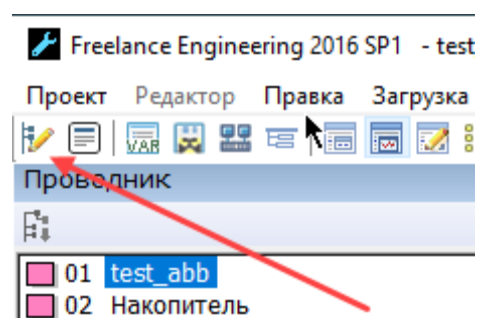


Рис.12 Кнопка «Конфигурирование»

Нажать на «test_abb» и вызвать контекстное меню. В меню выбрать «Вставить» - «Следующий уровень». В открывшемся окне выбрать единственно возможный элемент - «Конфигурация CONF», приняв настройки, заданные системой по умолчанию.

В созданном «test_abb [CONF]» создать следующие уровни (аналогично через контекстное меню):

1. Аппаратная часть (присвоить имя Software)
2. Программное обеспечение (присвоить имя Hardware)

В Software (SW) вставить следующие уровни:

1. Процессовая станция D-PS (присвоить имя PS)
2. Операторская станция D-OS (присвоить имя D-OS)
3. Накопитель пользовательских функциональных блоков P-FB (присвоить имя UFB)
4. Накопитель графических макросов P-MAC (присвоить имя MAC)
5. Шлюзовая станция D-GS (присвоить имя SCHL)

Созданный проект должен выглядеть как на рис.13.

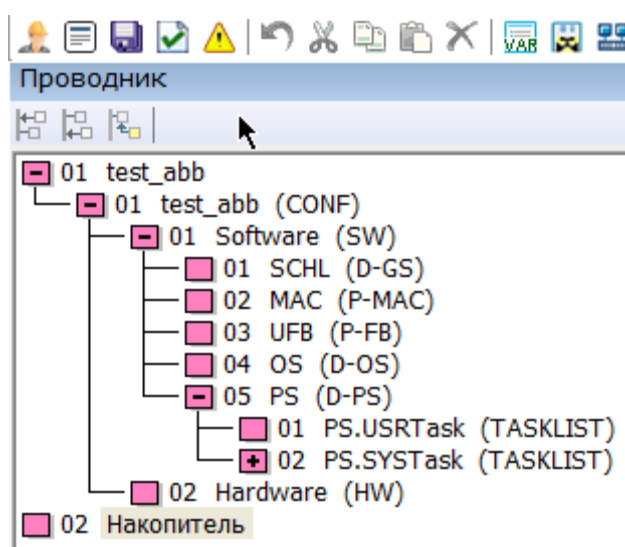


Рис.13 Созданное дерево проекта

Зайти в шлюзовую станцию «SCHL (GWY)» и поменять тип шлюза со стандартного DCP на TRN-шлюз (рис.14)

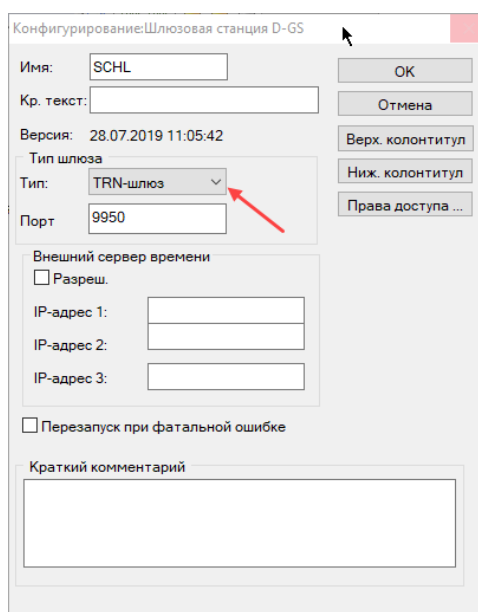


Рис.14 Конфигурирование шлюза

Мы создали скелет проекта. Теперь необходимо указать программе, что мы будем работать с эмулятором контроллера, а также сконфигурировать аппаратную часть проекта. Для этого необходимо перейти в структуру аппаратных средств (рис.15)



Рис.15 Кнопка «Структура аппаратных средств»

Кроме существующей станции (компьютер, с которого вы конфигурируете проект) ничего более в структуре аппаратных средств пока нет (рис.16).

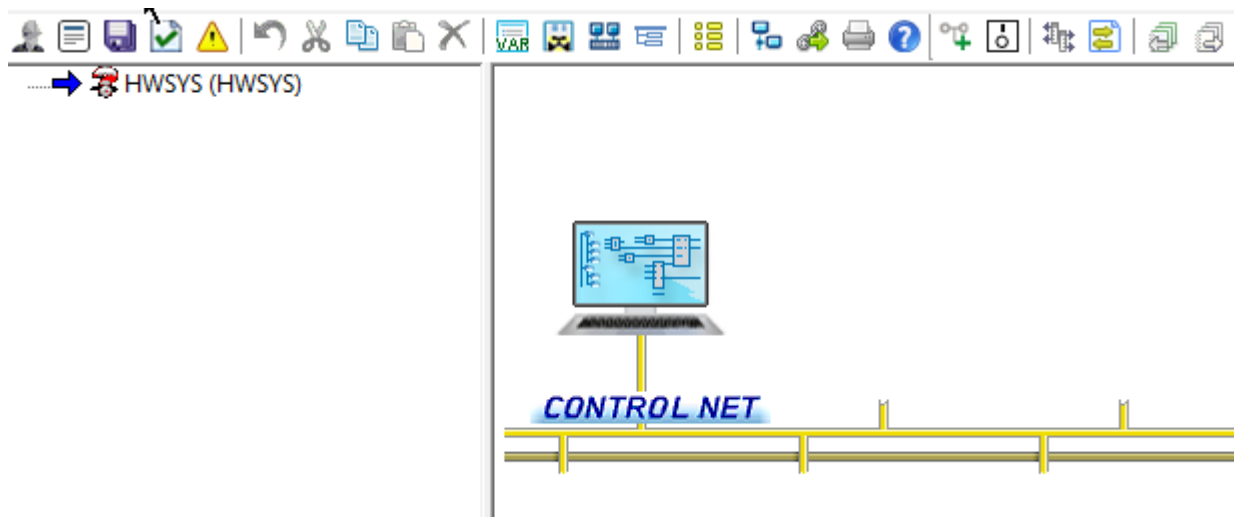


Рис.16 Структура аппаратных средств. Первоначальное состояние

Двойным нажатием по пустой области справа от графического элемента компьютера необходимо вызвать окно которое предложит выбор оборудования для вставки в аппаратную структуру. Необходимо вставить один шлюз, а также одну операторскую станцию выше жёлтой линии ЛВС, а также эмулятор контроллера в нижней части сети. Структура проекта должна принять вид как на рис.17.

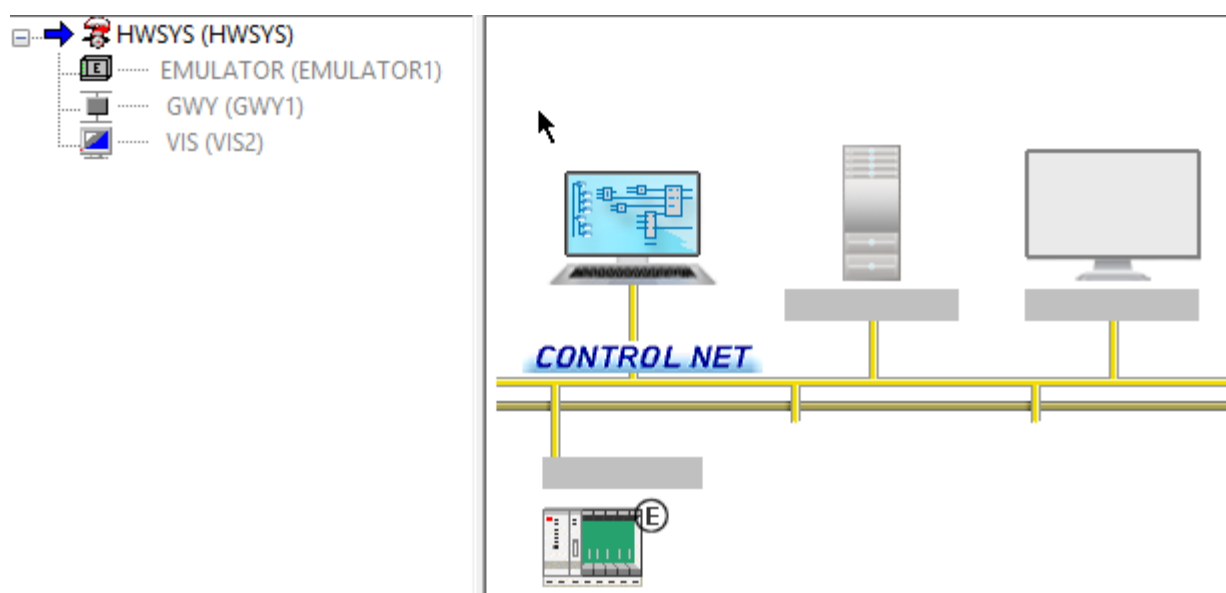


Рис.17 Структура аппаратных средств после вставки всех необходимых элементов

Теперь у нас в проекте есть скелет программной части и скелет аппаратной части. Осталось только их между собой сочленить (данная операция во Freelance называется

распределением). Для этого необходимо двойным щелчком мыши нажать на серые области над эмулятором и под шлюзом и операторской станцией.

После того как мы распределили все программные ресурсы по аппаратным средствам появятся подписи ко всем устройствам, а картинки из серых станут цветными как на рис.18.

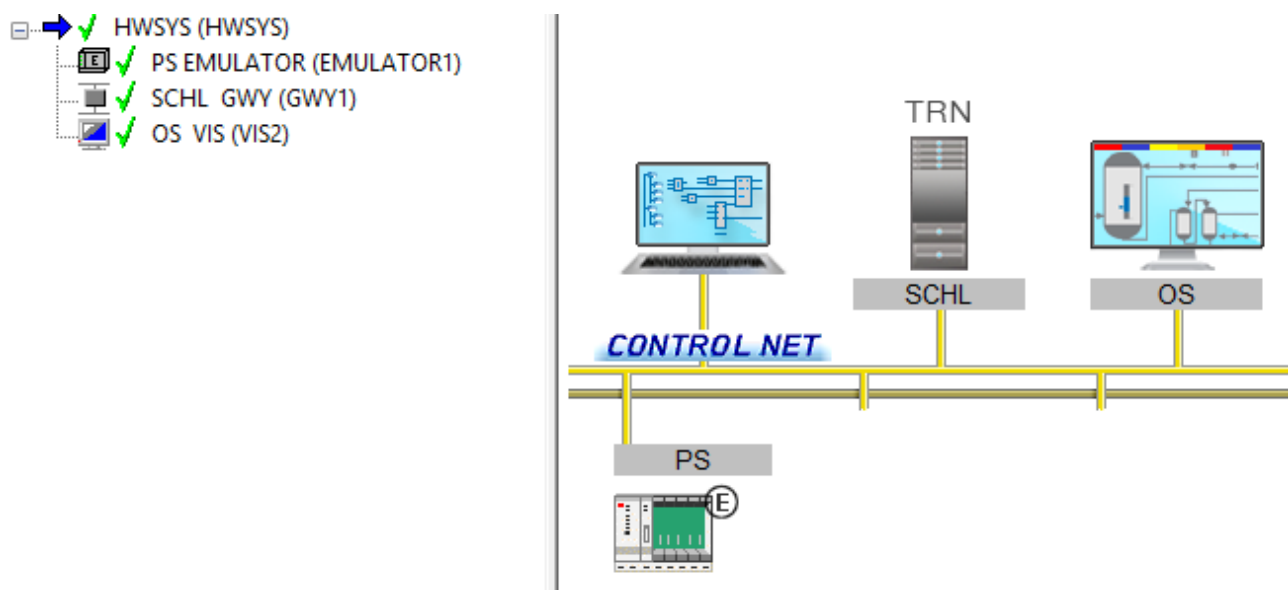


Рис.18 Распределённая структура аппаратных средств

Для возврата в окно конфигурирования программной части необходимо нажать на кнопку «Список переменных» (рис.19)



Рис.19 Кнопка «Список переменных»

4.2. Проверка проекта на наличие ошибок

Вы проделали большую работу и самое главное в инжиниринге убедиться, что всё что вы сделали до этого было сделано правильно. Необходимо регулярно (чем чаще тем лучше) проверять проект на наличие ошибок. После каждого изменения в проекте (создание программы, мнемосхемы с динамическими графическими элементами или написание пользовательского функционально блока) рекомендуется проверка проекта на наличие ошибок. Это поможет избежать накопления возможных неточностей в проекте (необъявленная переменная, указание неправильного типа данных, отсутствие начального значения для переменной и т.д.) и повысит производительность реализации программы т.к. откладывание поиска ошибок в проекте и исправление неточностей осложняет реализацию проекта.

Для проверки проекта необходимо выделить либо часть проекта (проверяется только выделенная область), либо выделить главную ветвь проекта (проект проверяется целиком) и нажать на клавишу «Проверить» (рис.20)

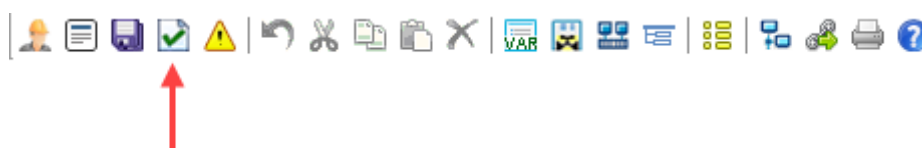


Рис.20 Кнопка проверить.

После проверки на наличие ошибок программа откроет окно в котором будут отображаться возможные ошибки, предупреждения и указания к исправлению ошибок и неточностей в проекте (рис.21)

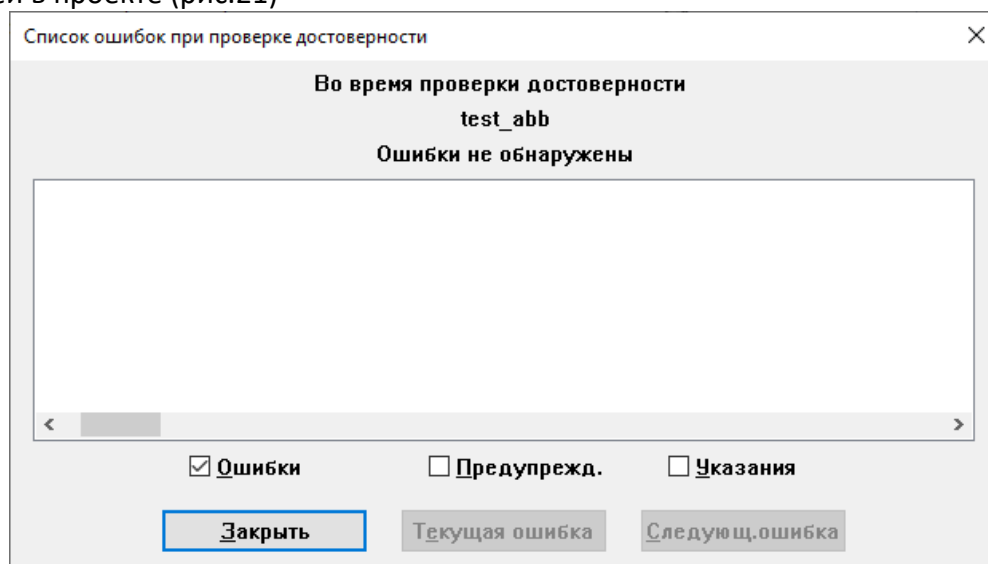


Рис.21 Проверка проекта на правдоподобие

Если все операции проделаны правильно, то система не выдаст ошибок и указаний и при этом цвет проверенного проекта окрасится преимущественно в зелёные цвета как на рис.22.

Если ошибки всё же есть, то достаточно их прочитать и, если описание не понятно, то просто щёлкнуть два раза по тексту ошибки и программа автоматически переместит вас в то место, где происходит сбой компиляции.

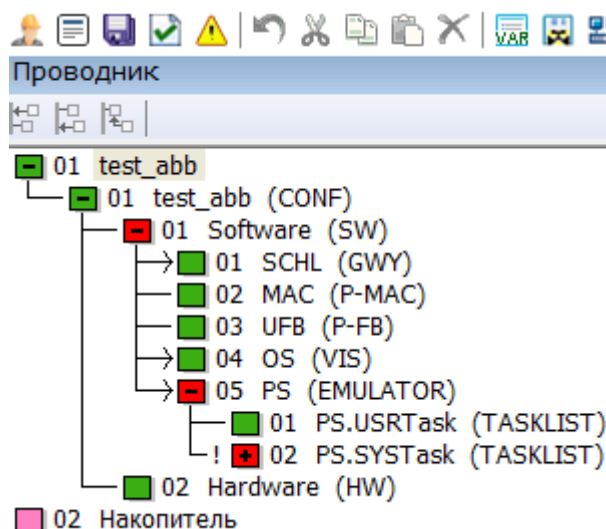


Рис.22 Структура проекта после проверки на наличие ошибок (ошибок не обнаружено)

4.3. Настройка эмулятора контроллера

После того как структура проекта создана, программная и аппаратные части между собой состыкованы остаётся только запустить эмулятор контроллера.

Т.к. все компоненты нашей системы (эмулятор контроллера, инженерная и операторская станции) находятся на одном компьютере, но являются самостоятельными элементами нам необходимо правильно сконфигурировать сеть, для чего необходимо нажать на кнопку «Конфигурация сети» (рис. 23) на панели задач.



Рис.23 Кнопка «Конфигурация сети»

В открывшемся окне указать IP-адреса и ID ресурсов аналогично рис.24.

Аппаратный узел		Ресурс			Линия
тип	имя	тип	имя	ID	IP-адрес 1
Инжен. ПК	Freelance Engineering	D-ES		21	local
EMULATOR	EMULATOR1	D-PS	PS	1	127.0.0.1
GWY	GWY1	D-GS	SCHL	23	127.0.0.1
VIS	VIS2	D-OS	OS	22	127.0.0.1

OK Отмена

Рис.24 Конфигурирование сети

Адрес 127.0.0.1 является локальным IP-адресом любого компьютера. Localhost — в компьютерных сетях, стандартное, официально зарезервированное, доменное имя для частных IP-адресов (в диапазоне 127.0.0.1 — 127.255.255.255). Для сети, состоящей только из одного компьютера, как правило, используется всего один адрес — 127.0.0.1, который устанавливается на специальный сетевой интерфейс «внутренней петли» в сетевом протоколе TCP/IP. При установке соединений в этой вырожденной «сети» присутствует только один компьютер, при этом сетевые протоколы выполняют функции протоколов межпроцессного взаимодействия.

Обратите внимание на столбец ID. ID – это индивидуальный идентификационный номер каждого приложения Freelance которые установлены на компьютере. Чтобы в последствии все программы запустились корректно и можно было загружать проекты с инженерной станции в операторскую станцию ID номера в проекте и в меню настройки Freelance должны совпадать. В нашем случае в программе для операторской станции ID=22 (рис.24). Для настройки ID приложения на компьютере необходимо открыть Пуск – Программы – ABB – Settings и на вкладке «Freelance Operations» ввести значение ID=22. Нажать кнопку «OK» и выйти из окна настроек (рис.25)

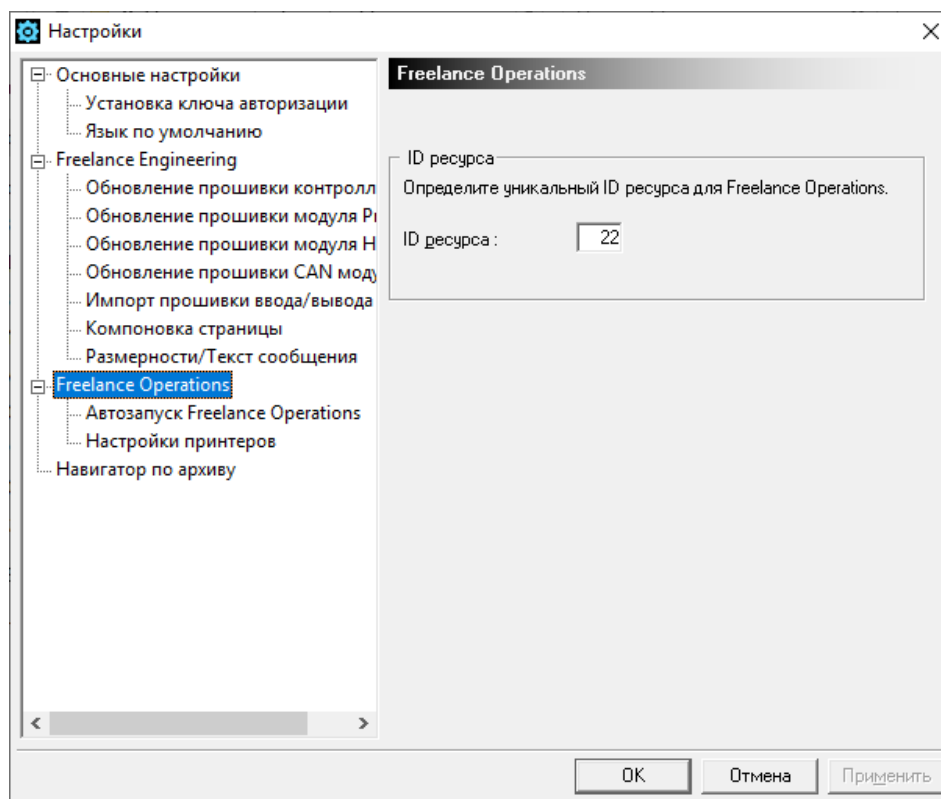


Рис.25 Настройки системы. Ввод ID ресурса операторской станции

Таким образом мы сконфигурировали аппаратную часть нашей системы, присвоили адреса, распределили ресурсы, но так и не запустили эмулятор контроллера на компьютере. Вызвав любой браузер необходимо в адресной строке набрать <http://127.0.0.1:8888/>. В открывшемся окне в поле «Station №» набрать 1 и нажать на кнопку «Initialize Controller».

Если все действия выполнены правильно, то страница обновится до состояния как показано на рис.26. После завершения настройки можете закрыть браузер.

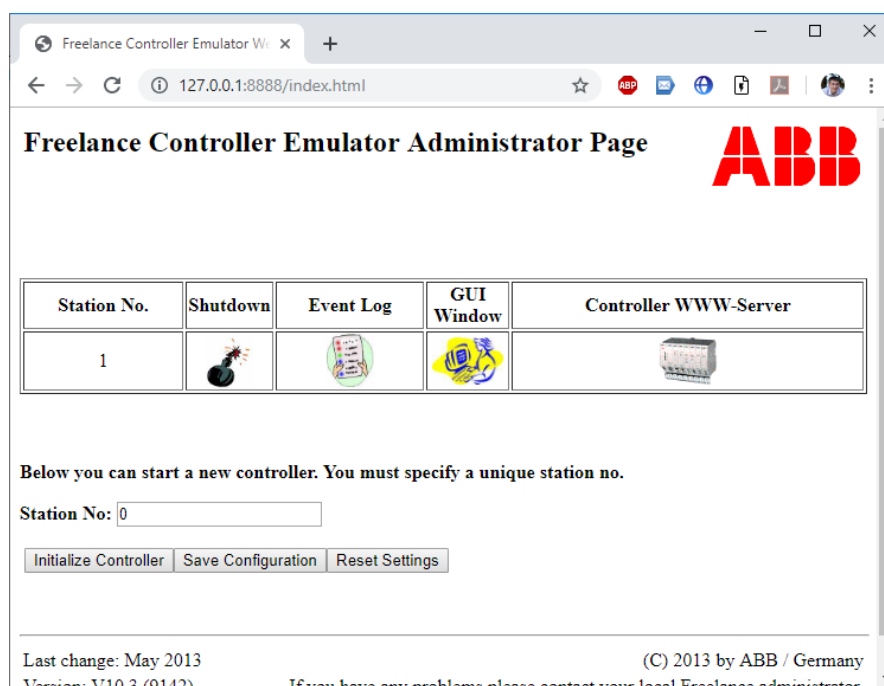


Рис.26 Окно запуска эмулятора контроллера

На данном этапе всё готово к созданию и запуску программы пользователя.

4.4. Реализация простого примера (проверка работоспособности проекта)

Создание пользовательских функциональных блоков и графики достаточно интересный и трудоёмкий процесс, а опробовать работу эмулятора хочется уже сейчас.

Можно запустить простейшую программу и наблюдать как выполняются действия эмулятором. Для этого необходимо:

1. Перейти в дерево проекта и создать в процессовой станции (PS) задачу с циклом 1 секунда;
2. В задачу вставить «Список программ PL»
3. В список программ вставить «FBD-программу»

После проделанных манипуляций проект должен выглядеть как на рис.27

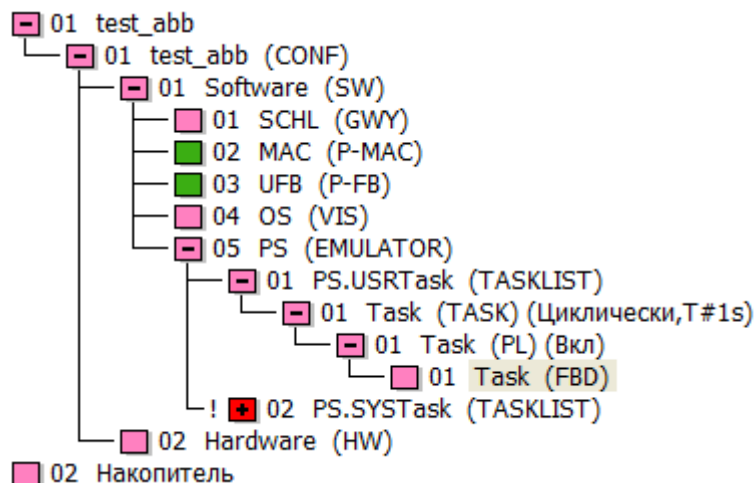


Рис.27 Структура проекта для тестового примера

Необходимо зайти в программу Task (FBD) и в правой части экрана отобразится пустое белое поле. В этом поле необходимо реализовывать программный код. Самый простой пример который может встретиться на практике это обычный счётчик типа $X=X+1$. Каждый цикл эмулятор будет прибавлять 1 к текущему значению X. Т.к. цикл мы задали 1сек, то 1 раз в секунду наше значение X должно увеличиваться на 1.

Нажмите правой кнопкой мыши в любом месте белого поля программы. В открывшемся меню (рис.28)

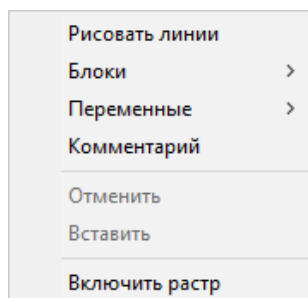


Рис.28 Контекстное меню FBD программы

Выберите «Переменные» - «Читать». После вставки одной переменной в поле нажмите на клавиатуре клавишу Esc для прекращения множественной вставки переменных.

Проделайте аналогичную операцию «Переменные» - «Записать».

Чтобы вставить блок «Сложение» необходимо нажать «Блоки» - «Арифметические» - «Базовая арифметика» - «Сложение (ADD)». После проделанных операций программа должны выглядеть как на рис.29

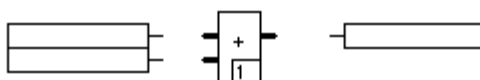


Рис.29 Промежуточный вариант создания простейшей программы

Для соединения пустых переменных с блоком «Сложение» необходимо, не отпуская левую кнопку мыши, тянуть связь от одной ниточки до другой (направление в данном случае не важно).

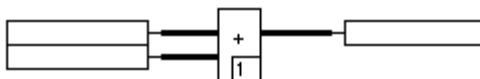


Рис.30

После соединения переменных с блоком сложения требуется вписать переменные в пустые поля. При двойном нажатии мыши по пустой переменной откроется окно. Ввести в окно переменную «X» латинскими буквами (рис.31)

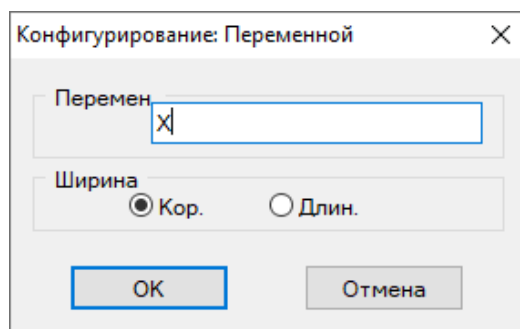


Рис.31 Ввод новой переменной в проект

Т.к. переменная до настоящего момента в проекте не существовала, то программа спросит какого типа должна быть эта переменная. В окне «Вставка новой переменной» (рис.32) выбираем тип переменной REAL.

Т.к. эмулятор контроллера у нас является единственным контроллером в проекте, то система автоматически распределяет нашу новую переменную ресурсу PS. В данном случае распределение демонстрируем в каком контроллере данная переменная будет вычисляться (это поле очень критично для заполнения если у вас в проекте будет несколько контроллеров в одной сети).

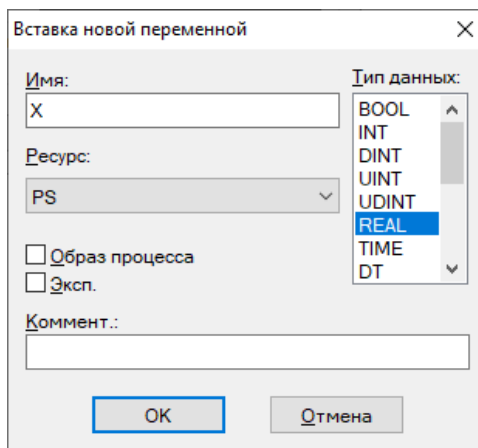


Рис.32 Окно «Вставка новой переменной»

Заполняем программу FBD так как показано на рис.33. **Программа чувствительная к регистру переменных поэтому будьте внимательны с написанием переменных с большой или маленькой буквы.** Если написать в программе «X» и «x», то это будет восприниматься программой как две разных переменных. Обратите также внимание на константу «1.0». Т.к. мы используем переменную типа REAL, то и константа должна быть написана с дробной частью. В противном случае программа не будет компилироваться (при проверке проекта будут показаны ошибки).

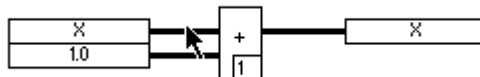


Рис.33 Заполнение пустых ячеек в программе

Программа создана и необходимо проверить проект на наличие ошибок. На данном этапе ошибок быть не должно. Программа должна выглядеть как на рис.34.

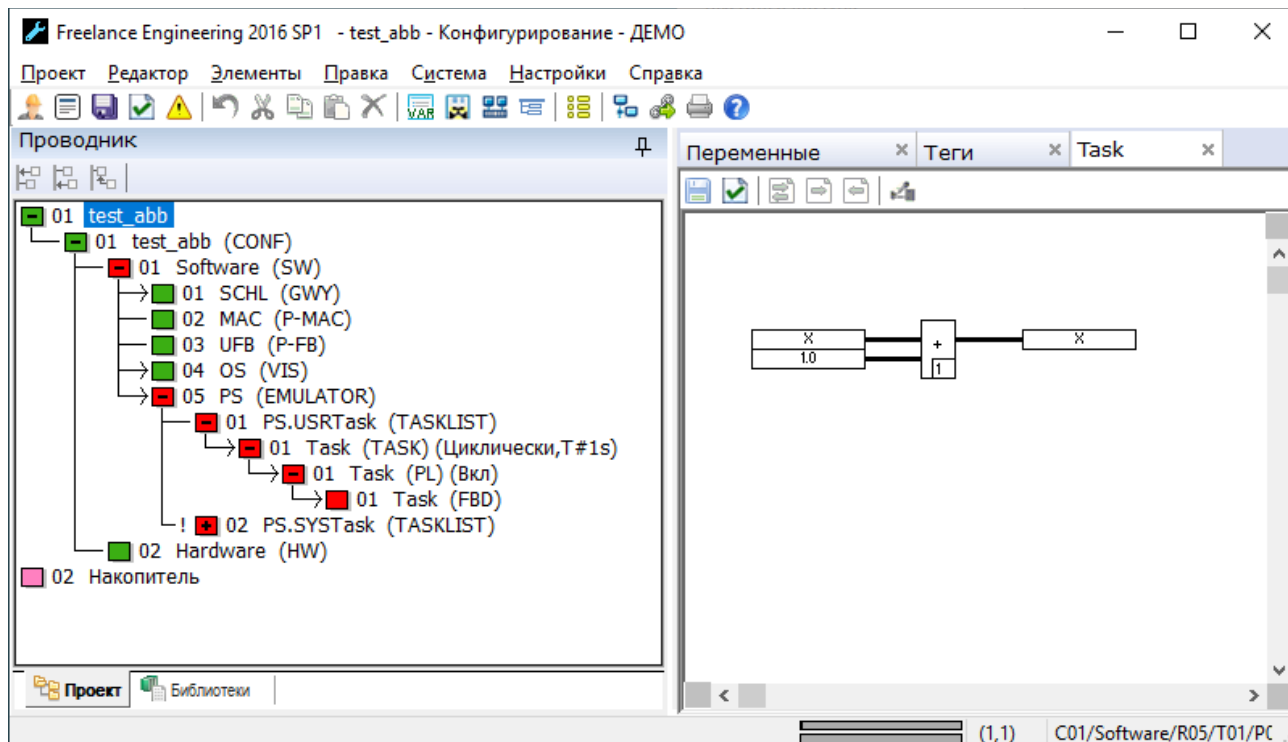



Рис.34 Проект готов к загрузке в эмулятор контроллера

Если ошибок нет, то нажимаем на иконку «Наладка» . После перехода в режим «Наладка» в дереве проекта рядом с операторской станцией и эмулятором контроллера появятся описания их текущего состояния (для операторской станции – «Нет связи» потому что мы её ещё не запускали, а для эмулятора контроллера – «Ошибка версии. Выполнение»). Ошибка версии означает, что крайние изменения, внесённые в программе не прогружены в эмулятор.

Вызвать контекстное меню «05 PS (EMULATOR)» и нажать «Загрузить» - «Вся станция» (рис.35). В открывшемся диалоговом окне подтвердить свои действия.

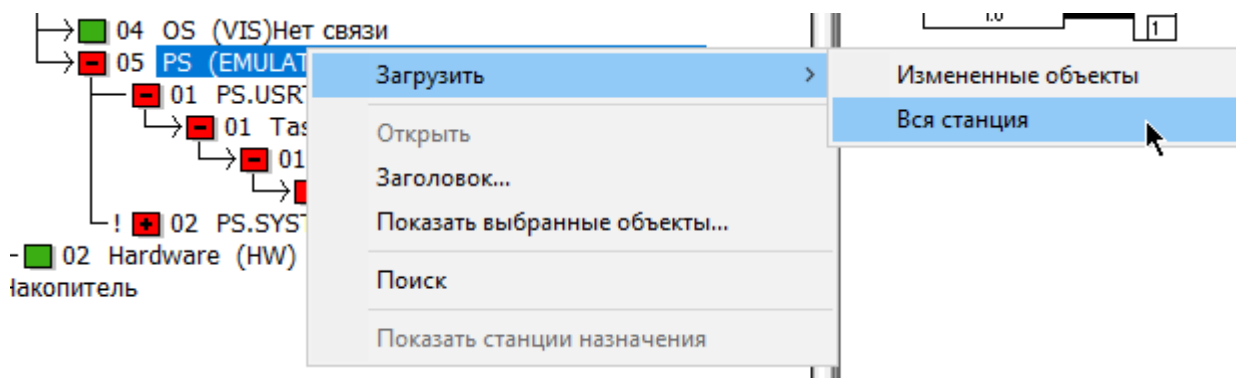


Рис.35 Заливка программы FDB в эмулятор контроллера

Проект должен окраситься в зелёный цвет, а при наведении курсора мыши на переменную «X» вы увидите как меняется текущее значение переменной. Если всё сделано корректно, то переменная будет увеличиться на 1 каждую секунду (рис.36)

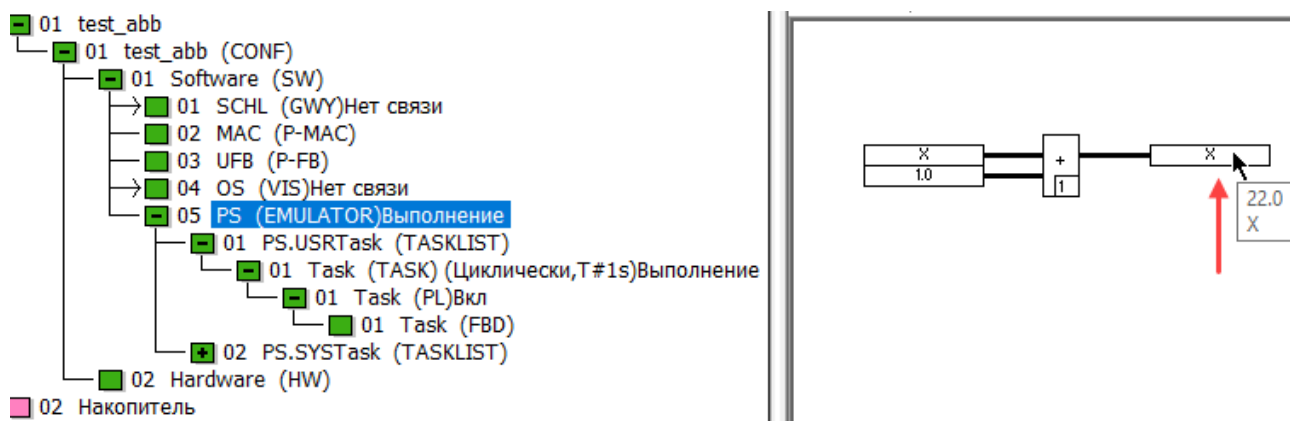


Рис.36 Проект в режиме «Наладка» с работающей в on-line FBD программой

Переменную «X» можно всегда обнулить или задать иное значение. В режиме наладки вызвать контекстное меню переменной и выбрать «Записать значение» (рис.37). В открывшемся окне также надо соблюдать корректность ввода новой переменной, т.к. вводить значение через точку с указанием дробной части – 28.0 или 104.1 и т.д.

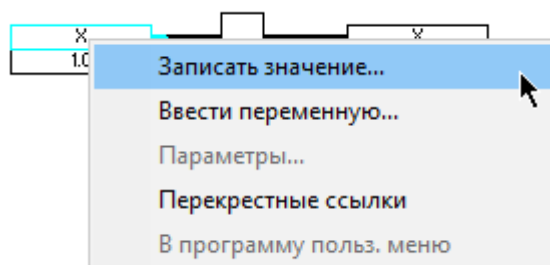


Рис.37 Контекстное меню переменной в режиме «Наладка»

Если всё проделано правильно, то можно переходить к следующему этапу освоения системы.

5. Создание функциональных блоков и программы пользователя

5.1. Функциональный блок апериодического звена

Для того чтобы создать пользовательский функциональный блок (ФБ) необходимо в созданном нами ранее накопителе пользовательских функциональных блоков «UFB [P-FB]» создать следующий уровень «Класс пользовательского функционального блока FB-CLASS» и присвоить ему имя «a_zveno». В созданном классе следующим уровнем вставить «ST-функциональный блок FB-ST» и оставить имя по умолчанию предлагаемое программой «a_zveno_P». Если все операции выполнены верно, то должен получиться трехуровневый список (рис.38)

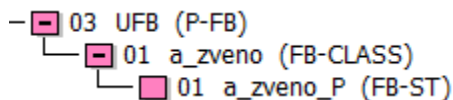


Рис.38 Структура накопителя пользовательских функциональных блоков

Двойным нажатием на «a_zveno_P [FB-ST]» войти в тело программы, создаваемой по умолчанию для каждого ФБ (объявление блока, набор системных переменных, завершение блока). Теперь необходимо написать саму программу реализующую функцию А-звена, для этого сначала необходимо указать переменные, которые будут участвовать в обработке нашего ФБ, указать их тип и будут ли они являться входными, внутренними или выходными. Для этого необходимо войти в ФБ и нажать на кнопку «Переменные пользовательского ФБ» (рис.39)

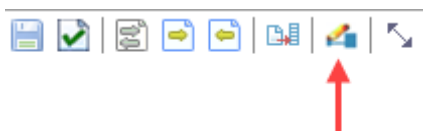


Рис.39 Кнопка «Переменные пользовательского функционального блока»

На экране отобразятся 5 переменных серого цвета (ClassName, TagName, ShortName, LongText, SelState) – это системные переменные, их тип и название на данном этапе нельзя изменить. Для того чтобы создать переменную необходимо 2 раза щелкнуть мышкой по последней пустой ячейке в столбце «Имя». Ячейка станет доступной для редактирования. Введите имя переменной и нажмите «Enter», в открывшемся окне укажите тип хранения переменной (входная – VAR_IN, внутренняя - VAR_DPS или выходная VAR_OUT) и после подтверждения дополнительно укажите тип создаваемой переменной. Всем созданным переменным, кроме выходной, можно присвоить начальные значения.

Для реализации А-звена необходимо набрать перечень переменных следующего вида (рис.40)

Имя	Тип д	Тип хран	Начальное	Мин.	Макс	Парамет	Комментарий
x	REAL	VAR_IN					Значение пар-ра на входе А-звена
k	REAL	VAR_IN	1.0				Коэффициент усиления А-звена
T	REAL	VAR_IN	10.0				Постоянная времени фильтра в сек
CycleTime	INT	VAR_IN	1000				Время цикла ПЛК (эмулятора)
y	REAL	VAR_OUT					Значение пар-ра на выходе А-звена
out	REAL	VAR_DPS					Значение пар-ра на выходе А-звена
Ta	REAL	VAR_DPS					Постоянная времени фильтра в мс
ClassName	TEXT	PARA_VIS					
TagName	TEXT	PARA_VIS					
ShortText	TEXT	PARA_VIS					
LongText	TEXT	PARA_VIS					
SelState	BOOL	PARA_VIS					

Рис.40 Окно «Переменные пользовательского функционального блока»

Обратите внимание на переменную CycleTime (время цикла исполняемой программы), которая задается в мс (миллисекунды). Она должна соответствовать времени цикла нашей программы, иначе ФБ будет работать некорректно.

Не забывайте вносить комментарии для переменных – это повысит читаемость программы и скорость исправления возможных ошибок в дальнейшей работе.

Необходимо перейти обратно в тело программы и написать функцию А-звена (рис.41). Все ранее записанные переменные автоматически представлены в заголовке ST программы (закрашены серым фоном)

Проверить программу на наличие ошибок. Если все действия выполнены правильно, то система ошибок не выдаст. Особое внимание программированию на ST следует уделять таким символам как запятые, точки, точка с запятой и двоеточие.

FUNCTION_BLOCK a_zveno_P
<pre> VAR_INPUT x : REAL; k : REAL; T : REAL; CycleTime : INT; END_VAR VAR_OUTPUT y : REAL; END_VAR VAR (* VAR_DPS *) out : REAL; Ta : REAL; END_VAR (* PARA_VIS ClassName : TEXT; TagName : TEXT; ShortText : TEXT; LongText : TEXT; SelState : BOOL; END_VAR *) </pre>

```

Ta := T*1000.0; (*переводим постоянную времени из секунд в миллисекунды*)
out := (k * TO_RE(CycleTime) * x / Ta) + (1.0 - TO_RE(CycleTime) / Ta) * out; (*функция А-звена*)
y := out; (*выход функционального блока*)
END_FUNCTION_BLOCK

```

Рис.41 Тело программы «А-звено»

Функциональный блок, реализующий функцию А-звена готов. Выйти из программы в список проекта. Проверить проект на наличие ошибок.

Примечание. Переменную CycleTime можно было сразу сделать типа REAL, но в данном случае показано также как в ST записать преобразование переменной из одного типа в другой (из INT в REAL)

5.2. Функциональный блок регулирующего клапана

Создать класс пользовательского функционального блока «drive», в класс блока вставить «ST-функциональный блок FB-ST». Названия, предложенные программой оставить по умолчанию. Если все операции проделаны верно, то должен получиться трехуровневый список (рис.42).

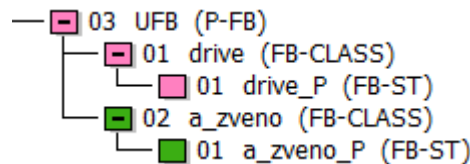


Рис.42 Структура накопителя пользовательских функциональных блоков

Двойным нажатием на «drive_P [FB-ST]» войти в тело программы и указать переменные, которые будут участвовать в обработке нашего функционально блока, указать их тип и будут ли они являться входными, внутренними или выходными. Для реализации реверсивного двигателя необходимо создать перечень следующих переменных (рис.43)

Имя	Тип дан.	Тип хране	Нач. зн	Ми	Ма	Па	Комментарий
UP_in	BOOL	VAR_IN					Входное значение "Больше"
DN_in	BOOL	VAR_IN					Входное значение "Меньше"
Tmv	INT	VAR_IN	25				Время хода исп-го механизма
idt	INT	VAR_IN	1000				Время цикла
MU	REAL	VAR_OUT					Выходное значение УП
MU	REAL	VAR_DPS	50.0				Внутренняя переменная УП
UP	BOOL	VAR_DPS	FALSE				Внутренняя переменная "Больше"
DN	BOOL	VAR_DPS	FALSE				Внутренняя переменная "Меньше"
ClassName	ТЕКСТ	PARAMETER					
TagName	ТЕКСТ	PARAMETER					
ShortText	ТЕКСТ	PARAMETER					
LongText	ТЕКСТ	PARAMETER					
SelState	BOOL	PARAMETER					

Рис.43 Кнопка «Переменные пользовательского функционального блока»

После чего перейти обратно в тело программы и написать программу реверсивного двигателя. Проверить программу на наличие ошибок. Если все действия выполнены правильно, то тело программы будет выглядеть следующим образом (рис.44):

```

FUNCTION_BLOCK drive_P
VAR_INPUT
    Open : BOOL;
    Close : BOOL;

```

```

Treg : INT;
CycleTime : INT;
END_VAR

VAR_OUTPUT
UP : REAL;
END_VAR

VAR (* VAR_DPS *)
UP_dps : REAL;
Open_dps : BOOL;
Close_dps : BOOL;
END_VAR

(*
PARA_VIS
ClassName : TEXT;
TagName : TEXT;
ShortText : TEXT;
LongText : TEXT;
SelState : BOOL;
END_VAR
Open_dps := Open;
Close_dps := Close;

IF (Close_dps AND NOT Open_dps) THEN
UP_dps := UP_dps - 100.0*TO_RE(CycleTime)/(TO_RE(Treg)*1000.0);
END_IF;

IF (Open_dps AND NOT Close_dps) THEN
UP_dps := UP_dps + 100.0*TO_RE(CycleTime)/(TO_RE(Treg)*1000.0);
END_IF;

IF UP_dps > 100.0 THEN UP_dps := 100.0;
ELSIF UP_dps < 0.0 THEN UP_dps := 0.0;
END_IF;

UP := UP_dps;
END_FUNCTION_BLOCK

```

Рис.44 Тело программы «Регулирующий клапан»

Функциональный блок, реализующий функцию реверсивного двигателя готов. Выйти из программы в список проекта. Проверить проект на наличие ошибок.

5.3. Программа пользователя

Мы создали пользовательские функциональные блоки для А-звена и регулирующего органа. Теперь можно приступить непосредственно к созданию программы, имитирующей работу одноконтурной системы автоматического регулирования (САР), структурная схема которой изображена на рис.8.

Вы уже умеете создавать задачи в проекте (задача TASK была создана для примера $X=X+1$). Сейчас необходимо создать новую задачу с циклом 100 миллисекунд (мс). Присвойте имя новой задаче asutp_task (рис.45)

Рис.45 Создание пользовательской задачи

В созданной задаче следующим уровнем создать «Список программ PL». Оставить имя списка по умолчанию (asutp_task). В списке программ уровнем ниже создать FBD программу и также оставить имя данное по умолчанию (asutp_task) см. рис.46.

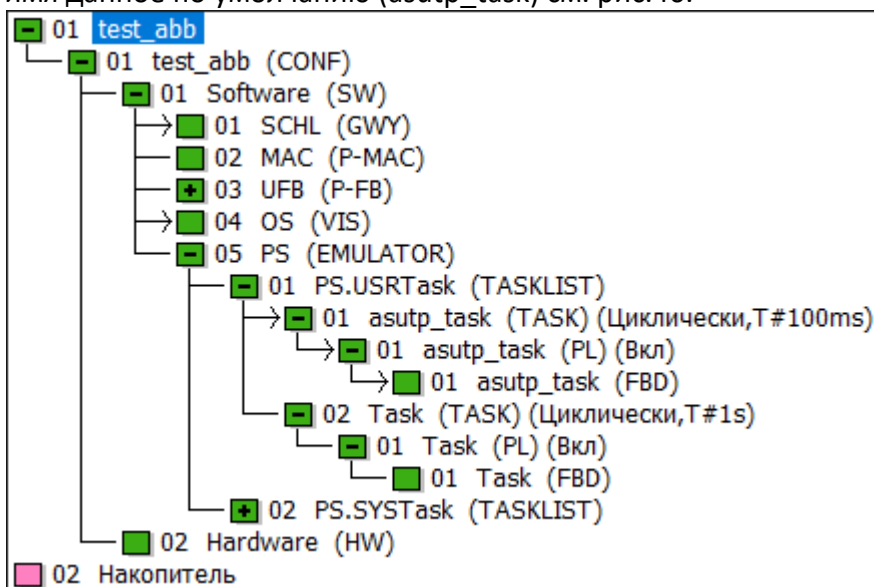


Рис.46 Структура процессовой станции

Откройте программу «asutp_task (FBD)». В открывшемся пустом окне необходимо создать структуру одноконтурной системы автоматического регулирования как на рис.47.

При создании структуры обратите внимание на последовательную нумерацию функциональных блоков внутри программы (прямоугольник с цифрой внизу каждого функционального блока). Важно чтобы нумерация ФБ была слева направо, сверху вниз. В противном случае эмулятор будет выполнять программу согласно нумерации блоков, что можем привести к ошибочным вычислениям. Чтобы поправить нумерацию необходимо нажать клавишу Cntrl и левой кнопкой мыши нажать на порядковый номер вычисления блока. После того как число исправлено нажать на Enter. Остальные блоки автоматически поменяют свою нумерацию. Меняйте нумерацию последовательно слева-направо.

Для рисования сложных (ломаных) линий (от объекта к объекту с поворотами) необходимо одновременно зажимать клавиши Cntrl+Shift и правой кнопкой мыши рисовать линию от одного объекта до другого. Система самостоятельно нарисует извилистую линию в зависимости от того как располагаются объекты на схеме.

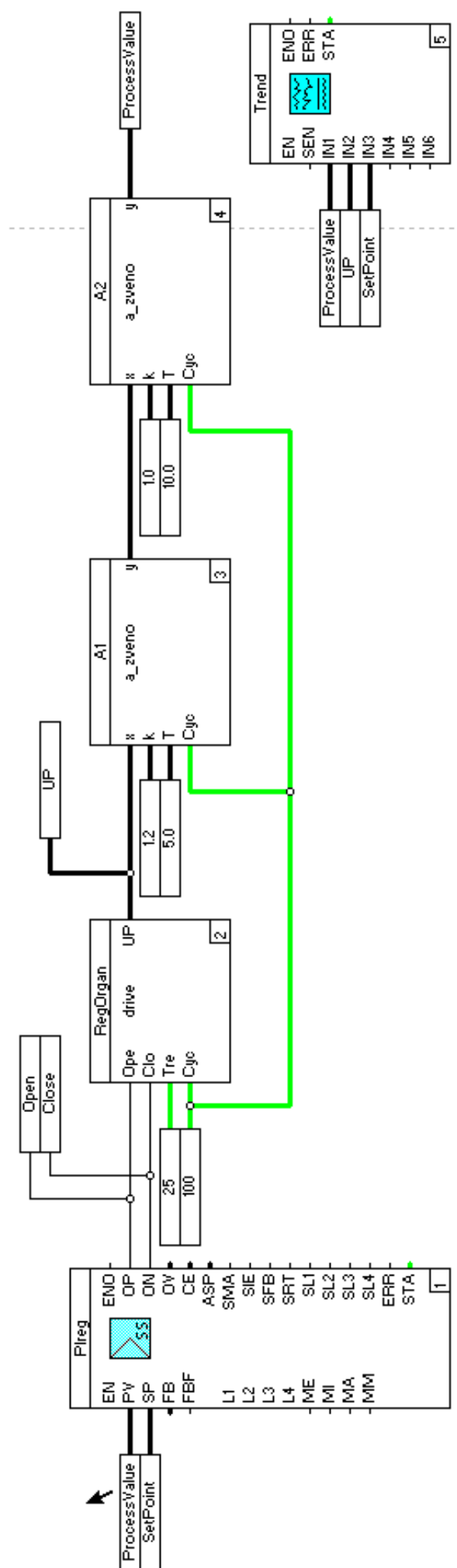


Рис.47 Реализация одноконтурной САУ на языке FBD

Для вставки экземпляров созданных ранее пользовательских функциональных блоков необходимо правой кнопкой мыши нажать в любом месте экрана и выбрать «Блоки» - «Польз. Функциональные блоки». Программа предложит на выбор «a_zveno» и «drive».

- для регулятора «Блоки» - «Регулятор» - «Импульсный регулятор, стандартный (C_SS)»;
- для блока сложения «Блоки» - «Арифметические» - «Базовая арифметика» - «Сложение» (ADD);
- для блока сбора трендов «Блоки» - «Сбор данных» - «Тренд» (TREND).

Каждому вставленному в программу блоку необходимо задать имя, однозначно определяющее его в проекте (имена функциональных блоков не могут повторяться для разных объектов), для этого двойным щелчком мыши по функциональному блоку откройте окно его настроек и в поле «Имя» задайте оригинальное имя ФБ в соответствии с рис.47. Посмотреть все используемые ФБ в системе можно при нажатии «Система» - «Список тэгов» (рис.48)

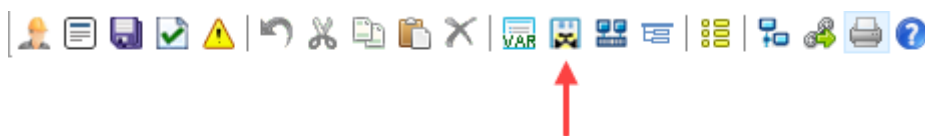


Рис.48 Кнопка «Список тэгов проекта»

Для того чтобы наш регулятор правильно работал необходимо его сконфигурировать должным образом. В первом окне параметров (рис.49) задается диапазон регулирования регулируемой величины (PV), размерность регулируемого параметра, возможность изменения задания непосредственно из операторской станции, а также в поле «Сообщения» можно задать значения уставок срабатывания сигнализации (как предупредительной, так и аварийной) при отклонении регулируемого параметра (более подробно см. раздел «Сигнализация»). **Внимательно изучите все три окна настроек (рис.49-51). Необходимо заполнить поля в вашей программе в точном соответствии с указанными ниже полями. В противном случае регулирования может не получиться.**

Параметры: Стандартный шаговый регулятор C_SS (1/3)

Основные данные

Имя: **PIreg** Кр. текст: Дл. текст: Обработка: ☒ Последов.: 1

Начало диапазона: **0.0** Конец диапазона: **100.0** Размерность: °C

Внутреннее задание: 50.0 ☒ Доступ

Выход: 0.0 ☒ Доступ

Сообщения

№	Тип	Знач.	Доступ	Гист.	Прик	Указ	Текст сообщ.
1	HH	95.0	<input type="checkbox"/>	0.3	1	-	Авар.выс
2	H	90.0	<input type="checkbox"/>	0.3	2	-	Высока
3	L	10.0	<input type="checkbox"/>	0.3	2	-	Низка
4	LL	5.0	<input type="checkbox"/>	0.3	1	-	Авар.низ

OK Отмена Сохранить Сброс Проверить Справка << >>

Рис.49 Окно параметров регулятора (1/3)

В окне параметров №2 (рис.50) можно задать настроечные параметры регулятора, включить или выключить составляющие регулятора (П-, И-), а также указать, как должен начать работать регулятор при старте проекта (поле «Режим работы»).

Параметры: Стандартный шаговый регулятор C_SS (2/3)

ПИД параметры

☒ П-звено Д-составл. Д-составляющ-я от Кп: 0.6
☒ И-звено ☐ Вкл ☐ PV ☒ CE Ти: T#6с
☒ Инвертировать ☐ полож. П-составляющ-я от Тд: T#0с
 ☐ отриц. ☐ PV ☒ CE Кд: 1.0
☐ Внешняя обратная связь ☒ Выкл
 Рабочая точка: 50.0 Тсинхр T#30с

Ограничение рабоч.режима

☐ Внутр. зад. ☐ Ручн.вых.
☐ Внеш. зад. ☐ Авт. вых.

Синхрон.Ручного с Авто

☒ Нет
☐ Настр. на старое зад.
☐ Настр. на текущее зад.

Рабочие режимы

☐ Внутр. ☒ Внешн.
☐ Ручной ☒ Авто.

OK
 Отмена
 Сохранить
 Сброс
 Проверить
 Справка
 << >>

Рис.50 Окно параметров регулятора (2/3)

В окне параметров №3 (рис.51) необходимо указать время хода исполнительного механизма, установить параметры импульсного алгоритма.

Параметры: Стандартный шаговый регулятор C_SS (3/3)

Функция градиента (время на диапазон)

	Задание	Выход
увелич:	T#0s	T#0s
пониж:	T#0s	T#0s

☒ Действ. в Руч. ☒ Действ. в Руч.

Шаговый алгоритм

Мин.длит.импульса: T#200ms

Мин.длительн.паузы: T#400ms

Запазд. внешн.обр.св.: T#100ms

Константы

Верхн. предел зад.: 100.0

Нижн. предел зад.: 0.0

Верхн.пред.вых.: 100.0

Нижн.пред.вых.: 0.0

Зона нечувст.%от СЕ: 0.1

Исполнительное устройство

Время до откр.: T#25s

Время до закр.: T#25s

Мак.дин.коэф.: 3.0

Контроль Вкл.: ☐

Эффективн.пределов выхода

☐ Активны в Ручн. и Авто.

☐ Активны в Авто. режиме

☒ Неактивны

Отслеживание задания

☒ Внут.зад.отслеж.текущ.зад.

☐ Внутр.зад. как настроено

OK Сброс

Отмена Проверить

Сохранить Справка

<< >>

Рис.51 Окно параметров регулятора (3/3)

После настройки всех ФБ необходимо создать переменные (согласно рис.47) и связать все объекты в единую САУ. К глобальным переменным, созданным в проекте можно обратиться по адресу «Система» - «Список переменных» (рис.52).

Имя	Комментарий	Типе	Рес.
		▼	▼
X	Тестовая переменная для X+1	REAL	PS
Close	Команда от ПИ-рег-ра - ЗАКРЫТЬ	BOOL	PS
Open	Команда от ПИ-рег-ра - ОТКРЫТЬ	BOOL	PS
ProcessValue	Регулируемая величина (темп-ра)	REAL	PS
SetPoint	Уставка для регулирования (т-ра)	REAL	PS
UP	Указатель положения рег.органа	REAL	PS

Рис.52 Список глобальных переменных

При создании переменной программа автоматически попросит вас указать тип переменной и, если это необходимо, использовать промышленную память (используется для резервируемых контроллеров, при этом к имени переменной слева дописывается знак «@». К переменной можно обращаться как со знаком «@» так и без него). В зависимости от типа переменной её входное/выходное значение примет определенный цвет, согласно рис.53.

















Тип данных/ состояние обработки	Цвет	Отобра- жение	Пример
BOOL	черный	узкая	
BYTE	серый	широкая	
DINT	зеленый	широкая	
DT	темно-желтый	широкая	
DWORD	розовый	широкая	
INT	светло-зеленый	широкая	
REAL	черный	широкая	
TIME	светло-желтый	широкая	
UDINT	коричневый	широкая	
UINT	бирюзовый	широкая	
WORD	темно-синий	широкая	
STRING	черный	широкая	
STRUCT	черный	широкая	
Состояние ошибки	красный	узкая	
Выделенные объекты	бирюзовый		
Не подключено	черный	узкая	

Рис.53 Соответствия типов переменных различным цветам

Программа, реализующая одноконтурную систему автоматического регулирования готова. Выйти из программы в список проекта. Проверить проект на наличие ошибок. Уже сейчас можно зайти в режим «Наладка» и убедиться, что система работает (значения переменных будут отображаться при наведении курсора мыши). Данный режим не совсем удобен для настройки регуляторов и требуется создание визуальной оболочки (SCADA системы) чтобы видеть как меняется значение регулируемой переменной при изменении настроек регулятора или изменении уставки.

6. Создание интерфейса оператора (визуализация процесса)

6.1. Создание мнемосхемы

Все мнемосхемы должны находиться внутри операторской станции в структуре проекта. Необходимо вставить в OS (VIS) следующий уровень «Мнемосхема FGR». Название оставить предложенное системой по умолчанию - OS.

Необходимо реализовать следующую мнемосхему (рис.54) которая состоит из статических и динамических элементов.

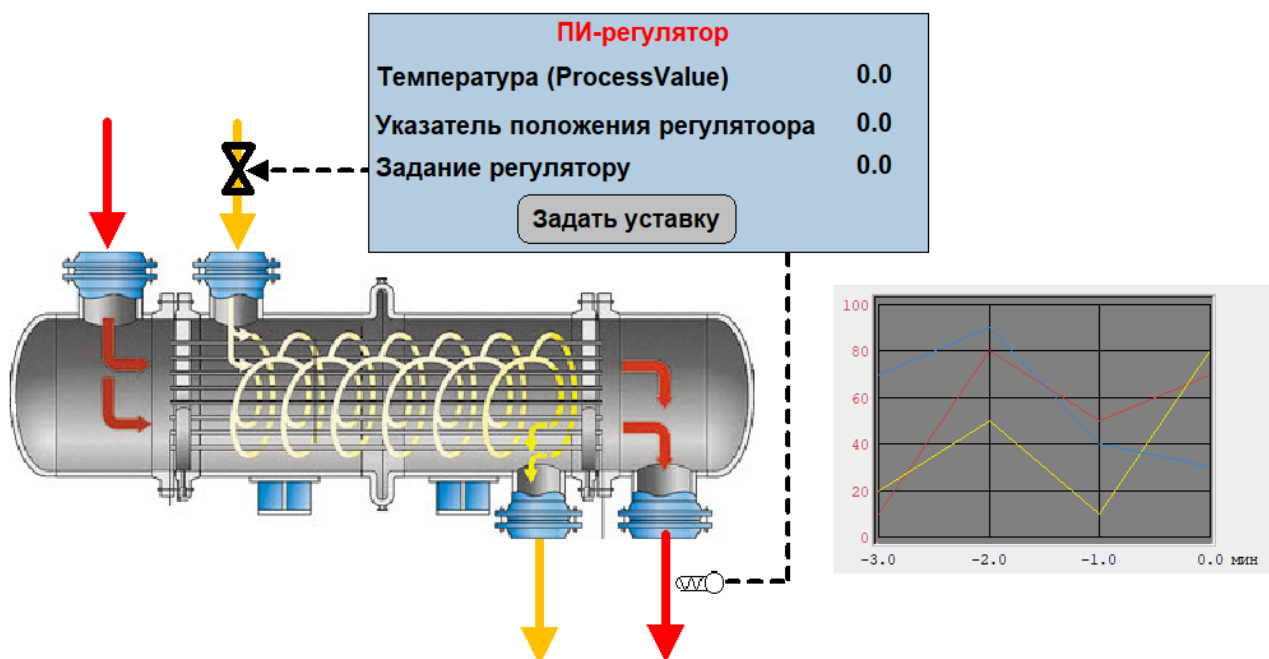



Рис.54 Мнемосхема управления технологическим процессом

Рисование статических объектов во Freelance Engineering не составляет особого труда, наиболее часто возникают вопросы с динамическими объектами, рассмотрим некоторые из них:

1) Создание кнопки

Для создания кнопки задания уставки (изменение задания регулятору) необходимо нажать соответствующую иконку «Кнопка»  на панели графического редактора (открывается вместе с открытием мнемосхемы) и щелчком мыши на мнемосхеме указать первую точку привязки, второй щелчок мыши производится в произвольном месте окна. Программа откроет окно настроек кнопки и в поле «Текст» необходимо написать «Уставка», после чего нажать кнопку «Действие» и в открывшемся окне (рис.55) в пункте «Тип действия» выбрать «Записать переменную», ниже в пункте «Записать переменную» нажать на клавишу «F2» и в открывшемся окне «Выбрать переменную/компонент» указать переменную «SetPoint». Сохраните изменения.

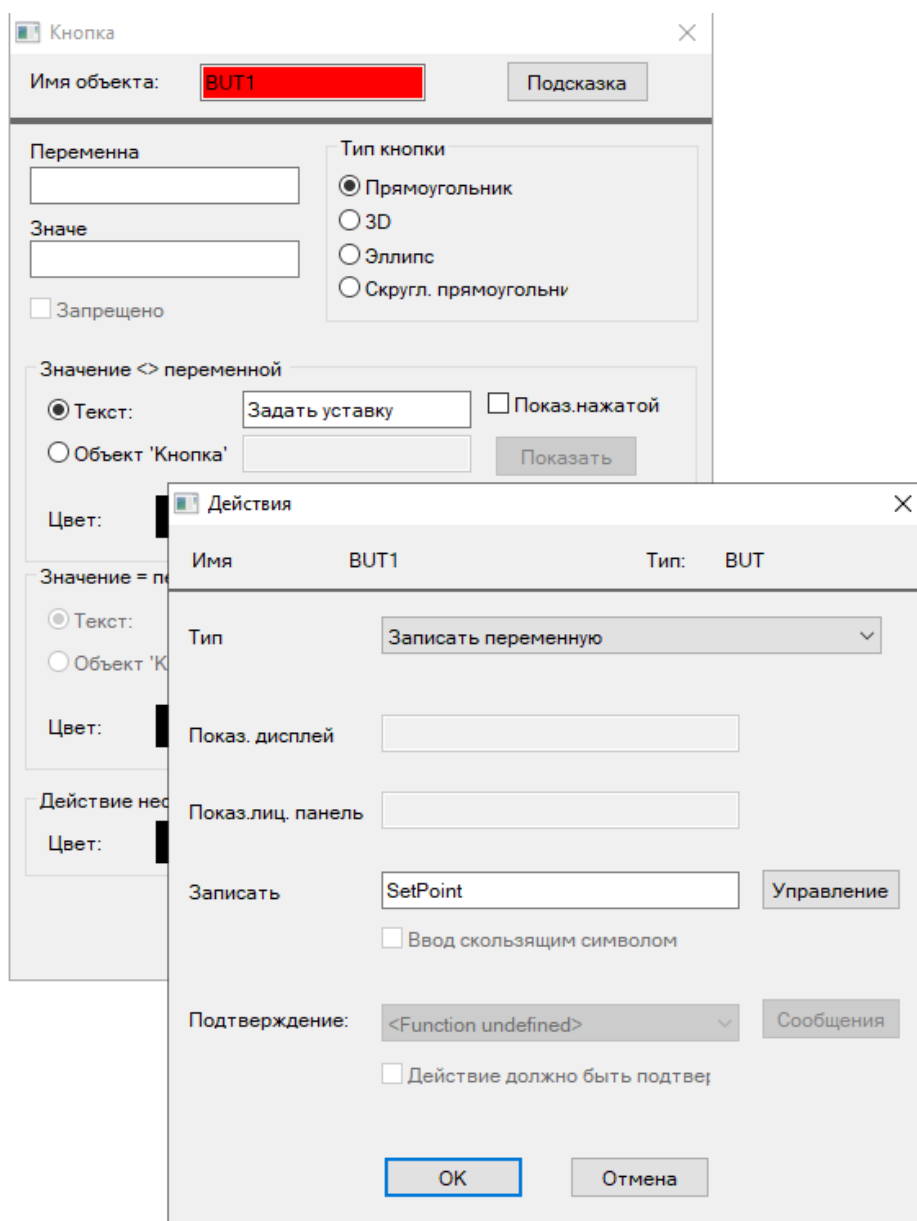



Рис.55 Панель настроек объекта «Кнопка»

2) Создание окна трендов

Для создания окна трендов нажать соответствующую иконку  на панели графического редактора и щелчком мыши на мнемосхеме указать первую точку привязки, второй щелчок мыши производится в произвольном месте окна. Программа откроет окно настроек окна трендов, в котором необходимо указать переменную отображения. Для этого щелкнуть по кнопке «Вставить», в поле «Переменная» нажать клавишу «F2» и выбрать переменную из списка глобальных переменных проекта. Для правильного отображения необходимо определить диапазон значений и для удобства чтения добавить комментарий о переменной (рис.56).

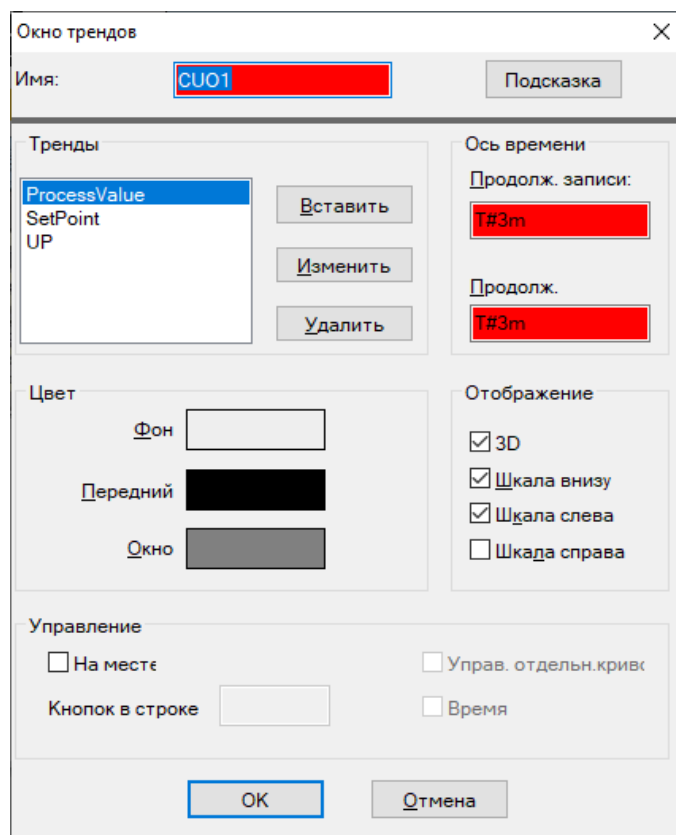



Рис.56 Панель настроек объекта «Кнопка»

Обратите внимание на продолжительность записи и отображения информации. В данных полях введите временной диапазон от 5 до 20 минут. Также отследите чтобы были правильно заданы диапазоны отображения тренда (в нашем случае для всех трёх переменных диапазон должен быть от 0 до 100).


3) Вставка рисунка

Для вставки не мнемосхему готового рисунка нажмите на панели инструментов иконку «Рисунок»  и нажатием на экране укажите, где будет точка привязки рисунка. В открывшемся окне нажмите на кнопку «Импорт» и укажите путь к нужному файлу.

4) Изменение фона мнемосхемы

Нажав правой кнопкой мыши на пустом месте мнемосхемы вызовется контекстное меню, в котором необходимо выбрать «Цвет фона».

5) Вставка цифрового значения переменной

Нажать кнопку «Алфавитно-цифровой дисплей» . В открывшемся окне в поле «Показать» нажать на F2 и выбрать переменную которую хотите отображать. Также для правильного отображения переменной необходимо понимать её диапазон изменения. Если вы знаете, что предел значений составляет от 0.0 до 100.0, то необходимо в окне «Параметры» «Алфавитно-цифрового дисплея» перейти на вкладку «Отображение» и в окне «Формат» нажать левой кнопкой мыши после чего на клавиатуре нажать F2. Откроется окно в котором будут показаны различные представления по отображению переменных в примерно таком виде f.fffff. Каждая буква f представляет собой число в котором будет отображаться цифровое значение. Т.е. для отображения числа 100.0 необходимо создать пользовательский тип fff.f (рис.57). Данные действия необходимо проделать для всех дисплеев где отображаются числовые значения переменных.

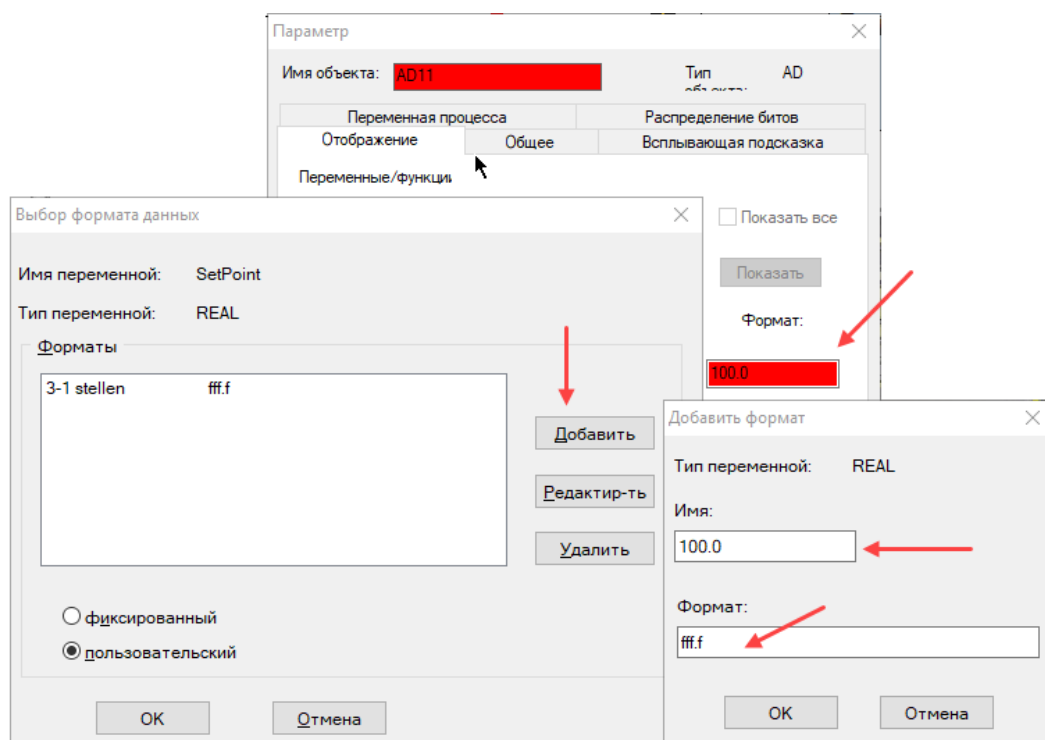


Рис.57 Ввод корректного отображения переменной на экране оператора

6.2. Загрузка мнемосхемы во Freelance Operations

После того как все действия выполнены и проект не имеет ошибок можно прогрузить мнемосхему в программное обеспечение оператора. Для этого необходимо открыть программу Пуск – Все программы – ABB - Freelance Operations. Программа запустится. Теперь программу нужно свернуть. Чтобы свернуть (не закрыть, а свернуть) программу необходимо в панели инструментов Freelance Operations нажать «Сервис» - «Система» в открывшемся окне нажать на кнопку «Операционная система».

Если вы правильно установили IP и ID адреса (см. главу 4.3), то во Freelance Engineering вы увидите, что система обнаружила операторскую станцию. Аналогично действиям по прогрузке эмулятора необходимо загрузить операторскую станцию OS в программу Freelance Operations. Вызвать контекстное меню «OS (VIS)» и нажать «Загрузить» - «Вся станция».

Откройте окно Freelance Operations и нажмите на кнопку «Мнемосхема» (рис.58)

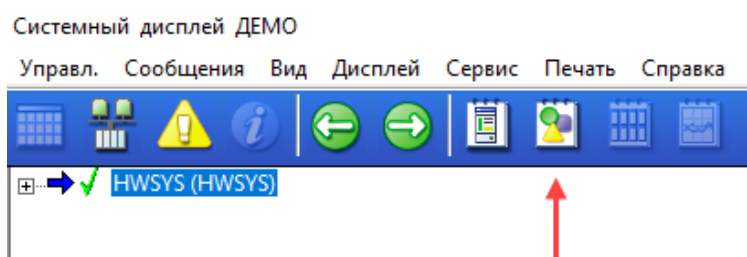


Рис.58 Кнопка «Мнемосхема»

Справа откроется небольшое окно с перечнем мнемосхем. Т.к. мы создали только одну мнемосхему OS. Двойным нажатием на OS перейдите на мнемосхему. Откроется созданная вами мнемосхема. Все числовые значения будут иметь нулевые значения т.к. при загрузке проекта задание у регулятора было равным нулю. Чтобы поменять задание регулятору нажмите на кнопку «Задать уставку» и введите значение от 0.0 до 100.0 (т.к. именно в этом диапазоне предполагается, что мы регулируем технологический процесс).

Вы увидите как начинают меняться значения переменных, а также в окне графика появится отображение трендов (рис.7).

Мнемосхема готова, вы, как оператор, можете управлять технологическим процессом, а также добавлять, менять, конфигурировать, отлаживать процесс автоматического регулирования.

7. Вместо послесловия

Данное пособие изначально планировалось как помощь в освоении системы Freelance инженерам тепловых электрических станций. В последствии пособие было доработано для студентов кафедры АСУТП НИУ «МЭИ» (на кафедре имеется лабораторный стенд с контроллером, модулями ввода/вывода и операторскими станциями). Пособие помогает пройти путь от создания нового проекта, до его загрузки в контроллер (в нашем случае эмулятор), а также загрузки в SCADA систему. Для более тщательного изучения системы необходимо гораздо большее количество часов практики, а также постоянное изучение инструкций и описаний от ООО «АББ».

Если что-то не получилось, то на сайте www.sharovin.ru в разделе «Лабораторные работы Freelance» выложен пример полностью реализованный по данному методическому пособию. Это поможет вам более детально разораться со всеми вопросами и сделать процесс освоение материала более приятным.

Если вам понравилось данное пособие или вы хотите высказать автору предложения или пожелания, то можно писать на e-mail sharovin@yandex.ru