

## Рабочая тетрадь к курсу Systeme Platform «SEP-11-el. Базовый курс разработки» (Разработка проекта и обслуживание систем автоматизации на базе Systeme Platform)

Версия документа: 2023-02-15



## Оглавление

Введение	5
Установка компонентов Systeme Platform	7
Требования к рабочему месту	8
Настройки системы	8
Об АСУ ТП	9
O Systeme Platform	. 10
Архитектура	. 10
1. O SePlatform.Server	. 11
Архитектура	. 11
Работа с SePlatform.Server	. 13
Сервисные приложения	. 13
Конфигуратор	. 13
Статистика	. 14
Просмотрщик лога кадров	. 15
2. Работа с SePlatform.DevStudio	. 16
Настройка развёртывания проекта	. 23
Построение и развёртывание проекта	. 23
3. Знакомство и работа с отладочными инструментами SePlatform.Tools	. 25
Работа с SePlatform.OpcExplorer	. 25
Работа с сигналами. Подключение к OPC DA Server	. 25
Подключение к OPC UA Server	. 29
4. Установка и настройка серверной части на Linux. Создание узла для экземпляра SePlatform.Server на	
Linux в SePlatform.DevStudio.	. 31
5. Конфигурирование SePlatform.Domain	. 41
Конфигурирование SePlatform.Domain на OC Windows	. 41
Конфигурирование SePlatform.Domain на ОС Linux	. 46
6.Подключение к OPC UA Server Linux машины	. 52
7. Модификация проекта SePlatform.DevStudio	. 55
Работа с атрибутами	. 55
Работа с логикой	. 58
8. Работа с компонент Systeme Platform платформы – SePlatform.Historian	. 63
9. Работа с событиями	. 73
Настройка генерации событий в SePlatform.DevStudio	. 73
Просмотр событий	. 74
8. Модификация проекта SePlatform.DevStudio (часть 2)	. 74
Передача данных между объектами (ссылки)	. 74





2

	Подключение к исполняющему компоненту через DeveloperStudio	76
	Добавление внешних исполняющих компонентов и реализация передачи данных между машинами.	77
	Объектно-ориентированный подход	82
	Использование типов	82
	Использование ссылок в типах	84
	Применение аспектов. Использование сокетов	89
9.	Наследование SePlatform.HMI	100 106
	Создание проекта	107
	Добавление экранной формы	107
	Добавление элементов	109
	Добавление функций	111
	Добавление внешних переменных	112
	Работа с элементами АР	115
	Добавление элементов АР определённого вида	115
	Добавление источников данных	120
	ApSource_Main	121
	Использование глобальных объектов. Каскадирование источников	121
	Типизация	123
	Создание демонстрационного проекта в SePlatform.HMI	126
	Создание типа с датчиком	126
	ApSource_Main (Global)	127
	info	131
	Добавление цветовой индикации	133
	Открытие форм через обработчик	135
	Добавление анимации	137
	Создание ссылок на основе примитива	138
	Создание ссылок на основе графического типа	140
	Ap_Pressure	141
	Работа с параметром инициализации	142
	Наследование	144
	Установка дополнительных библиотек и готовых решений в SePlatform.HMI	148
	Добавление графиков в проект SePlatform.HMI	150
	Отображение событий в проекте SePlatform.HMI	151
	Использование безопасности. Установка SePlatformSecurity. Конфигурирование OpenLDAP	154
	Работа с SecurityConfigurator	157





3

Настройка SePlatform.secutiry.agent.XML	159
Добавление в проект компонентов безопасности 10. Резервирование	161 164
Резервирование источников	164
Резервирование серверов	166
EthernetAdapter	166
11. Работа с SePlatform.AccessPoint	169
Резервирование истории	174
Заключение	175





## Введение

Данная рабочая тетрадь создана с целью упрощения изучения компонентов Systeme Platform для инженеров и разработчиков с любым уровнем опыта в АСУ ТП (от начинающего, до продвинутого). Результатом прохождения данного материала является простой проект, который охватывает базовые возможности основных компонентов Systeme Platform.

Содержание тетради полностью дублирует видеокурс обучения на образовательной платформе «Learning 4U» Центра обучения Systeme Electric, а его задания повторяют содержание видеообучения, чтобы учащийся мог выбрать наиболее подходящий для него формат, а также иметь возможность повторить курс обучения в случае отсутствия доступа к видеоматериалам.

В рамках программы обучения вы разработаете и изучите условный демонстрационный проект «Нефтегазовая Труба», включающий в себя разработку резервируемой архитектуры, работу с базами данных, настройку человеко-машинного интерфейса, графиков и аварий, а также настройку прав пользователей и многое другое.

Построение демонстрационного проекта будет происходить от простого к сложному. В процессе разработки проекта необходимо будет периодически переключаться между компонентами SCADA и обращаться к средствам отладки.

Для прохождения курса обучения вам будут предоставлены две виртуальные машины: с ОС Windows и с OC Astra Linux Смоленск. Также всем учащимся будет предоставлена дополнительная виртуальная машина, на которой развернут проект сервера для имитации логики работы контроллера и других устройств полевого уровня управления.

К учебному проекту также приложены дополнительные инструкции, которые можно найти в документе <u>«Краткий конспект слушателя многодневного обучения»</u>







5

#### Целевая аудитория

- SCADA-инженеры и специалисты по развертыванию систем диспетчеризации и управления
- Инженеры АСУ
- Операторы и инженеры обслуживающих компаний
- Системные архитекторы и специалисты по интеграции программного обеспечения
- Коммерсанты и специалисты по продвижению программного обеспечения

#### Требования к слушателям

- Успешное завершение вводного вебинара SEP-01 «Systeme Platform единое ПО для любых задач автоматизации»
- Понимание основных принципов развертывания SCADA систем
- Базовые знания по темам: Парадигмы ООП; Основы Java Script; Основы Linux и Windows; Основы сетей в Windows и в Linux

**Примечание:** Базовые знания по указанным выше темам, а также инструкция по взаимодействию с виртуальными машинами описаны в видео 00-1 – 00-5 в рамках части 0 предоставленной программы обучения на учебной платформе «Learning 4U!»

## Требования к слушателям

#### Подготовка к обучению заранее



#### Парадигмы ООП

Совокупность объектов, иерархия, инкапсуляция, наследование, полиморфизм



#### Основы Java Script

Создание простых переменных, преобразование типов, вызов функций, проверка выполнения условий, работа с числами и строками



#### Основы Linux и Windows

PuTTY, SSH, работа с терминалом, редактирование текстовых файлов, управление службами



#### Основы сетей в Windows и в Linux

Адрес сети, адрес хоста, класс сети, основы DHCP, сетевые папки, изменение IP адреса сетевого адаптера







## Урок 01. Установка и настройка требуемых компонентов на Windows

## Установка компонентов Systeme Platform

Для начала необходимо установить компоненты (выделенные фиолетовым) согласно схеме:



Данная схема будет дополняться компонентами и связями по мере прохождения материала. Процесс установки и настройки компонентов описан в документации и видеокурсе обучения. Компоненты не требуют сложной настройки на начальном этапе. Добавляемые элементы будут отмечены зелёным.







## Требования к рабочему месту

**Примечание**: данные требования необходимы только в случае прохождения обучения на собственной машине, а также для понимания первоначальных действий по настройке машин на объекте.

В рамках прохождения обучения в Systeme Electric вам будут предоставлены заранее преднастроенные виртуальные машины.

Рабочие места должны быть оборудованы:

- Двумя машинами с OC Windows 10 (желательно с актуальным пакетом обновлений);
- Виртуальной машиной с ОС Astra Linux Смоленск;
- Не менее 4 ГБ оперативной памяти;
- Все компьютеры должны быть в одной сети;
- Отключен брандмауэр;

На компьютерах предварительно необходимо установить сторонние компоненты:

- .NET Framework 3.5
- .NET Framework 4.6.1 и выше
- Microsoft Visual C++ 2013 Redistributable (x64)
- Microsoft Visual C++ 2013 Redistributable (x86)
- Microsoft Visual C++ 2015-2019 Redistributable (x64)
- Microsoft Visual C++ 2015-2019 Redistributable (x86)
- OPC .NET API 2.00 Redistributables актуальной версии
- OPC Core components Redistributables (x64) актуальной версии
- OPC Core components

Дополнительно:

- Браузер Google Chrome
- Notepad++
- PuTTY для подключения по протоколу SSH
- WinSCP опционально

Совет по установке сторонних компонентов:

Можно сначала включить .NET Framework 3.5. После этого можно запустить дистрибутив SePlatform.Alarms или SePlatform.Trends. В состав данных дистрибутивов входят основные сторонние компоненты, которые необходимы платформе Systeme Platform. После установки SePlatform.Alarms или SePlatform.Trends можно удалить, сторонние компоненты останутся.

### Настройки системы

Для корректной работы *SePlatform.Historian* необходимо настроить блокировку страниц памяти. Зайти в Панель управления → Просмотр: Мелкие значки → Администрирование → Локальная политика безопасности → Локальные политики → Параметры безопасности → Назначение прав пользователя → Блокировка страниц в памяти → Добавить пользователя или группу → Дополнительно → Поиск → Добавить пользователей Все и АНОНИМНЫЙ ВХОД → Нажать ОК.





8

## Об АСУ ТП

АСУ ТП (Автоматизация систем управления технологическим процессом) — комплекс управляющих компьютерных устройств и их объединений с целью управления разнообразными процессами. Используется преимущественно на промышленных предприятиях (в т.ч. энергетической, нефтегазовой сферы, производства), а также для управления транспортом, инженерными системами и т.п.

## Экосистема Systeme Electric



**Нижний уровень** (полевой уровень, уровень подключаемых устройств) – это уровень того физического оборудования, которое находится прямо на объекте. Этим оборудованием необходимо управлять и мониторить его работу. Это уровень датчиков, измерительных устройств, контролирующих управляемые параметры, а также исполнительных устройств, воздействующих на эти параметры (краны, задвижки, системы вентиляции, насосы, двигатели, и т.п.). На этом уровне осуществляется согласование сигналов датчиков с входами устройства управления, а вырабатываемых команд с исполнительными устройствами.

**Средний уровень** (контроль, сбор и обработка данных, уровень управления оборудованием) — это уровень контроллеров.

ПЛК (Программируемый логический контроллер, PLC) получает информацию с контрольноизмерительного оборудования и датчиков о состоянии технологического процесса и выдает команды управления в соответствии с запрограммированным алгоритмом управления на исполнительные механизмы. По сути, работает с нижним уровнем, обменивается информацией с верхним уровнем.

Верхний уровень (уровень операторских и диспетчерских станций) — это комплекс программноаппаратных средств, выполняющих функции сбора и предварительной обработки данных от датчиков технологического процесса о состоянии оборудования и исполнительных механизмов. Здесь происходит контроль данных на достоверность, обеспечение человеко-машинного интерфейса (HMI). С верхнего уровня оператор получает информации о текущем состоянии технологического процесса в виде мнемосхем, графиков, таблиц, сигнализаций. Это уровень SCADA и MES систем, который работает со средним уровнем либо со сторонними системами верхнего уровня для сбора, передачи, хранения, обработки и анализа данных.





9

## O Systeme Platform

#### Архитектура

Программный комплекс Systeme Platform состоит из компонентов, используемых для разработки, исполнения и сопровождения проектов автоматизации. Функциональность платформы охватывает верхний уровень АСУ.

## Экосистема продуктов Systeme Soft

Комплексный подход к цифровизации - **EcoSysteme** 



Ключевая единица платформы – домен (совокупность вычислительных средств для исполнения проекта автоматизации). В него входят серверные компоненты: *SePlatform.Server* (сервер вводавывода), *SePlatform.Historian* (сервер истории).

К подсистемам инфраструктуры Systeme Platform платформы относится SePlatform.Security (безопасность и разграничение прав пользователей), SePlatform.Diagnostics (комплексные функции диагностики,

мониторинга и аудита) и SePlatform.Licensing (обеспечивает лицензирование продуктов).

*SePlatform.AccessPoint* – единая точка доступа, объединяющая серверные (в т.ч. резервируемые) компоненты и удалённые домены платформы. Служит точкой предоставления данных сторонним системам верхнего уровня, и пользовательской среде Systeme Platform платформы.





Пользовательская среда для работы с визуальной частью проекта, обеспечивающая человекомашинный интерфейс: *SePlatform.HMI* (мнемосхемы), *SePlatform.Alarms* (события и тревоги) и *SePlatform.Trends* (графики).

Среда разработки и управления для создания, тестирования и отладки приложений включает *SePlatform.DevStudio* и *SePlatform.Tools*. Большая часть компонентов Systeme Platform платформы кроссплатформенная.

## 1. O SePlatform.Server

Основной задачей *SePlatform.Server*, как компонента Systeme Platform, является выполнение функций сервера ввода/вывода. *SePlatform.Server* выполняет сбор технологических данных с коммуникационных устройств в ходе мониторинга контролируемых объектов. На основе полученной информации осуществляется контроль над технологическим процессом. Управление может происходить как по команде оператора при передаче собранных данных на верхние уровни АСУ ТП, так и по встроенным алгоритмам. *SePlatform.Server* является шлюзом для работы SCADA-системы с устройствами ввода/вывода

информации. Одновременно сервер может поддерживать соединение с несколькими промышленными контроллерами. Возможна установка нескольких экземпляров *SePlatform.Server* на одном компьютере. *SePlatform.Server* работает в системе в виде службы. Для запуска/остановки сервера достаточно запустить/остановить службу *SePlatform.Server*.

#### Архитектура

Основными составляющими *SePlatform.Server* являются программные модули. Набор модулей подбирается в соответствии с поставленной задачей.







#### Задачи, решаемые сервером:

- Сбор данных с устройств и сторонних серверов:
  - о IEC 104 Master (спецификация ГОСТ Р МЭК 60870-5-104)
  - о IEC 61850 Client (спецификация МЭК 61850)
  - о Modbus TCP Master (спецификация Modbus TCP)
  - о Modbus RTU Master (спецификация Modbus RTU)
  - о OPC DA Client (спецификация OPC DA)
  - **OPC HDA Client** (спецификация OPC HDA)
  - о OPC UA Client (спецификация OPC UA)
  - о Hub Module (TCP, Файловый интерфейс)
  - SQL Connector (SQL)
  - SNMP Manager (SNMP (v1, v2, v3), ICMP)
  - Syslog Server (Syslog)
  - о NetDiag (сети TCP/IP)
- Предоставление данных клиентам:
  - о IEC 104 Slave (спецификация ГОСТ Р МЭК 60870-5-104)
  - о Modbus TCP Slave (спецификация Modbus TCP)
  - о Modbus RTU Slave (спецификация Modbus RTU)
  - о OPC DA Server (спецификация OPC DA)
  - о OPC HDA Server (спецификация OPC HDA)
  - о OPC AE Server (спецификация OPC AE)
  - о OPC UA Server (спецификация OPC UA)
  - о TCP Server Module (TCP, Файловый интерфейс)

Ядро является центральным компонентом сервера для реализации инфраструктуры сервера, интерфейсов работы с модулями, сигналами и их свойствами, остальными подсистемами. Основные функции ядра:

- Управление работой модулей;
- Управление резервированием;
- Запись и чтение данных из БД;
- Отправка и принятие уведомление об изменении сигналов;
- Пересчёт из физических значений в инженерные и обратно (линейная зависимость, и с изломом);
- Выполнение по событиям, таймеру, расписанию;

#### Резервирование:

Для повышения надёжности и минимизации времени отсутствия основных функций системы сбора при невозможности ПОЛНОГО ДУБЛИРОВАНИЯ серверов используется ГОРЯЧЕЕ РЕЗЕРВИРОВАНИЕ. Горячим резервированием управляет Модуль резервирования.

Логическая обработка данных:

Промежуточной обработкой данных занимается Logics Module. Вычисления, проводимые модулем, вынесены на уровень ядра.

#### Возможности модуля логики:

- Пересчёт значений из физических в инженерные и обратно;
- Пересчёт по формуле (логические, арифметические, битовые операции);
- Выполнение алгоритмов по событию, таймеру, расписанию;
- Вызов функций из внешних динамических библиотек (\*.dll);
- Перехват генерируемых событий и тревог;
- Опциональное изменение свойств сигнала VQT (Модуль Write VQT);
- Разбор буфера для выделения кода технологического объекта и кода события



(Модуль Data Buffer).

#### Генерация событий и тревог:

На основе полученных и обработанных данных, сервер может по заранее определённым правилам генерировать и предоставлять пользователям сообщения о событиях и тревогах (Модуль OPC AE Server). Возможности:

- Генерация событий по спецификации ОРС АЕ;
- Предоставление по ОРС DA, UA, AE;
- Сохранение событий в историю;
- Отправка информации о событиях по электронной почте;

Прочие возможности SePlatform.Server:

- Сохранение текущих значений сигналов в файлы срезы XML (Модуль Snapshot) и бинарные файлы (Модуль FS Generator);
- Диагностика сетевых устройств (Модуль NetDiag);
- Предоставление данных для записи в сервер истории (Модуль History Module);

Сервисное обслуживание сервера:

- Редактирование конфигурации с помощью SePlatform.DevStudio;
- Просмотр статистики с помощью приложения Статистика;
- Просмотр журналов работы модуля с помощью приложения Просмотрщик лога кадров;
- Управление работоспособностью сервера и резервной пары с помощью приложения Управляющий;
- Для отладки используется SePlatform.OpcExplorer и SePlatform.EventLogViewer.

#### Работа с SePlatform.Server

Каждый экземпляр SePlatform.Server работает в системе в качестве службы. Сервер, как отдельный компонент, можно конфигурировать с помощью сервисного приложения *Конфигуратор*. Также можно конфигурировать сервер в составе проекта автоматизации с помощью SePlatform.DevStudio. В том числе, с помощью подсистемы SePlatform.Domain можно разворачивать конфигурации на экземпляры SePlatform.Server.

#### Сервисные приложения

В комплект поставки SePlatform.Server входят сервисные приложения для управления, конфигурирования, просмотра статистической информации. Данные приложения предназначены для установки на автоматизированных рабочих местах администраторов. Все сообщения о работе сервера и его компонентов фиксируются в журнал приложений ОС Windows.

#### Конфигуратор

Сервисное приложение Конфигуратор входит в состав клиентской части дистрибутива SePlatform.Server.

Приложение Конфигуратор можно запустить: Пуск → SePlatform → Конфигуратор. В новых проектах, созданных с помощью *SePlatform.DevStudio*, *Конфигуратор* не используется. *Конфигуратор* позволяет решать следующие задачи пользователя:

- Открытие и просмотр адресного пространства сервера;
- Конфигурирование дерева сигналов сервера;
- Формирование перечня свойств сигналов сервера;
- Конфигурирование и настройка модулей сервера;
- Сохранение созданной конфигурации в файл;
- Загрузка готовой конфигурации из файла;
- Отображение статистической информации сервера и журнала конфигурирования;



- Создание резервной копии текущей конфигурации;
- Создание и редактирование пароля для защиты конфигурации от несанкционированного доступа.

Состав окна Конфигуратора:

- 1. Главное меню;
- 2. Панель инструментов;
- 3. Область конфигурирования;
- 4. Область настраиваемых параметров;
- 5. Область логирования: блокировки, история команд и журнал событий.

Файл Сервер Модули Сигналы Свой	тва Блокировки Помощь	
Unnesti Mozyak Utatikotika	Номер Има Значени	о Описание
8		0
Блокировки История комана, Журнал события Время Тип блокировки Пользовал	Dofaarna Vameruma Vaannaa ena Komaarep Voen	
	6	

#### Статистика

Сервисное приложение *Cmamucmuka* предназначено для просмотра статистических данных сервера ввода/вывода *SePlatform.Server*, сервера исторических данных *SePlatform.Historian* и сервера лицензирования *SePlatform.LicServer*. В статистике сервера можно увидеть количество лицензируемых тегов и модулей у готовой конфигурации.

Приложение *Статистика* можно запустить: Пуск → SePlatform → Статистика. Сервисное приложение *Статистика* позволяет:

- Подключаться к вышеперечисленным серверам по интерфейсу TCP/IP;
- Сохранять текущий набор статистических данных в файл;
- Загружать для просмотра файлы статистических данных.





#### Просмотрщик лога кадров

Сервисное приложение Просмотрщик лога кадров в составе дистрибутива SePlatform.Server предназначено для просмотра журналов работы модулей SePlatform.Server.

Приложение Просмотрщик лога кадров можно запустить: Пуск → SePlatform → SePlatform.Server → Просмотрщик лога кадров. Сервисное приложение Просмотрщик лога кадров позволяет:

- Отображать журнал работы модуля как в оперативном режиме, так и сохранённые журналы;
- Фильтровать и искать записи;
- Экспортировать сообщения о работе модуля в табличный файл.

Журнал позволяет отображать:

- Сообщения о работе модуля;
- Технологические данные, с которыми работает модуль.

p	Дата	Время	Описание	<u>^</u>
0	18.04.2017	10:18:54:750	Журнал открыт для записи	
1	18.04.2017	10:18:54:750	> CModule::OnInitialize	
2	18.04.2017	10:18:54:820	< CModule::OnInitialize	
3	18.04.2017	10:18:55:300	> CModule::OnWorkingMode	
4	18.04.2017	10:18:55:300	< CModule::OnWorkingMode	
5	18.04.2017	10:18:55:410	> CModule::OnStart	Журнал открыт для записи
6	18.04.2017	10:18:55:430	< CModule::OnStart	
7	18.04.2017	10:20:12:734	> CModule::OnStop	
8	18.04.2017	10:20:12:744	< CModule::OnStop	
9	18.04.2017	10:20:12:824	> CModule::OnUninitialize	
10	18.04.2017	10:20:12:824	< CModule::OnUninitialize	0 1 2 3 4 5 6 7 8 5
11	18.04.2017	10:20:12:824	Журнал закрыт	
12	18.04.2017	10:20:15:311	Журнал открыт для записи	-





## 2. Работа c SePlatform.DevStudio

SePlatform.DevStudio – это инструмент для построения проектов автоматизации (конфигураций SePlatform.Server). В основе построения проектов используется объектно-ориентированный подход.

SePlatform.DevStudio позволяет работать над несколькими проектами в команде благодаря возможности модульного построения проектов и развёртывать их с помощью подсистемы развёртывания SePlatform.Domain. SePlatform.DevStudio создаёт конфигурации для SePlatform.Server на основании

решения, в котором описывается информационная модель обмена данными между компонентами и логика их работы.





#### Знакомство с компонентами разработки. Развертывание простого проекта

Для начала необходимо научиться настраивать простой проект развёртывания. В примере будем применять простую конфигурацию с одним сигналом на основной экземпляр *SePlatform.Server*.

Создание простой конфигурации:

- **2.1** Запустите *SePlatform.DevStudio* Пуск  $\rightarrow$  SePlatform  $\rightarrow$  SePlatform.DevStudio.
- **2.2** Создайте проект развёртывания. Файл → Создать проект → Проект развертывания



2.3 Задайте проекту имя Demo\_Dev, затем нажмите «готово»

🗄 Созда	ать проект	×
咨	Проект	
Шаблон	Имя проекта	-
проект	Demo_Dev	
	Размещение	
	C:\Users\SM01\Desktop\Demo_Dev	- 📂
	Решение	
	Добавить в решение	-
	Разместить проект и решение в одной и той же папке	
	Назад Далее Готово	Отмена





Перед Вами среда разработки, которая содержит:

- а. Панель управления;
- b. Обозреватель решений (используется для навигации по проекту);
- с. Панель элементов (содержит список элементов, которые можно расположить внутри того или иного места в проекте);
- d. Панель со свойствами;
- е. Панель атрибутов;
- f. Область журнала и поля для ввода формул;
- g. Рабочую область.

Demo_Dev - SePlatform.Development Stud     Dovers Rus Dependent Development	io V		- 0 ×
	윤 🏷 🗇 local 🕞 🗃 📇 🔹 🗸 😜 🔒 🔒		Systên
🕞 Обозреватель решений	A X Domain 4 X	Паг Своиства	<b>→</b> ‡
Поиск	A 🕹 Demo Dev Domain Seplatform Domain 💼 Demo Dev omv	a Domain SePlatform	n.Domain
E Bemo Dev		О Общие	
	~	Annec	local
		Mus	Demain
	Server      DetermetAdapter	¥ 1828	Domain
b			d
	f .		
		@ Атрибуты	<b>~</b> 1
		as Domain SePlatform	n.Domain
		Атрибут	Значение
Панель элементов	▼ ↓ × 100 % × 器田	Комментарий	Домен - это описани
оиск	р Диаграмма Таблица		среды исполнения,
🔛 Контроллеры	^ [жЖурнал → Ф х		для построения
Ссылка на SePlatform.Domain	😒 Ошибок 0 1 Предупреждений: 0 🕕 Сообщений: 0 1 🗅 א א סיר א א א א א א א א א א א א א א א א א א א		конфигураций. Здесе
🚷 Приложение	Текст Элемент Файл Проект		машины, которые бу
Компьютер			представлены в наше
Узел SePlatform.Domain С		1	
Рабочее место	9		
Ceto Ethernet			e
Файловый обмен			
~	Журнал Формулы		

Подсказка: для удобства работы можно опционально произвести несколько настроек интерфейса:

Чтобы упростить работу с формулами нужно перейти Файл → Параметры, выбрать параметр «Формулы» и отметить два чек-бокса: «Перенос строк» и «Подсветка текущей строки»







18

Чтобы сменить язык интерфейса нужно так же в меню Файл → Параметры выбрать «Интерфейс» и сменить язык на Русский, либо Английский (он сменится после перезапуска DevStudio)

🔅 Параметры

🌼 Параметры		$\times$
Интерфейс	Интерфейс	の 🟠
Среда исполнения Безопасность Отладка Сертификаты Аутентификация Формулы Фильтрация Публикация модулей Импорт модулей	Язык интерфейса       Русский         Документация       Покальная (необходимо установить документацию на компьютере)         Па сайте (требуется доступ к интернету)       Путь к документации (покальный файл или адрес сайта)         С:\Program Files (x86)\SePlatform\SePlatform.Documentation\Index.htm	
	Закрыть (	Отменить

Для отображения инструментов выравнивания можно перейти в меню Файл — Панели инструментов и отметить галочкой «Диаграмма»

Файл Правка	Вид	Проект Помощь			
Салл Правла С Обозреватели Поиск № Дето_De		Проект Помоща Обозреватель решений Атрибуты Буферизация Панель элементов События Результаты поиска История Формулы Журнал		Dom WIN T	• ₩ ₩ <sub>+</sub> ! nain SePlatform.Domain
		Свойства Пересчет Дерево сигналов Панели инструментов	•	net/	Стандартная
		Вернуться назад Перейти вперед Подняться на уровень выше Обновить Сброс макета окон	Ctrl+- Ctrl+Shift+- Ctrl++ F5	* *	Мастеры Диаграмма





**2.4** Для упрощения навигации и структуры в проекте можно использовать комментарии. Комментарии используются для удобства разработки проектов. Они отображаются только в проекте DevStudio. Желательно указывать комментарии для всех компонентов, используемых в проекте. Комментарий также можно посмотреть при наведении курсора на элемент, либо во вкладке Атрибуты.

Чтобы добавить комментарий к элементу, щёлкните на него ПКМ → Комментарий. Задайте комментарий элементу *Домен*.

윪 Domain 텍 🔆	🖂 Комментарий 🛛 🗙
Адрес: юса	Домен - это описание среды исполнения,
EDU-WIN	которое используется для построения
🛶 Ethernet 🖉	конфигурации. здесь будут описаны все машины, которые будут представлены в нашем проекте.
	Да Отмена

2.5 Перейдите внутрь элемента Domain. Здесь расположены элементы Cemb Ethernet и Узел SePlatform.Domain, который необходим для описания исполняющих компонентов и серверов ввода-вывода, соответствующих серверному компьютеру. Задайте комментарии для Cemu и Узла SePlatform.Domain.

Укажите в качестве имени *Узла SePlatform.Domain* (в панели свойств) имя своей виртуальной машины (вместо host), на которой Вы сейчас находитесь (имя машины можно посмотреть через Пуск → Параметры → Система → О программе → Имя устройства).



2.6 Перейдите внутрь Узла SePlatform.Domain . Здесь в свойстве «Адрес» у Ethernet Adapter введите IP-адрес машины, на которой Вы разрабатываете проект (IP-адрес машины можно посмотреть через Пуск → Параметры → Сеть и интернет → Свойства, в поиске меню «Пуск» ввести команду СМD, далее в командной строке ввести ipconfig). Задайте комментарий Ethernet Adapter как указано на рисунке ниже.

EDU-WIN* 🛥 🗙		
📳 Demo_Dev.Domain, EDU-WIN Vsen SePlatform.Domain 🖷 🛛 Demo_Dev.omx		
ЕthernetAda на Стмена	<ul> <li>Свойства</li> <li>ЕthernetAdapter Adam</li> <li>Общие</li> <li>Адрес</li> <li>Сеть</li> <li>Имя</li> </ul>	▼ 項 ×       rep Ethernet       10.0.0.51       Ethernet       EthernetAdapter





**2.7** Так как *SePlatformServer* уже установлен (его можно найти в списке служб), можно начинать его описывать.

Processes Performance App	history	Startup Users Details Services		
Name	PID	Description	Status	Group
Schedule	1440	Task Scheduler	Running	netsvcs
SCPolicySvc		Smart Card Removal Policy	Stopped	netsvcs
SDRSVC		Windows Backup	Stopped	SDRSVC
seclogon		Secondary Logon	Stopped	netsvcs
SecurityHealthService	6900	Windows Security Service	Running	
SEMgrSvc		Payments and NFC/SE Manager	Stopped	LocalService
SENS	1720	System Event Notification Service	Running	netsvcs
Sense		Windows Defender Advanced Threat	Stopped	
SensorDataService		Sensor Data Service	Stopped	
SensorService		Sensor Service	Stopped	LocalSystemN
SensrSvc		Sensor Monitoring Service	Stopped	LocalServiceA
seplatform.domain.agent	2840	seplatform.domain.agent	Running	
SePlatform.LicenseServer.A	2908	SePlatform.LicenseServer.Agent	Running	
seplatform.net.agent	2880	seplatform.net.agent	Running	
SePlatform.Server	2888	SePlatform.Server	Running	
🔍 SessionEnv		Remote Desktop Configuration	Stopped	netsvcs
SgrmBroker	5344	System Guard Runtime Monitor Bro	Running	
SharedAccess		Internet Connection Sharing (ICS)	Stopped	netsvcs
SharedRealitySvc		Spatial Data Service	Stopped	LocalService
ShellHWDetection	2092	Shell Hardware Detection	Running	netsvcs
🔍 shpamsvc		Shared PC Account Manager	Stopped	netsvcs
Smphost		Microsoft Storage Spaces SMP	Stopped	smphost
SmsRouter		Microsoft Windows SMS Router Serv	Stonned	LocalServiceN

- 2.8 Внутрь *Узла SePlatform.Domain* learning 00 перетяните из панели элементов *SePlatform.Server*,
- **2.9** Задайте ему комментарий: «В данном компоненте будет описан экземпляр сервера вводавывода, на который в дальнейшем будет залита конфигурация».

Demo_Dev - SePlatform.Development Studio	다 II 🔳 (U) Ctri-Alt-Del 🕮 EN 윩 📋 🖸 🗊 .II	0		- a ×
Файл Правка Вид Проект Помощь				Continue.
		티카루 베 교 태 비 후 .		C Systeme
🖂 Обозреватель решений 🔹 🏚 🗙	EDU-WIN" + X		🗢 🔛 Свойства	• # ×
Поиск Р -	Demo Dev.Domain, EDU-WIN Vser SePlatform.Domain      Demo_Dev.omx	Server -	Server SePlatform.Se	rver *
Demo_Dev			<ul> <li>Общие</li> </ul>	
			Пароль	
			Отладочный вывод	Нет
	EtherretAda #0		Имя	Server
	G Sener ♥		@ Атрибуты Server SePlatform Se	v a x
			Атрибут	Значение
Г <sup>О</sup> Панель элементов 🔹 🛚 🗶	100% * 撥任 <	3	Комментарий	Экземпляр сервера
Поиск Р -	Диаграмма Таблица			ввода-вывода на вин
Папка обмена	Сь Журнал	- 0	I X	
Последовательный пор	O Ошибок 0 1 Предупреждений: 0 0 Сообщений: 0 4 Ст 50	Поиск по журналу	D -	
SePlatform.Server	Текст	Элемент Файл Проект		
Резервный SePlatform.Server				
Дублирующий SePlatform.Server				
SePlatform.Imitator				
SePlatform.Historian			Activate Windo	
Дублирующий SePlatform.Historian	Хирнаа Формузы		Go to Settings to ac	tivate Windows.
Загоузка решения завершена	Всего: 2, из них выделено:	1		





По умолчанию при добавлении Сервера ввода-вывода в конфигурации уже присутствуют модули OPC DA, AE, HAD, UA, TCP Server и модуль истории. Для начала будем использовать модуль OPC DA Server для просмотра оперативных значений сигнала.

**2.10** Перейдите внутрь *SePlatform.Server.* Чтобы описать, что будет происходить в этом экземпляре сервера, добавьте элемент *Приложение* (из панели элементов). Здесь описывается дерево объектов и параметров, которое будет обслуживаться данным сервером. Задайте ему комментарий как показано на рисунке ниже.



**2.11** Перейдите в элемент *Application* и перетяните сюда из панели элементов *Логический объект* и так же задайте ему коментарий.

Demo_Dev - SePlatform.Development Studio			- ⊡ X
			• The form
Состреватель решении • 4 х	Application "9 A	CEOUCTEA	• # X
I DALK	AC Demo Dev.Domain.EDU-WIN.Server, Application Приложение III Demo_Dev.omx • Object •	Објест Логический об	BEKT
C Demo_Dev	8	Tur	Ê
	Colect #2 #7	Уровень доступа	публичный
		Область доступа	глобальная
		Аспект	
		Представляемый объект	
	Комментарий Х	Има	Object
	До Отмена	<ul> <li>Фрибуты</li> <li>Оbject Логический об Атрибут</li> </ul>	тах ъект т Значение
🖓 Панель элементов 🔹 Ф 🗙	80 % Y RE	Комментарий	В данном объекте будут
Поиск Р +	Диаграмма Таблица		описаны параметры (Сигналы)
🗄 🛄 Карты адресов	🕞 Журнал 🗧 🖣 🗙		1.
Р 🔲 Ссылки	😢 Ошибок 0 🕴 Предупреждений: 0 🚯 Сообщений: 0 🔔 📭 🧤 🖓		
Физический объег	Текст Элемент Файл Проект		
Логический объект			
Сокет			
Исполняемый модуль приложения	I and a second sec		
Значения атрибутов	Журнал Формулы		
Sarroute a neurebulg takenuleba	Rearry 1 us you subseteeo 1		





#### 2.12 Внутрь этого логического объекта добавьте параметр типа Uint4 из панели элементов.

E Demo_Dev - SePlatform.Development Stu	udio	× ·		– ø ×
Файл Правка Вид Проект Помощь				
	80	○ loal ·西田。 《 》 《 ▲ ▲ 》 笔 名 电 白   二 キ 의 可 + 브   キ キ 即 同 田 트	+ [	Système     plattore
Обозреватель решений	• # ×	Object* 49 X	🗢 😭 Свойства	• # X
Поиск	ρ-		🔆 🧳 Object Логический о	бъект •
Demo_Dev			Общие	
		🔾 🗊 . Parameter	Тип	
			Уровень доступа	публичный
			Область доступа	глобальная
			Аспект	
			Представляемый объект	τ
			Имя	Object
	/		<ul> <li>Датрибуты</li> <li>Object Логический о</li> </ul>	• # ×
			Атрибут	Значение
ПО Панель заементов	• a ×	30 % × 38E	Комментарий	В данном объекте будут
Поиск	p -	Диаграмма Таблица		описаны параметры
Событие	^	🕞 Хурнал 👻 9	×	(сигналы)
4 Параметр		🕲 Ошибок 0 I Предупреждений: 0 🕕 Сообщений: 0 🔟 🗅 🏠 Поиск по журналу 🔎		
0 bool	10	Текст Элемент Файл Проект		
0 intl				
() int2				
1 uint2				
🖸 int4			Activate Window	
U uint4		Журнал Формулы	Go to Settings to acti	
Загрузка решения завершена		Bcero 1		

Параметры – сигналы, которые будут в конфигурации *SePlatform.Server*. По отношению к другим компонентам (контроллерам, другим серверам), параметры характеризуются свойством <u>Направление</u>. *Входные параметры* влияют на состояние объекта, отправляют управляющие воздействия (команды, отправляемые на контроллер). *Выходные параметры* характеризуют состояние объекта (состояние, текущие значения, полученные от контроллера).

Простой проект готов, осталось его построить и развернуть (применить конфигурацию к определённому экземпляру сервера).

#### Настройка развёртывания проекта

Перед тем, как применить конфигурацию на определённый экземпляр *SePlatform.Server*, необходимо настроить *SePlatform.Domain*. Подсистема развёртывания состоит из двух служб: **SePlatform.Net.Agent** и **SePlatform.Domain.Agent**. В ходе дальнейшего усложнения проекта в данной методичке будет подробное изложение настройки определённых XML-файлов, относящихся к службам **SePlatform.Net.Agent** и **SePlatform.Domain.Agent**. На данном этапе разработки учебного проекта для развёртывания достаточно настроек по умолчанию.

#### Построение и развёртывание проекта

При построении решения *SePlatform.DevStudio* строит связи согласно информационной модели обмена данными между элементами в проекте. По описанным связям создаётся конфигурация, которая с помощью подсистемы развёртывания будет применена к нужному экземпляру сервера. Для развёртывания конфигурации необходимо сначала **Построить решение**, затем **Перейти к развёртыванию** для применения конфигурации к нужному экземпляру.



**2.13** Нажмите на кнопку Построить решение. На данном этапе SePlatform.DevStudio компилирует решение и строит связи между компонентами для построения конечных конфигураций SePlatform.Server.

2.14 Нажмите на кнопку Перейти к развёртыванию. Откроется Мастер развёртывания, в





## котором отображаются сконфигурированные экземпляры *SePlatform.Server* в проекте **2.15** Нажмите на кнопку **Применить конфигурацию**.



**2.16** Во время применения конфигурации подсистема развёртывания останавливает службу нужного экземпляра SePlatform.Server, подкладывает новую готовую конфигурацию для данного экземпляра в папку проекта.

File Home Share View	v								
← → × ↑ 🔤 > Demo_Dev	> Demo_Dev	/ > bin > loc	al → Doma	in > EDU-WIN	> Server	>			
Name	1	^		Date modified		Туре		Size	
🖈 Quick access	platform.serve	er.cfa		16.11.2022 10:	35	CFG File		32	КВ
Desktop 🖈	platform.serve	er.cfa.tmp		16.11.2022 10:	36	Compresse	d (zipp	4	KB
🕹 Downloads 🛛 🖈									
🖹 Documents 🛛 🖈									
и запускает службу Task Manager	ý						- C	ı ×	
и запускает службу Task Manager File Options View Processes Performance App I	Y nistory Star	rtup Users	Details 5	Services				ı ×	
и запускает службу Task Manager File Options View Processes Performance App I	y history Star PID C	rtup Users Description	Details S	Services	Status	-	- C	ı ×	^
и запускает службу Task Manager File Options View Processes Performance App I Name SePlatform.LicenseServer.A	y history Star PID C 2908 S	rtup Users Description SePlatform.Lic	Details ScenseServe	Services er.Agent	Status Runnin	9	- C	1 ×	^
и запускает службу Task Manager Task Manager Orocesses Performance App I Name SePlatform.LicenseServer.A seplatform.net.agent	history Star PID C 2908 S 2880 s	rtup Users Description SePlatform.Lic seplatform.ne	Details Serve	Services er.Agent	Status Runnin Runnin	- g g	- C	1 ×	^
и запускает службу Task Manager Tile Options View Processes Performance App I Name SePlatform.LicenseServer.A seplatform.net.agent SePlatform.Server	y history Star PID C 2908 S 2880 s 2888 S	rtup Users Description SePlatform.Lic seplatform.ne SePlatform.Se	Details ScenseServe t.agent rver	Services er.Agent	Status Runnin Runnin Runnin	- 9 9	- C	ı ×	^
и запускает службу Task Manager File Options View Processes Performance App I Name SePlatform.LicenseServer.A seplatform.net.agent SePlatform.Server SessionEnv	y history Star PID C 2908 S 2880 s 2888 S R	rtup Users Description GePlatform.Lic GePlatform.Se Remote Deskt	Details ScenseServe t.agent rver op Config	Services er.Agent uration	Status Runnin Runnin Stoppe	9 9 9	- C Group netsvcs		^
и запускает службу Task Manager File Options View Processes Performance App I Name SePlatform.LicenseServer.A Seplatform.net.agent SePlatform.Server SessionEnv SegmBroker	y history Star PID D 2908 S 2880 s 2888 S 8 5344 S	rtup Users Description SePlatform.Lic SePlatform.Se Remote Deskt System Guard	Details Serve t.agent rver op Config Runtime I	Services er.Agent uration Monitor Bro	Status Runnin Runnin Runnin Stoppe Runnin	9 9 9 d 9	- C Group netsvcs		^

Fewer details | Q Open Services

**В результате** получается запущенный экземпляр SePlatform.Server на локальной машине с новой готовой конфигурацией, которая описана в проекте. Время применения конфигурации и её версию можно увидеть в Мастере развёртывания в столбце Активная версия.





## 3. Знакомство и работа с отладочными инструментами SePlatform. Tools

Для отладки и диагностики компонентов *SePlatform Platform* используются набор инструментов *SePlatform.Tools*, в состав которого входят приложения:

- SePlatform.OpcExplorer инструмент для работы с OPC серверами по протоколам DA, AE, HDA, UA, который позволяет:
  - Просматривать адресное пространство источников данных;
  - о Просматривать, изменять, генерировать значения сигналов;
  - о Просматривать и квитировать события и тревоги;
  - о Просматривать изменения сигналов на графике;
  - Запрашивать исторические данные;
- SePlatform.EventLogViewer инструмент для удобного просмотра системных журналов для диагностики работоспособности системы, который позволяет:
  - Просматривать события системных журналов на локальном и удалённом компьютерах;
  - о Импортировать и экспортировать системные журналы в файл;
  - о Фильтровать и искать события.

Работа с SePlatform.OpcExplorer

Для просмотра значений сигналов можно использовать SePlatform.OpcExplorer.

Работа с сигналами. Подключение к OPC DA Server

*SePlatform.OpcExplorer* позволяет подключаться к различным OPC (DA, AE, UA, HDA) источникам, просматривать, изменять значения.

**3.1** Откройте *SePlatform.OpcExplorer* (Пуск  $\rightarrow$  SePlatform  $\rightarrow$  SePlatform.OpcExplorer).

Подсказка, в этом инструменте также можно сменить язык интерфейса на Русский нужно выбрать Файл → Параметры после чего выбрать параметр интерфейс и в выпадающем списке «Сменить язык» выбрать Русский язык







**3.2** Выберите в поле «Компьютер» имя своей машины, нажмите клавишу Enter. Подсказка – имя BM можно посмотреть в свойствах ПК щелкнув ПКМ по иконке «Этот Компьютер»

Y 🛄 Thi	Collapse	û Home	About			
> 31 -	Manage	Find a setting	Your PC is monitored and protected.			
	Pin to Start	System	See details in Windows Security			
	Map network drive					
> 🕂 D(	Open in new window	🖵 Display	Device specifications			
> 🎝 M	Pin to Quick access	(如) Sound	Device name DESKTOP-IHJVGQ8			
> 📰 Pi	Disconnect network drive	14 55515	Processor Intel(R) Xeon(R) CPU X5650 @ 2.67GHz 2.50 GHz (2 processors)			
> 📑 Vi 👘	Add a network location	Notifications & actions	Installed RAM 12.0 GB			
> 느 Le —		) Focus assist	Device ID A0511214-3D69-42F3-84D7-D2E114BF3629			
_	Delete	0	Product ID 00330-80000-00000-AA815			
> 💣 Net	Rename	O Power & sleep	System type b4-bit operating system, xb4-based processor			
2 items	Properties	-	ren and touch in the perior touch input is available for this display			

#### **3.3** В поле «Источник данных» выберите Server.



#### **3.4** Нажмите на кнопку Подключиться к источнику.

SePlatform.Service - OPC	Explorer				
Файл Вид Проект Пом	ощь				
00 🔨 -   💐	desktop-ihjvgq8	٠	SePlatform.Server	•	<b>€,</b> <del>%, -</del>
🕂 Обозреватель	-	ņх			Подключиться к источнику
Поиск		<u>ب</u> م			

3.5 Раскройте дерево ОРС источника, к которому произведено подключение, и выберете сигнал, имеющийся в конфигурации. Добавьте сигнал в инспектор сигналов (в дереве ПКМ по сигналу → Добавить в инспектор сигналов).





**Подсказка:** чтобы добавить поля просмотра нужно кликнуть ПКМ по типу поля и в появившемся окне выбрать необходимый параметр



Так как данный сигнал не получает данные с коммуникационного модуля, по умолчанию у него плохое качество (код 28) и пустое значение.





С помощью инспектора сигналов можно просматривать и изменять текущие значения сигналов.

**3.6** Для изменения значения сигнала дважды щёлкните мышкой в графе «Значение» данного сигнала в инспекторе. В появившемся окне установите необходимое значение.







3.7 Для того, чтобы не вводить значения вручную, можно добавить генератор значений. Для его работы необходимо включить его в настройках (Файл → Параметры → Инспектор → Возможность использования генераторов значений → Сохранить → Применить). После этого снова откройте редактор значения сигнала, установите флаг Генерировать значения, нажмите кнопку Добавить, выберите Случайный генератор и нажмите ОК. Теперь у сигнала будут генерироваться случайные значения.

Файл	Вид Проект Помощь											
	Новый проект	Ctrl+N	•	🔗 Se	Platform.Se	rver		• .	🔩 -	4		
2	Загрузить из файла	Ctrl+O	x	Obie	ct.Paramete	rμ	Инспек	тор1 ≁¤	×			
	Сохранить в файл	Ctrl+S	5	Тип	Сигна	ал	Значение		Качес	TEO		
	Сохранить в другой файл			14	Object.Par	ameter	123	хороц	Jee: 216 -	Loca		
ġ.	Параметры											
	Выход Service											
Ø	Параметры										×	<
	Интерфейс Сертификаты Аутентификация Тревоги и события	Инспе	ктор тображ озможн	ать мил	лисекунды	генерат	ODOB SHAVE	ний			26	8
	Инспектор У Ччиљтрация Источники данных Графики Реперы Обозреватель Журнал	B	нимани	е! Гене отпр	рация значен авке недопус	ний мож	ет привести команд	1K				
								3a	крыть	Отменить	Применить	]

🔍 Инспектор 1 - SePlatform.Service - OPCExplorer





#### Подключение к OPC UA Server

В *DevStudio* внутри сервера есть модуль OPC UA Server. Этот модуль является кроссплатформенным: с его помощью можно подключиться не только к серверу, установленному на OC Windows, но и к серверу на OC Linux.

😑 Demo_Dev - SePlatform.Developmen	t Studio	V			– Ø ×
Файл Правка Вид Проект Помо	ощь				
· · · · · · ·	3 # 3	C local · 프 프 . · · · · · · · · · · · · · · · ·	표 # # # # 🗇 🗄 🗷 #	• [	Système platform
🖸 Обозреватель решений	<b>→</b> ↓ ×	Server 40 X	=	😭 Свойства	- ų ×
Поиск	ρ-	Qa Demo_Dev.Domain.EDU-WIN. Server SePlatform.Server ➡ Demo_Dev.omx - UaSe	rver 🗸 🧶	UaServer OPC UA Ce	рвер -
Demo_Dev		[ ເຊິຍ ປະສາດ ເຊິຍ ການ	[10]	Общие Имя точки поакоючения	SePlatformOpcUaServer
		Asserver	-	Запретить клиентам изменение сигналов	Нет
			peeg-1	Переводить в состояние SUSPENDED в резерве	Нет
		HdaServe	apagp	Имя	UaServer
				• Сетевые параметри	al de la constante de la const
			r m l	Порт ОРС ТСР протокола	62544
				Порт НТТР протокола	8080 🗸
		TcpSerr	er 110	@ Атрибуты	• # ×
		1 <u></u>	<u> </u>	UaServer OPC UA Ce	рвер -
				Атрибут	Значение
С Панель элементов	• 4 ×	80% * 1013			
Поиск	p -	Диаграмма Таблица			
🕴 🛅 BACnet	^	🖓 Журнал	<b>-</b> ‡ ×		
EtherNet/IP		😢 Ошибок: 0 1 Предупреждений: 0 🕕 Сообщений: 0 上 🗅 🖓	Поиск по журналу 🔎 👻		
Modbus		Текст Элеме	нт Файл Проект		
<ul> <li>Болого С</li> <li>Коммуникационные модули</li> <li>Эм ЭК 60870-5</li> <li>Эм ЭК 61850</li> </ul>					
🕴 📴 Служебные модули				Activate Window	
🚷 Приложение	~			Go to Settings to act	
<	>	Журнал Формулы			
Saroutica paulaulus tassacillaus		Rearce 7 us your suspension 1			

**3.8** Для подключения к OPC UA Server перейдите в *SePlatform.OpcExplorer*, введите IPадрес своей машины, нажмите клавишу Enter, нажмите на кнопку с монитором (Запрос проверки связи ICMP), расположенную слева.

🔍 10.0.0.51 - SePlatform.Service - OPCExp	lorer	V	_	×
Файл Вид Проект Помощь				
🔾 💭 🔨 🚽 🛝 10.0.0.51	•	🕸 SePlatform.Server 🔹 🔹 🎭 🔹 🥩 🍃 🕂 🚅		
🔧 Обозреватель	<b>-</b> ↓ ×	Оbject.Parameter 1 Инспектор 1 10.0.0.51 1 Х SePlatform.Server 1		Ŧ
Поиск	~ ۵	Image: 10.0.0.51         10.0.0.51         10.0.0.51         ▷ <sup>™</sup> SePlatform.Server		

3.9 Нажмите на кнопку с изображением знака

÷

(Добавить источник).

**3.10** В строке «Спецификация» выберите UA (tcp), в графу «Сервер» впишите Имя точки подключения, указанную в *DevStudio (см. рисунок выше),* сравните порт из *OpcExplorer* с портом OPC TCP протокола из *DevStudio.* Нажмите OK.

Спецификация	🚱 UA (tcp)	•
Сервер	SePlatformOpcUaServer	
Порт		62544





29

- **3.11** В *OpcExplorer* в поле Источник данных выберите добавленный сервер и подключитесь к нему.
- Онспектор 1 SePlatform.Service OPCExplorer × Файл Вид Проект Помощь 🔾 💭 🔨 🖌 😣 10.0.0.51 GePlatformOpcUaServer • 🎭 🗣 🚽 🖛 🔻 💽 🕕 🖲 🥔 🌆 🚦 Обозреватель ▼ # × Object.Parameter 12 Инспектор 1 + × 10.0.0.51 12 SePlatformOpcUaServer 12 Ŧ 2 Тип Сигнал Значение Качество Время Источник 
   ObjectParameter
   123
   xopowee: 216 - Local Override
   22.11.2022
   13:45:36
   SePlatform.Server

   ObjectParameter
   123
   xopowee: 216 - Local Override
   22.11.2022
   13:45:36
   SePlatform.OpcUs
   🔺 🐓 SePlatform.Server на desktop-ihjvgq8 NativeTypes A Diject 1 Paran E Service GePlatformOpcUaServer на 10.0.0.51 D 🚞 Server NativeTypes
   🔺 🛞 Object Parameter 5 777001 Service 777001
- **3.12** Добавьте Параметр из логического типа Object в Инспектор сигналов.

Теперь Вы подключены к серверу Windows по OPC UA. Можно как считывать значения, так и записывать. При этом значение переменной уже было задано по другому протоколу и можно увидеть, что мы получаем идентичную информацию по всем протоколам.





# 4. Установка и настройка серверной части на Linux. Создание узла для экземпляра SePlatform.Server на Linux в SePlatform.DevStudio.

Для выполнения работ вам потребуется подключиться ко второй виртуальной машине с уже предустановленной системой Linux.



#### **Имя пользователя** – SE **Пароль** – 12345678

Для начальной работы нужно установить *SePlatform.Server* и *SePlatform.Domain*. Дистрибутивы находятся в сетевой папке машины Linux.

4.1 2333333333Откройте менеджер файлов.







**4.2** В открывшемся окне выберите Сеть  $\rightarrow$  Distr  $\rightarrow$  Linux . В данной папке предварительно собраны все дистрибутивы

с: Ceть > distr > Linux		
У Избранное	access.txt	
Компьютер	ARM_AccessPoint.txt	
Сеть	ARM_WorkStation.txt	
- 🖉 Samba	Idap pass.txt	
- 📴 distr	☑ Idap.txt	
— 🦲 Документация техническая	seplatform.accesspoint-1.0.x+fix29135.b14.r87226.deb	
>- 📔 Сторонние компоненты	SePlatform.Domain-1.0.6+b4.r86393-x64.deb	
- 🚞 Linux	seplatform.historian-1.0.11+b2.r87074.deb	
>- 📒 Students	seplatform.hmi.alarms-ru_RU-1.1.0+b1.r5878.x86_64.deb	
>- 🚞 sw.addition	SePlatform.HMI.charts-1.0.0+b6.r85537-x64.deb	
- 📔 Windows	seplatform.hmi.commonlib-ru_RU-1.0.3+b1.r5865.x86_64.deb	
	seplatform.hmi.desktop-ru_RU-1.0.1+b71.r87239.x86_64.deb	
	SePlatform.hmi.security-1.0.0+b13.r86667-x64.deb	
	seplatform.hmi.securityconfigurator-ru_RU-1.0.1+b3.r5773.x86_64.deb	
	seplatform.hmi.setpoints-ru_RU-1.0.1+b3.r5772.x86_64.deb	
	seplatform.hmi.tables-ru_RU-1.2.0+b6.r85536.x86_64.deb	
	seplatform.hmi.trends-ru_RU-1.0.x+b78.r5463.x86_64.deb	
	SePlatform.Security-x64-1.0.0+b11.r86668.deb	
	seplatform.server-1.0.x+fix29135.b14.r87226.deb	
		~

**4.3** Чтобы работать с дистрибутивами их необходимо скопировать на вашу BM в любой каталог, в который позволяет система. Удобно работать через папку /home/se/distr







Документацию по работе Systeme Platform можно посмотреть в разделе «Работа с OC Linux»

**4.4** Чтобы начать работу с терминалом Linux нужно запустить терминал Fly (Пуск  $\rightarrow$  Системные  $\rightarrow$  терминал Fly)



**4.5** Далее необходимо открыть папку с дистрибутивами через терминал. Для этого нужно ввести cd /home/se/distr





**4.6** Сначала необходимо установить *SePlatform.Server* на OC Linux описаны в документации: SePlatform.Server → SePlatform.Server → Подготовка к работе → Стандартная установка OC Linux.

Находясь в папке с установочным пакетом, введите команду <mark>sudo dpkg -i SePlatform.server-</mark> \*\*\*\*\*\*\*.deb

**Подсказка:** чтобы избежать ошибок при вводе команд Linux можно использовать функцию «Автодополнение» для этого достаточно ввести часть команды, а затем нажать клавишу «tab» на клавиатуре

👿 distr : bash — Терминал Fly	_ 🗆 ×
Файл Правка Настройка Справка	
🔐 🎴 📋 📋 ks 🗸 🤡	
Выбор ранее не Выбранного пакета seplatform.server. (Чтение Базы данных … на данный момент установлено 123527 файлов и каталогов.) Подготовка к распаковке seplatform.server-1.0.x+fix29135.b14.r87226.deb … Распаковывается seplatform.server (1.0.x+fix29135.b14.r87226) … Настраивается пакет seplatform.server (1.0.x+fix29135.b14.r87226) … Searching for existing OPC UR Server certificates at /opt/SePlatform/SePlatform.Server/Ce s/private. No suitable OPC UR Server certificate were found. Creating OPC UR Server certificate. Senerating a 4096 bit RSA private key 	rtificate
Searching for existing OPC UA Client certificates at /opt/SePlatform/SePlatform.Server/Ce s/private. No suitable OPC UA Client certificate were found. Creating OPC UA Client certificate. Generating a 4896 bit RSA private key	rtificate
++ writing new private key to '/tmp/tmp.rxp5D9of5M/temp.key' 	
Updating OpcUa Server PAM configuration file. Created symlink /etc/systemd/system/multi-user.target.wants/seplatform.server.service → / md/system/seplatform.server.service. Created symlink /etc/systemd/system/graphical.target.wants/seplatform.server.service → /l d/system/seplatform.server.service. se@astrapc:~/distr\$ ■	lib∕syste ib∕system
m Ш П В 2 distr - Менед ► distr : bash 😔 🕁 📢 Е	N 0 11:09

нажмите клавишу Enter.

Для установки SePlatform.Domain, находясь в папке с установочным пакетом, введите команду sudo dpkg -i SePlatform.domain-\*\*\*\*\*\*\*.deb после установки SePlatform.Server и SePlatform.Domain, вернитесь в DevStudio для описания Linux машины в среде исполнения.





#### 4.7 Описание конфигурации *SePlatform.Server* на Linux машине в DevStudio

Для описания ещё одной машины в проекте необходимо перейти в элемент **Domain** (здесь описываются все машины, участвующие в проекте).

Перейдите в элемент *Domain* через Обозреватель решений.



**4.8** Перетяните на рабочую область **Узел SePlatform.Domain** из панели элементов, задайте ему имя, совпадающее с именем машины с OC Linux, адрес (любой), добавьте комментарий как на рисунке ниже (ПКМ по *Узел SePlatform.Domain* → *Комментарий*).







**4.9** Перейдите внутрь созданного *Узла SePlatform.Domain* для Linux машины (в дальнейшем astra), выделите *EthernetAdapter*, в свойстве Адрес укажите IP-адрес Linux машины

**Подсказка:** чтобы узнать IP адрес BM Linux можно переключившись на нее в панели управления кликнуть ПКМ по иконке «сетевое соединение ethernet» и выбрать «сетевое соединение»

🗾 distr : bas	— Терминал	ı Fly		$\sim$		- 0	×
Файл Прав	а Настройн	а Справка					
🔐 🖴	l 📋 Is	~ <	)				
se@astrap dpkg: owu : Her Tak se@astrap Bwbop ран (Чтение б Nogroro8k Pacnako8ы Hactpau8a Created s d/system/ Created s form.net. Created s system/se Created s ystem/se Se@astrap	::«\$ sudo ixa: cann iro файла ::«\$ cd / ::«\$ cd / ::«\$ cd / isa gaнны а к распа а к распа a к распа a к распа main.ser ymlink /e service. ymlink /e service. ymlink /e service. ymlink /e service. ymlink /e service. ymlink /e service. ymlink /e service. ymlink /e service. ymlink /e service. ymlink /e service.	dpkg -i ot acces: unu кат nome/se/i sudo dj pанного и к на g coßke Sel platform т seplat tc/systeu m.domain tc/systeu tc/systeu tc/systeu tc/systeu tc/systeu tc/systeu st.servi tc/systeu st.servi tc/systeu st.servi tc/systeu	SePlatform.Doma s archive 'SePla anora distr okg -i SePlatfor nakera seplatfor aHHBŬ MOMEHT yCT Platform.Domain- domain (1.0.6+b form.domain (1.0 nd/system/dbus-s nd/system/dbus-s nd/system/graphi service. nd/system/dbus-s nd/system/graphi ce. a dbus (1.10.26-	in-1SePlatform.[ tform.Domain-1.0.6+t m.domain. anoBneno 123570 1.0.6+t4.r86393) .6+t4.r86393) eplatform.domain user.target.want cal.target.want cal.target.want cal.target.want seplatform.net.ag user.target.want	Domain-1.0.6+b4.r86393-x64.de Platform.Domain-1.0.6+b4.r8 файлов и каталогов.) -x64.deb n.agent.service → /lib/syste ts/seplatform.domain.service gent.service → /lib/systemd. ts/seplatform.net.service → s/seplatform.net.service → s/seplatform.net.service → s/seplatform.net.service → s/seplatform.net.service → s/seplatform.net.service → s/seplatform.net.service →	deb 36393-x64.deb' emd/system/sep e → /lib/syste /system/seplat /lib/systemd/ /lib/systemd/s	
					изменить соединения		0
1					О программе		-
1	78	2 3 4	distr - Менед	🚬 distr : bash		\$II EN <b>0</b> 11:	43

После чего в отобразившемся диалоговом окне можно будет посмотреть IP адрес LinuxBM

Проводное соединение 2 (по	умолчанию)		
Основное			
Интерфейс:	Ethernet (eth0)		
Аппаратный адрес:	00:50:56:11:7E:12 vmxnet3 10000 Мбит/с		
Драйвер:			
Скорость:			
Защита:	Нет		
ІР-адрес:	10.0.0.50		
Широковешательный адрес.	10.0.0.255		
Маска подсети:	255.255.255.0		
Шлюз по умолчанию:	10.0.254		
Primary DNS:	10.0.0.254		
IPv6			

в свойстве Сеть при помощи двойного нажатия кнопки мыши выберите Ethernet,




**Подсказка:** в качестве альтернативы вместо навигации по меню для того, чтобы узнать IP-адрес можно использовать команду sudo ifconfig в терминале Fly.

<ul> <li>Свойства</li> <li>ЕthernetAdapter Адапт</li> <li>Общие</li> <li>Адрес</li> <li>Сеть</li> <li>Имя</li> </ul>	ep Ethernet 10.0.0.50 EthernetAda	• ‡ ×
Укажите путь к элем Rouce Ethernet	ленту	Х

добавьте комментарий как на рисунке ниже.



Поднявшись на уровень выше к элементу **Domain**, можно заметить, что два узла подключены к одной сети, т.е. имеют адаптеры, которые работают через одну общую сеть и связаны между собой. В



качестве альтернативы можно, зажав ЛКМ провести связь между узлом EDU-Astra и сетью Ethernet, это также добавит связь.





**4.10** Перейдите внутрь узла *astra* и перетяните на рабочую область *SePlatform.Server* из панели элементов, дайте ему имя, отличное от имени сервера внутри виндового узла. Задайте ему комментарий как на рисунке ниже.



**4.11** Перейдите внутрь *AstraServer*. Перетяните из панели элементов *Приложение*, дайте ему название и комментарий как на рисунке ниже.







На машинах с OC Linux не доступны OPC DA и HDA Server, поэтому их можно удалить.

erver*	AstraServer * + ×			
Demo De	v.Domain.EDU-ASTRA. AstraServer SePlatform.Server 📾	Demo_Dev.omx	<ul> <li>DaServer</li> </ul>	-
	HistoryModule	Удаление	sanımı DaServer?	





**4.12** Перейдите внутрь *Astra\_Application* и перетяните из Панели элементов *Логический объект*, назовите его SandBox.

	2.C. [scal] · 비 프 : [석 : [북 씨 석 : [북 씨 번 만] 프 주 의 비 부 프   박 주 환 [고 테 프 후 .	÷	O Systeme
🖾 Обозреватель решений 🔹 🎗	K Server * Astra_Application * + X	🕼 Свойства	- ū ×
Поиск	🦷 🚜 <u>Demo Dev.Domain.EDU-ASTRA AstraServer, Astra Application</u> Приложение 📫 Demo,Dev.omx - SandBox - 😤	🧳 Object Логический об	ekt •
4 👶 Demo_Dev		🔿 Общие	
4 🔄 Demo_Dev	Car Canadan and	Тип	
A Dalita	- Satisfield	Уровень доступа	публичный
Demo_Dev.omx     Oppenentus	1	Область доступа	глобальная
4 🔓 Domain		Аспект	
EDU-ASTRA		Представляемый объект	
<ul> <li>III EDU-WIN</li> </ul>		Имя	SandBox
A Server			
A Gerver			
▲ Server ▲ AcServer ▲ AcServer ▲ @ Application ▲ Ø Object ○ Parameter ♥ Bausen sanarentne ♥ ■		@ Атрибуты	• # X
AcServer AcServ		<ul> <li>Фтрибуты</li> <li>Object Логический об</li> </ul>	• 0 X
Accover     A		<ul> <li>Фтрибуты</li> <li>Object Логический об Атрибут</li> </ul>	тах ьект т
		<ul> <li>Фрибуты</li> <li>Оbject Логический об Атрибут</li> </ul>	• Q X вект • Значение
A Adever A Adever A Adever A Adplication A Object O Parameter Paramete		<ul> <li>Фрибуты</li> <li>Оbject Логический об Атрибут</li> </ul>	← а × ъект – Значение
A derver		<ul> <li>Атрибуты</li> <li>Object Логический об Атрибут</li> </ul>	¥₿Х вест ÷ Энзмение
A Server     A dever     A devever     A devever     A dever     A dever     A dever     A devev		<ul> <li>Орус Атрибуты</li> <li>Object Логический об Атрибут</li> </ul>	<ul> <li>↓ X</li> <li>best</li> <li>3navenne</li> </ul>

**4.13** Перейдите внутрь *SandBox* и добавите в этот логический объект *параметр* типа uint4.

Demo_Dev - SePlatform.Development Studi				- u x
Файл Правка Вид Проект Помощь				
	이 C local · 田田, 야, 오 과 와, 池 원 한 드	· # - 최 표 # # # # # # # # # # # #		Système platform
🖸 Обозреватель решений 🗸 👻	X Server* SandBox* + X		🗢 😭 Свойства	- ü ×
Поиск	Demo Dev.Domain.EDU-ASTRA.AstraServer.Astra.Application.SandBox Логический	объект 🖽 Demo_Dev.omx - Parameter	• O Parameter Параметр	
4 💰 Demo_Dev			Общие	
4 🛃 Demo_Dev	QQ		Уровень доступа	публичный
🔺 🦾 Файлы	O 🚳 Parameter		Область лостипа	rachathwas
📑 Demo_Dev.omx			Userse goorgina	
Определения     Определения			направление	отсутствует
4 💑 Domain			Тип	uint4
EDU-ASTRA			Имя	Parameter
EDU-WIN				
EthernetAdapter				
4 G Server				
AeServer				
A Application				
a 🥥 Object				
Parameter			П Атрибуты	• a ×
- Панель элементов			O Parameter Параметр	
Поиск			Annalis	2
Р 🔁 Ссылки			Агриоут	эначение
Погический объект				
Событие				
() bool	AND N. H. MIT.			
(1) int1	Record of the second se		<b>_</b>	
1 uint1	диаграмма Таолица			
@ int2	🚡 Журнал	- 4	×	
🕐 uint2	🖸 Ошибок: 0 1 Предупреждений: 0 🚯 Сообщений: 0 ⊥ 🗅 🐚	Поиск по журналу 🔎		
int4	Текст	Элемент Файл Проект		
10 uint4			Activate Window	
(B) int8	XVDHAL ODDERA		Go to Settings to acti	vate Windows.
2	V Hyperen augusta			
загрузка решения завершена	BCEro: 1,	из них выделено: 1		

**4.14** На уровне сервера для подключения OPC UA зададим уникальное имя точки подключения **Astra\_UA** для удобства дальнейшего отображения.

🖯 Demo_Dev - SePlatform.Development Stud	dio				,	× .						-	o ×
Файл Правка Вид Проект Помощь													
	8	🗇 local	• 🖽 🗄	≝ , ∝ , 😰	A & . 96	8 5 5 =	+ =  1	+ 11 +	+ = =	1111日中	÷		Systême platform
🖸 Обозреватель решений	• # ×	Server*	SandBox*	AstraServer* ☆ ×						-	😭 Свойства		<b>-</b> ↓ ×
Поиск	<i>p</i> -	C Demo D	Dev.Domain.EDU-	ASTRA, AstraServer	SePlatform.Server	Demo_Dev.omx		UaServer		- #	🕑 UaServer OPC UA Cer	рвер	
🔺 🚓 Demo_Dev	~	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1									Общие	_	^
4 💽 Demo_Dev											Имя точки	Astra_UA	
🔺 🦾 Файлы					Astra Applicati	==== 😤			LA Ae	CAE Ceoseo	подключения		
Demo_Dev.omx					G SandBox						Запретить клиентам	Her	~
4 🗁 Определения		(  F	HistoryModu	le 💷					T GR U	Server #	изменение сигналов		
4 Ba Domain			Модуль истор	unun					5 OF	C UA Cepser	Переводить в	Нет	
EDU-ASTRA				÷							состояние SUSPENDED		
Astraserver				-					Te Te	pServer mp	Mus	UsServer	
Astra Application			1	F					7_10	P Server	Constant and the second second	Gaberver	
HistoryModule		- interest									Сетевые параметра		1
TcpServer											порт ОРС ТСР	02544	
( UaServer											Dept UTTP sporteress	8080	
Ethornet Adapter	~										портнитепротокола	0000	v

Подсказка: чтобы посмотреть список серверов в Linux, которые связаны с SePlatform нужно запустить Терминал Fly и ввести команду - sudo systemctl list-units | grep seplatform

Конфигурация *SePlatformServer* на Linux машине готова. Но для того, чтобы развернуть эту конфигурацию, сначала нужно настроить XML-файлы **SePlatform.Domain.Agent** и **SePlatform.Net.Agent** на Wndows и Linux машинах.





# 5. Конфигурирование SePlatform.Domain

SePlatform.Domain представляет собой набор из двух служб, которые отвечают за то, чтобы развернуть конфигурации на определённой машине: SePlatform.Net.Agent и SePlatform.Domain.Agent. SePlatform.Net.Agent связывает данный компьютер с другими компьютерами в cemu SePlatform.Net. SePlatform.Domain отвечает за применение конфигурации определённой службы к определённому Systeme Platform серверу на каждой конкретной машине.

Файлы конфигураций находятся в папке:

- OC Windows: C:\Program Files\SePlatform\SePlatform.Domain
- OC Linux: /opt/SePlatform/SePlatform.Domain

Конфигурирование SePlatform.Domain на OC Windows

**5.1** Сначала перейдите в элемент **Определения** при помощи Обозревателя решений.



Выделите элемент Domain и в свойстве Адрес задайте CentralNode.

Demo_Dev - SePlatform.Development Studio	V		- 0 ×
Файл Правка Вид Проект Помощь			
	히 이 local · 프 프 육 육 육 영 원 원 원 부 부 비 표 바 부 비 두 빈 토 관 트 수 및		Systême platform
🕞 Обозреватель решений 🔷 🖣	X Server SandBox Demo_Dev ta X Domain ta Demo_Dev ta	🗢 😭 Свойства	• # ×
Поиск	<sup>™</sup> {) Demo Dev Cogepseimo e Mogylis = Demo Dev.omx • Domain	- Tomain SePlatform	m.Domain -
🔺 🚓 Demo_Dev		Общие	
🔺 🚰 Demo_Dev		Адрес	CentralNode
🔺 🦾 Файлы	الم	Имя	Domain
Bemo_Dev.omx	Councia His Om:System Councia His Om:Server Councia His Om:PLC		
<ul> <li>Определения</li> </ul>	Bell-Win Bell-Win Edu-		
· To Domain	de Ethernet		
a in EUO-ASIRA	<u>6</u>		
Aeserver			
Astra Application			
HistoryModule			
TcpServer			
🥑 UaServer		Appec Appec SePlatform.Do	main в сети
EthernetAdapter		Φ Ατριδντω	• a ×
С/ Панель элементов • 4	×	a Domain SePlatfor	n.Domain •
Р Настройки		Атрибут	Значение
Р Списания устройств		Комментарий	Домен - это описание
П Тип атрибута			среды исполнения,
D О Ссылка на модуль			которое используется
()) Пространство имён			конфигураций. Здесь
Аспект	100 % V 8F		будут описаны все
Пип приложения	Диаграмма Таблица		представлены в нашем
Физический тип	🔓 Журнал 👻 ң з	<	
	😮 Ошибок 1   Предупреждений: 0 🕕 Сообщений: 6 🔔 🐚 🗤 Поиск по журналу 🔎		
	Текст Элемент Файл Проект	A	
SePlatform Domain	😵 Ошибка применения к AstraServer по адресу local. Удаленный сервис вернул ошибку: Нет экземпляра ком		
	Журнал Формулы		
Применение к AstraServer по адресу local конфил	рации версии 1.0.0.d2 Всего: 4, из них выделено: 1		





#### **5.2** Задайте Linux машине адрес AstraNode.

Demo_Dev - SePlatform.Development Studio	× ·			- 0 ×
Файл Правка Вид Проект Помощь				
⊕ ⊕ • ↑ ℃ 🖓 🖻 🖫 🖨 ै	C local · 프 프 , 오 A A , 원 원 한 전 드 후 피 제 # 브 부 후	비료원로수,		Systême platform
🕞 Обозреватель решений 🔹 🖣 🗙	Server* SandBox* Domain* 12 X Domain 12 Demo_Dev 12	-	😭 Свойства	<b>-</b> ↓ ×
Поиск 🖉 🤟	🐇 <u>Demo_Dev, Domain</u> SePlatform.Domain 🗰 Demo_Dev.omx 🔹 EDU-ASTRA	- *	EDU-ASTRA Vsen SePlatform	.Domain •
	EDU-WIN (*) G. Sener Ethernet Adapter Ethernet Adapter		Ogune     Aspec     Aspec     Aspec     Aspec     Aspec     Aspec     Aspec     Aspec     Aspec     Aspec spas SeRatform.Domain s	eNode
🖓 Панель элементов 🔷 🖛 🗙			@ Атрибуты	- ù ×
Поиск 🔎 🗸			EDU-ASTRA Ysen SePlatform	.Domain •
Контроллеры			Атрибут Знач	чение
Ссылка на SePlatform.Domain			Комментарий Моя	BM Linux
Приложение				
Компьютер				
🦉 Узел SePlatform.Domain	100 % × WH			
Рабочее место	Диаграмма Таблица			
Ceть Ethernet	Га Журнал	• 1 ×		
Файловый обмен	Ошибок: 1 (). Преаупрежаений: 0 (). Сообщений: 6 4 (). 50	Поиск по журналу		
Последовательная шина	Текст Элемент Файл	Проект		
	Ошибка применения к AstraServer по адресу local. Удаленный сервис вернул ошибку: Нет экземпляра ком		Activate Windows	
	Журнал Формулы		Go to Settings to activate V	
Применение к AstraServer по адресу local конфигура	ции версии 1.0.0.d2 Всего: 3, из них выделено: 1			

После этого можете переходить к конфигурированию SePlatform.Domain.

**5.3** Установите Notepad++.Дистрибутив находится в сетевой папке \\gw\shared\se.platform\sw.addition)

→ ~ ↑ <mark>→</mark> >	Network >	gw > shared > se.platform	> sw.addition >			5 V	, Search sw.addition
	Name	^	Date modified	Туре	Size		
P Quick access	ope	en-sans	18.10.2022 18:23	File folder			
Desktop	* Pu	TTY	24.09.2022 12:00	File folder			
Downloads	* 😽 Far	30b5400.x64.20190523	12.08.2022 8:29	Windows Installer	9 812 KB		
Documents	* 🛞 Mo	odbusPollSetup64Bit	15.08.2022 11:51	Application	1 782 KB		
Pictures	*	w lext Document	13.10.2022 16:26	Text Document	0 KB		
h Music	😽 npr	p.7.9.2.Installer.x64	30.06.2022 10:55	Application	4 037 KB		
Videos	19 opt	en-canc	14.10.2022 10:07	Compressed (zipp	1 141 KB		
OneDrive	🧐 Wir	nSCP-5.21.2-Setup	12.08.2022 8:17	Application	11 202 KB		
This PC							
3D Objects							
Desktop							
Documents							
Downloads							
h Music							
Dicturer							
Fictures							

**5.4** Перейдите в папку с файлами конфигураций на Wndows машине(C:\Program Files\SePlatform\SePlatform.Domain), откройте файл

SePlatform.net.agent.xml с помощью Notepad++.

**5.5** Найдите строчку с тэгом **SePlatform.Net.Agent**. В атрибуте **Name** необходимо вписать содержимое в свойстве Адрес элемента *Domain* из DevStudio: CentralNode.

10	L>				
11					
12	SePlatform.Net.Agent	Name="CentralNode"	NetEnterPort="1010"	ParentAgentPort="1020">	
13	I I				





42

5.6 Уберите комментарий с блока <<u>ChildAgents></u>: выделите с 27 по 34 строчки (нажмите ПКМ и выберите Раскомментировать выделенное). В этом блоке описаны все дочерние узлы. В разрабатываемом проекте главным узлом является виндовая машина, а дочерним – линуксовая.



5.7 В атрибуте Name тэга ChildAgent введите содержимое свойства Адрес Linux машины,

25		
26	-	>
27		
2.8	¢.	<childagents></childagents>
29	Þ	<childagent fort="1020" name="AstraNode"></childagent>
30	1	<address "172.16.18"="" ,="" alac=""></address>
31		<address value="172.16.13.75"></address>
32	-	
33	-	
34		т
3.2		1





в атрибуте Address value укажите IP-адрес Linux машины. Дополнительно можно закомментировать строчку со вторым адресом, она нам не нужна



#### 5.8 Coxpanute SePlatform.net.agent.xml.

**Подсказка**: Notepad++ сохранит внесенные изменения после подтверждения прав администратора. После того как права были подтверждены нажмите «Сохранить» еще раз

#### **5.9** Откройте файл SePlatform.domain.agent.xml с помощью Notepad++.

**5.10** Найдите строчку с тэгом EntryPointNetAgent. В атрибуте Name необходимо вписать содержимое в свойстве Адрес элемента *Domain* из DevStudio: CentralNode.



В тэге <InstalledComponents> в атрибуте ServiceName описывается имя службы, которая обслуживает исполняющий компонент – элемент SePlatform.Server в проекте DeveloperStudio. Имя этой службы можно посмотреть в диспетчере задач во вкладке «Службы». Это же имя должно быть прописано в атрибуте ServiceName в файле SePlatform.domain.agent.xml.

r Task Manager File Options View				- 0	×
Processes Performance App	history	Startup Users Details Services			
Name	PID	Description	Status	Group	^
SePlatform.LicenseServer.A	2908	SePlatform.LicenseServer.Agent	Running		
keplatform.net.agent	2880	seplatform.net.agent	Running		
SePlatform.Server	2888	SePlatform.Server	Running		
<ul> <li>SessionEnv</li> <li>SgrmBroker</li> <li>SharedAccess</li> </ul>	5344	Remote Desktop Configuration System Guard Runtime Monitor Bro Internet Connection Sharing (ICS)	Stopped Running Stopped	netsvcs	~

Fewer details Que Open Services





Тэг Components внутри <Server> описывает установленные компоненты, работающие в рамках роли сервера. Атрибут Name должен содержать имя элемента SePlatform.Server в проекте в DeveloperStudio. То есть служба SePlatform.Server обслуживает исполняющий компонент Server внутри узла, характеризующего виндовую машину.



## **5.11** В атрибуте Name укажите «Server\_Central» в качестве Alias

26	2	>			
27		<seplatform.server< th=""><th>Name="Server_Central"</th><th>ServiceName="SePlatform.Server"</th><th>DefaultActivation="1"/&gt;</th></seplatform.server<>	Name="Server_Central"	ServiceName="SePlatform.Server"	DefaultActivation="1"/>
28			1	17.1	

**5.12** Далее в коде находится строчка **Installed Name**, которая описывает связку службы с компонентом по имени. Ее название также необходимо изменить на Server\_Central

L	43	>				
L	4.4	<component< th=""><th>InstalledName="Server</th><th>_Central" Name="Server"</th><th>StorageLimitSize="0"</th><th>StorageLimitNum="0"/&gt;</th></component<>	InstalledName="Server	_Central" Name="Server"	StorageLimitSize="0"	StorageLimitNum="0"/>

#### Сохраните SePlatform.domain.agent.xml.

5.13 Откройте Диспетчер задач и перейдите на вкладку Службы. Здесь при помощи ПКМ перезапустите службы SePlatform.Net.Agent и SePlatform.Domain.Agent.

Теперь можете переходить к конфигурированию *SePlatform.Domain* на Linux машине.





## Конфигурирование SePlatform.Domain на OC Linux

## **5.14** Откройте папку хранения файлов конфигурации на Linux

Компьютер/Файловая система/opt/SePlatform/ SePlatform.Domain

🛅 SePlatform.Domain - Менеджер фай/	юв	_ 🗆 ×
Файл Вид Сеть Сервис Настройк	а Справка	
🛛 🖉 🗸 🔊 🗸 🔿 🖉 🗖 🗖	апки 🔻 🗊 Вид 🗸 📑 🔍	
Адрес Компьютер УФайловая система 3	opt > seviatform > seviatform.Domain	0
SePlatform.Domain	libtcp_transport.so	
>- Construction Server	seplatform.domain.agent	
>- 🚞 parsecfs	🔮 seplatform.domain.agent.xml	
>- 🚞 proc	seplatform.net.agent	
— 🚞 root	🟮 seplatform.net.agent.xml	
>- <mark>*</mark> run		
- 📒 sbin		
- srv		
>- 📒 sys		
>- 📒 tmp		
>- 🚞 usr		
>- <mark></mark> var		
∨- 🚰 Сеть		
>- 🏓 Samba		
√ distr		
Документация техническая		
>- Сторонние компоненты		
Linux		
>- Students		
>		
and a start and a store		
SePlatform.Domain		م 🔶 م
	SePlatform.D 🛌 se : bash — T	😔 🕁 🕼 🛯 0 16:58

5.15 Откройте файл SePlatform.net.agent.xml с помощью программы «Редактор Kate». Кликните ПКМ по файлу и выберите «Открыть с помощью» далее выберите «Из меню → Офис→ Редактор Kate после чего нажимаем «Да»(для удобства можно поставить отметку «Открыть по умолчанию») в конце нажимаем «Открыть».

🦳 SePlatform.Domain - Менеджер файл		- 🗆 ×
Файл Вид Сеть Сервис Настройка Сорона Сервис Настройка Сорона Сервис Настройка Сорона Сервис Настройка Сорона Сервис Настройка Сорона Сервис Настройка Сервис Настрои Сервис	Справка Выберите программу для открытия файлов д С × /opt/SePlatform/SePlatform.Domain/seplatform.net.agent.xml	Ø
<ul> <li>SePlatform.Domain</li> <li>SePlatform.Server</li> <li>parsecfs</li> <li>proc</li> <li>root</li> <li>ro</li> <li>sbin</li> <li>srv</li> <li>sys</li> <li>tmp</li> <li>usr</li> <li>var</li> </ul>	Гарриложение  Firefox  Second Promium	
<ul> <li>Ceris</li> <li>Samba</li> <li>distr</li> <li>Документация техническая</li> <li>Сторонние компоненты</li> <li>Linux</li> <li>Students</li> <li>swaddition</li> <li>vie down</li> </ul>	<ul> <li>Установить по умолнании</li> <li>Все приложения</li> <li>Другая</li> <li></li></ul>	م ٢ م
	SePlatform.D 🚬 se : bash — Т 🂋 Выберите п	😔 🕁 🕼 en 🛛 17:02





emo_Dev.omx  EDU-ASTRA  EDU-ASTRA  EDU-ASTRA  EDU-ASTRA  Cet. Ethernet  Ethernet  EthernetAdapter  Ethernet	STRA ■ EDU-ASTRA Ysen SePlatform.Domain Appec AstraNode //Ma EDU-ASTRA @ Atpu6yttel EDU-ASTRA @ Atpu6yttel EDU-ASTRA Ysen SePlatform.Domain Atpu6yttel EDU-ASTRA Ysen SePlatform.Domain Atpu6ytel Mon BM Linux	e_Dev.omx   EDU-ASTRA  EDU-ASTRA  EDU-ASTRA  EDU-ASTRA  EDU-ASTRA  Cens Ethernet  EDU-ASTRA  EDU-ASTRA EDU-ASTRA  EDU-ASTRA EDU-ASTRA EDU-ASTRA EDU-ASTRA EDU-ASTRA EDU-ASTRA EDU-ASTRA EDU-ASTRA EDU	mo_Dev.omx       • EDU-ASTRA         • EDU-ASTRA       • • • • • • • • • • • • • • • • • • •	Demo_Dev.omx   EDU-ASTRA	lemo_Dev.omx   EDU-ASTRA  Ethernet  Cere Ethernet  EthernetAdapter  Ethe	Domain ユ Demo_Dev ユ		🗢 🔚 Свойства	▲ 前 ×
Cere Ethernet		Image: Eburnet       AstraServer         Image: EburnetAdapter       Image: EburnetAdapter         Image: EburnetAdapter       Image: EburnetAdapter      <	Cere Ethernet       Image: EthernetAdapter         Image: EthernetAdapter       Image: EthernetAdapter         Image: EthernetAdapter	Construction     C	Image: Cere Ethernet       Image: AstraNode         Image: Cere Ethernet       Image: AstraNode         Image: Cere Ethernet       Image: Cere Ethernet         Image: Cere Ethernet<	Demo Dev.omx	- EDU-ASTRA	🗸 🔆 🕘 EDU-ASTRA Узел	SePlatform.Domain •
Appec AstraNode Visca Ethernet Certs Ethernet B EDU-ASTRA B EDU-AS	Appec AstraNode Имя EDU-ASTRA	Agpec AstraNode VMa EDU-ASTRA Cers Ethernet EthernetAdapter EthernetAdapter EthernetAdapter EthernetAdapter Max EDU-ASTRA Mon SP Laboration AstraNode VMa EDU-ASTRA Mon SP Laboration Mon SP Linux Mon SP Linux	Agree AstraNode Visa EDU-ASTRA AstraServer Ethernet EthernetAdapter EthernetAdapter EthernetAdapter EthernetAdapter EDU-ASTRA EDU-ASTRA EDU-ASTRA EDU-ASTRA Cens Ethernet AstraNode Visa EDU-ASTRA EDU-ASTRA EDU-ASTRA EDU-ASTRA AstraNode Visa EDU-ASTRA EDU-ASTRA EDU-ASTRA AstraNode Visa EDU-ASTRA EDU-ASTRA AstraNode Visa EDU-ASTRA EDU-ASTRA EDU-ASTRA AstraNode Visa EDU-ASTRA EDU-ASTRA EDU-ASTRA AstraNode Visa EDU-ASTRA EDU-ASTRA EDU-ASTRA EDU-ASTRA EDU-ASTRA EDU-ASTRA EDU-ASTRA Mos BM Linux	Appec AttraNode Uwa EDU-ASTRA Uwa EDU-ASTRA Wwa	Agpec AstraNode Viva EDU-ASTRA Cere Ethernet Cere Ethernet Cere Ethernet Adapter Ethernet Mos BM Linux			🔿 Общие	
Cerb Ethernet	<ul> <li>EDU-ASTRA</li> <li>EDU-ASTRA</li> <li>EDU-ASTRA</li> <li>EDU-ASTRA</li> <li>EDU-ASTRA</li> <li>EDU-ASTRA Узел SePlatform.Domain</li> <li>Атрибуть</li> <li>EDU-ASTRA Узел SePlatform.Domain</li> <li>Атрибут</li> <li>Значение</li> <li>Комментарий</li> <li>Мов ВМ Linux</li> </ul>	Image: EDU-ASTRA       Image: EDU-ASTRA         Image: EDU-ASTRA       Image: EDU-ASTRA         Image: EthernetAdapter       Image: EthernetAdapter         Image: EthernetAdapter       Image:	Ethemet       Image: Ebu-ASTRA         Cens Ethemet       Image: EthemetAdapter         Image: EthemetAdapter       Image: EthemetAdapter         <	Ethernet       Image: Cers Ethernet         Image: Ethernet       Adapter	Ethernet       Image: Cers Ethernet         Image: Cers Ethernet	$\wedge$		Адрес	AstraNode
	@ Атрибуты ▼ Ф × ■ EDU-ASTRA Узел SePlatform.Domain ▼ Атрибут Значение Комментарий Моя BM Linux	Силентерчалист В Атрибуты – Ф Х EDU-ASTRA Узел SePlatform.Domain – Атрибут Значение Комментарий Мов ВМ Linux	Солентериариев ВОU-ASTRA Узел SePlatform.Domain Атрибут Значение Комментарий Моя ВМ Linux	Силепер-харива Ш Солепер-харива Ш DU-ASTRA Узел SePlatform.Domain - Атрибут Значение Комментарий Моя ВМ Linux	Шетенскийций         Шетенскийций         ВDU-ASTRA Узел SePlatform.Domain         Атрибут       Значение         Комментарий       Моя BM Linux	Cerь Ethernet	EDU-ASTRA      AstraServer	Имя	EDU-ASTRA
	@ Атрибуты     < д ×       @ EDU-ASTRA Узел SePlatform.Domain     -       Атрибут     Значение       Комментарий     Моя BM Linux	@ Атрибуты ► 4 × EDU-ASTRA Узел SePlatform.Domain ► Атрибут Значение Комментарий Моя BM Linux	@ Атрибуты	Шарибуты         • Ф ×           EDU-ASTRA Узел SePlatform.Domain         •           Атрибут         Значение           Комментарий         Моя ВМ Linux	@ Атрибуты • Ф : EDU-ASTRA Узел SePlatform.Domain Атрибут Значение Комментарий Моя ВМ Linux				
	@ Атрибуты     + 4 ×       @ EDU-ASTRA Узел SePlatform.Domain     +       Атрибут     Значение       Комментарий     Моя BM Linux	@ Атрибуты        4       4         @ EDU-ASTRA Узел SePlatform.Domain           Атрибут       Значение          Комментарий       Моя ВМ Linux	Ф Атрибуты       • 0 ×         EDU-ASTRA Узел SePlatform.Domain       •         Атрибут       Значение         Комментарий       Моя ВМ Linux	Ф. Атрибуты     • 4 ×       EDU-ASTRA Узел SePlatform.Domain     •       Атрибут     Значение       Комментарий     Моя ВМ Linux	@ Атрибуты     • р. :       III EDU-ASTRA Узел SePlatform.Domain       Атрибут     Значение       Комментарий     Моя BM Linux				
(2) Атрибуты – 0.	EDU-ASTRA Узел SePlatform.Domain     Атрибут Значение     Комментарий Моя BM Linux	EDU-ASTRA Узел SePlatform.Domain - Атрибут Значение Комментарий Моя BM Linux	EDU-ASTRA Узел SePlatform.Domain     -       Атрибут     Значение       Комментарий     Моя BM Linux	EDU-ASTRA Узел SePlatform.Domain       -         Атрибут       Значение         Комментарий       Моя BM Linux	Ш EDU-ASTRA Узел SePlatform.Domain         Атрибут       Значение         Комментарий       Моя BM Linux			@ Атрибуты	• # X
EDU-ASTRA Ysen SePlatform.Domain	Атрибут Значение Комментарий Моя BM Linux	Атрибут Значение Комментарий Моя BM Linux	Атрибут Значение Комментарий Моя ВМ Linux	Атрибут Значение Комментарий Моя 8M Linux	Атрибут Значение Комментарий Моя 8M Linux			📗 EDU-ASTRA Узел	SePlatform.Domain +
Атрибут Значение	Комментарий Моя BM Linux	Комментарий Mos BM Linux	Комментарий Mos BM Linux	Комментарий Моя BM Linux	Комментарий Моя BM Linux			Атрибут	Значение
Комментарий Mos BM Linux								Комментарий	Mos BM Linux

**5.16** В атрибуте Name тэга <SePlatform.Net.Agent> введите содержимое свойства Адрес линуксового узла в среде разработки *DevStudio*.

Дочерние агенты прописывать ненужно.

- **5.17** Сохраните файл **SePlatform.net.agent.xml** и закройте.
- 5.18 Откройте файл SePlatform.domain.agent.xml с помощью редактора Kate. Здесь в атрибуте Name тэга EntryPointNetAgent введите содержимое свойства Адрес линуксового узла в среде разработки *DevStudio*.

В атрибуте ServiceName тэга <InstalledComponents> описывается имя службы, которая обслуживает исполняющий компонент – элемент SePlatform.Server в проекте в DeveloperStudio. Эта служба в OC Linux называется seplatform-server.service.





Тэг **Components** внутри <Server> описывает установленные компоненты, работающие в рамках роли сервера. Атрибут Name должен содержать имя элемента SePlatform.Server в проекте DeveloperStudio.

sepia	trorm.comain.agent.xmi * — Kate	
йл І	Правка Вид Закладки Сеанс Сервис Настройка Справка	
R	seplatform.domain.agent.xml 🛛 😣	a 🛛
:</td <td>xml version="1.0" encoding="utf-8"?&gt;</td> <td>`:Widea</td>	xml version="1.0" encoding="utf-8"?>	`:Widea
<b>▼</b> <s< td=""><td>ePlatform.Domain.Agent Name="NDA"&gt; <!-- Name - имя доменного агента (опционально)--></td><td>Lingan</td></s<>	ePlatform.Domain.Agent Name="NDA"> Name - имя доменного агента (опционально)	Lingan
▼ > > > > >	  EntryPointNetAgent – точка доступа Net-Arента. > Name – agpec (Name) Net-areнта, предназначенного для взаимодействия "внешнего" мира с данным > одоченным агентом (входящие запросы от клиентов, административного приложения в трее и т.д.) > Address – IP-agpec или имя хоста точки доступа (опционально, по-умолчанию подразумевается IP-agpec машины, где запущена рабочая станция)	in the second se
	» - Port — - номер порта точки доступа (значение NetEnterPort в конфигурации Net-Агента).	
	ContryPointNetAgent Name: "AstraNode" Address="127.0.0.1" Port="1010"/>	
▼ > > >	<1 > Ter InstalledComponents описывает перечень установленых экземпляров компонентов на машине, где работает данный доменный агент. > Злементы из этого перечня "маппятся" на экземпляры компонентов в реальных ролях доменного агента.	
<b>↓</b> ≥	> (InstalledComponents)	
<ul> <li>A</li> <li>A</li> <li>A</li> <li>A</li> <li>A</li> <li>A</li> <li>A</li> <li>A</li> <li>A</li> </ul>	< <br <li></li>	
	> SePlatform.Server Name="AstraServer" ServiceName="seplatform.server.service" DefaultActivation="1" />	
•	Coerver / (I dubabilitier al enig puille cepteral,/	
•	< Ізг Components onucubaer ucraновленные конпоненты, работающие в ранках роли сербера <components cache="" domainstorage="" jusr="" local="" server"="" storagepath=""> <!-- StoragePath nyть к nanke,_B<br-->которую будут сохраняться конфигурации,</components>	
Стро	ка: 28. столбен: 45 ВСТАВКА Отступы: 4 У ЦТЕ-8 У XML	v F
		,
Ц Г	ООИСК И ЗАМЕНА	
7 🞚	🇉 🦳 🖳 🔁 SePlatform.D., 🔽 se : bash — T., 📝 seplatform.d. 🚓 🖓 🕼 🕼 🛯	19:11
		1911

То есть служба **SePlatform-server.service** обслуживает исполняющий компонент **AstraServer** внутри узла, характеризующего линуксовую машину.







**5.19** Необходимо повторить связку с сервисами. Ниже в коде в InstalledName и Name необходимо сменить имя на AstraServer

<b>P</b>	seplatform.domain.agent.xml
	CinstalledComponents> CinstalledComponents> SinstalledComponents> Tar – наименование типа компонента (на текущий момент поддерживается: SePlatform.Server) Name – иня установленного экземпляра компонента, по котороми Будет разрешаться маплинг в списке компонентов в конкретных ролях ServiceName – иня сервиса, под которым зарегистрирован экземпляр компонента
	"1" - Компонент Всегда перезапускается; ="0" - Компонент перезапускается; ="0" - Компонент перезапускается, только если он находился в состоянии "Выполняется" в момент принятия конфоигурации. > <b>GSePlatform.Server</b> Name="AstraServer" ServiceName="seplatform.server.service" DefaultRctivation="1" />
• -	⟨Server⟩ Добавляет агенту роль сервера
	<1 Тэг Components описывает цстановленные компоненты, работающие в рамках роли сервера> <b>Components StoragePath="/usr/local/DomainStorage/cache/server"&gt;</b> <1 StoragePath - путь к nanke,_B которую Будут сохраняться конфигурации, построенные для локальных экземпляров БеPlatform.Server. Указанная папка должна существовать в файловой системе>
• > >	» » <br » » InstalledName – имя установленного экземпляра компонента (из перечня установленных экземпляров компонентов InstalledComponents)
	» Name – имя данного компонента в проекте автоматизации » » StorageLimitSize – при превышении общего размера (Байт) конфигураций, имеющие наиболее раннюю
	метку оренена одудт удаляться. >>>> StorageLimitNum - при превышении общего количества конфигураций, имеющие наиболее раннюю метку
	оренено одууг удолятося. >>>> Необязательные параметры со значением по умолчанию "0", при котором функция неактивна
	<pre>Component InstalledName= RstraServer" Name: "AstraServer" StorageLimitSize="0" StorageLimitNum="0"/&gt;</pre>
	< Ter Ontions - Danametra Dafota>
трок	ка: 45, столбец: 69 ВСТАВКА Отступы: 4 🗸 UTF-8 🗸 XML 🗸

**Подсказка:** Ниже приведены два рисунка из краткого конспекта пользователя на которых указаны взаимосвязи проекта DevStudio с редакцией файлов xml, а также названиями сервисов в Windows и Linux:









- **5.20** Сохраните файл **SePlatform.domain.agent.xml**. Вернитесь в командную строку
- 5.21 Находясь в командной строке машины с ОС Linux, введите команды для перезапуска служб SePlatform.Net.Agent и SePlatform.Domain.Agent:

sudo systemctl restart seplatform.net.service sudo systemctl restart seplatform.domain.service

#### 5.22 И для просмотра статуса служб SePlatform.Net.Agent и SePlatform.Domain.Agent: sudo systemctl list-units –type service

📰 se : sudo — Терминал Fly	$\checkmark$	_ 🗆 ×
Файл Правка Настройка Справка		
🗳 🎴 📋 📋 k 🗸 🏈		
UNIT	LOAD ACTIVE SUB	DESCRIPTION
acpi-support.service	loaded active exited	LSB: Start some power management scripts
acpid.service	loaded active running	ACPI event daemon
avahi-daemon.service	loaded active running	Avahi mDNS/DNS-SD Stack
console-setup.service	loaded active exited	Set console font and keymap
cron.service	loaded active running	Regular background program processing dae
dbus.service	loaded active running	D-Bus System Message Bus
fly-dm.service	loaded active running	The FLY login manager
getty@tty1.service	loaded active running	Getty on tty1
keyboard-setup.service	loaded active exited	Set the console keyboard layout
kmod-static-nodes.service	loaded active exited	Create list of required static device nod
libflygetexe-bin.service	loaded active running	The FLY get exec service
networking.service	loaded active exited	Raise network interfaces
NetworkManager-wait-online.service	loaded active exited	Network Manager Wait Online
NetworkManager.service	loaded active running	Network Manager
ofono.service	loaded active running	oFono Mobile telephony stack
parlogd.service	loaded active running	PARSec events logging daemon
parsec.service	loaded active exited	Intitialize Parsec Subsystem
polkit.service	loaded active running	Authorization Manager
quota.service	loaded active exited	Initial Check File System Quotas
royolog.ocrvice	loaded active running	System Logging Service
seplatform.domain.service	loaded active running	SePlatform.Domain Agent
seplatform.net.service	loaded active running	SePlatform Net Agent
sepialionm.server.service	ioaded active running	Serialiurm.Server
slapd.service	loaded active running	LSB: UpenLDHP standalone server (Lightwei
ssh.service	loaded active running	UpenBSU Secure Shell server
systemd-journal-flush.service	loaded active exited	Flush Journal to Persistent Storage
systemd-journald.service lines 1–28	loaded active running	Journal Service
<b>2</b> 1		2
👻 🖭 🗂 🖨 🛄 2	itform.D 🛌 se : sudo — T	📝 seplatform.d 😔 🗗 🖓 🛛 EN 💿 19:40





**5.23** Те же действия нужно повторить для Windows: перезапустить службы domain и net agent

Processes Performance App	history	Startup Users	Details Services			
Name SensorService	PID	Description Sensor Servic Sensor Monit	e toring Service	Status Stopped Stopped	Group LocalSystemN LocalServiceA	^
seplatform.domain.agent	8732 2908 8760 9596	seplatform. SePlatform. seplatform. SePlatform.	Start Stop	Punning unning unning		
Fewer <u>d</u> etails   🍓 Open	Services		Open Services Search online Go to details			

Теперь сервисы перезапущены и взаимодействуют с актуальными версиями xml файлов





# 6.Подключение к OPC UA Server Linux машины

#### **6.1** Откройте Журнал приложений (Пуск $\rightarrow$ Se Platform $\rightarrow$ Service – Log Viewer)

В нем можно увидеть сообщение, что соединение seplatform.net.agent с дочерним узлом произошло успешно, т.е связь между BM Windows и BM Linux есть

**Подсказка:** Для удобства просмотра сообщений можно настроить параметры фильтрации при помощи ПКМ → Фильтровать (либо F4)

Источник	Время	Сообщение	^	Настройка фил	ьтра	
eplatform.net.agent	29.11.2022 13:20:33	Установлено соединение с дочерним узлом 10.0.0.50 : 1020 (ASTRANODE)		Micros	oft-Windowe-	estartManager
eplatform.net.agent	29.11.2022 13:14:04	Разорвано соединение в точке входа для: 127.0.0.1 : 50064	Cox	ранить в файл	F2	er Profiles Service
eplatform.net.agent	29.11.2022 13:13:59	Новое входящее соединение в точке входа: 127.0.0.1 : 50064	Oun	TROBATH	F4	insrv
eplatform.net.agent	29.11.2022 13:13:59	Новое входящее соединение в точке входа: 127.0.0.1 : 52223		orposoro		MI
eplatform.net.agent	29.11.2022 13:13:59	Разорвано соединение в точке входа для: 127.0.0.1 : 63370	Kon	провать	Ctrl+C	
eplatform.net.agent	29.11.2022 13:13:59	Разорвано соединение в точке входа для: 127.0.0.1 : 63357	Най	ги текст	Ctrl+F	
eplatform.net.agent	29.11.2022 13:13:59	Разорвано соединение с дочерним узлом 10.0.0.50 : 1020 (ASTRANODE)	Пер	ейти ко времени		
eplatform.net.agent	29.11.2022 10:28:35	Установлено соединение с дочерним узлом 10.0.0.50 : 1020 (ASTRANODE)	Over	стить журнал	Пробел	
eplatform.net.agent	29.11.2022 10:28:17	Разорвано соединение с дочерним узлом 10.0.0.50 : 1020 (ASTRANODE)		cinite adjuntan		
eplatform.net.agent	28.11.2022 19:59:08	Новое входящее соединение в точке входа: 127.0.0.1 : 63370	Спр	авка	F1	lent
eplatform.domain.agent	28.11.2022 19:59:08	Приложение запущено.		- Jecum	Center	-
eplatform.domain.agent	28.11.2022 19:59:08	Прочитали конфигурацию агента из файла: C:\Program Files\SePlatform\SePlatform.Domain\seplatform.domain.agent.xml		✓ seplatfe	orm.domain.ag	lent
eplatform.domain.agent	28.11.2022 19:59:08	Запуск приложения		SePlatf	orm.LicenseSei	ver.Agent
eplatform.domain.agent	28.11.2022 19:59:08	Уровень логирования: 2		Seplation Seplation	orm.net.agent	
eplatform.domain.agent	28.11.2022 19:59:08	seplatform.domain.agent v.1.0.6.4 02/10/2022 22:29		C Seplat	orm.server	
eplatform.domain.agent	28.11.2022 19:59:06	Приложение остановлено.			Server Aes	erver
eplatform.net.agent	28.11.2022 19:59:06	Разорвано соединение в точке входа для: 127.0.0.1 : 63356			orm Server Hist	erver on/Module
eplatform.domain.agent	28.11.2022 19:59:06	Получена команда завершения работы.		SePlatf	orm Server Uas	enver
eplatform.net.agent	28.11.2022 19:58:16	Новое входящее соединение в точке входа: 127.0.0.1 : 63356		Softwa	e Protection P	latform Service
eplatform.net.agent	28.11.2022 19:58:16	Новое входящее соединение в точке входа: 127.0.0.1 : 63357		System	Restore	
eplatform.net.agent	28.11.2022 19:58:16	Установлено соединение с дочерним узлом 10.0.050 : 1020 (ASTRANODE)		tvnserv	er	
eplatform.net.agent	28.11.2022 19:58:16	TCPNetEnterPoint: принимаем входящие соединения. Порт: 1010		VMToo	ls	
eplatform.net.agent	28.11.2022 19:58:16	TCPNetEnterPoint: пытаемся слушать порт 1010		VMUpg	radeHelper	
eplatform.net.agent	28.11.2022 19:58:16	ParentAgent: принимаем входящие соединения. Порт: 1020		VSS		
entablemen and examt	30.11.3033.10.50.15	Description of the second seco	v	Window	us Error Report	ina

# **6.2** В элементе *AstraServer* внутри линуксового узла *astra* в OPC UA Сервер в свойстве «Имя точки подключения» должно бфть задано имя «Astra\_UA»

	401 M 1 1 2			O plan
С Обозреватель решений	• 4 ×	AstraServer 🕫 🗙	Свойства	- q
Поиск	ρ-	🔹 Demo_Dev.Domain.EDU-ASTRA_AstraServer_SePlatform.Server 📫 Demo_Dev.omx 💌 UsServer 💌	UaServer OPC UA Ce	рвер
🖡 🛃 Demo_Dev			• Общие	í
		Actes Andrest at S	Има точки подключения	Astra_UA
			Запретить клиентам изменение сигналов	Нет
			Переводить в состояние SUSPENDED в резерве	Нет
		TCP Server	Имя	UaServer
			• Сетевые параметри	ы
			Порт ОРС ТСР протокола	62544
			Порт НТТР протокола	8080
			Использовать НТТР протокол	Нет
			Использовать НТТР протокол @ Атрибуты	нет
			Использовать НТТР протокол @ Атрибуты @ UaServer OPC UA Ce	нет • Ф
			Использовать НТТР протокол @ Атрибуты У UaServer OPC UA Ce Атрибут	нет • Ф рвер Значение
<sup>(5)</sup> Панель элементов	- a ×		Использовать НТТР протокол Ф Атрибуты ИаServer OPC UA Ce Атрибут	нет рвер Значение
Г/) Панель элементов Покск	<b>× ۹ ×</b> - م		Использовать НТТР протокол @ Атрибуты @ UaServer OPC UA Ce Атрибут	Нет • Ф рвер Значение
<mark>б<sup>0</sup> Панель элементов</mark> Помск ≰ ⊆а ВАСпеt	× ¢ ×	100 % *) 881 Лаготана Табана	Использовать НТТР протокол @ Атрибуты @ UAServer OPC UA Ce Атрибут	нет • ф значение
R <sup>(2)</sup> Панкль элементов Поиссе 1 ⊂ 26 ACnet 1 ⊂ 26 TherNet/IP	<b>۰ ۹ ×</b> - م	100 %, * ) 888 Дигоромка Таблица Таблица	Использовать НТТР протокол @ Атрибуты @ UaServer OPC UA Ce Атрибут	Нет • Ф реер Значение
<ul> <li>Панель закиентов</li> <li>Полос</li> <li>I all BACnet</li> <li>I all BACnet</li> <li>I all BACnet</li> <li>I all BACnet</li> </ul>	• <b>4 x</b> \$	100 % ~ 881 Догурама Су Хурна ~ 8 >	Использовать НТР протокол @ Атрибуты @ UaServer OPC UA Ce Атрибут	нет • а реер Значение
C) Панила Злементов Поисс 1 ag B4/Let 3 ag B4/Let 3 ag B4/Let 3 ag 04/Let 4 ag 04/Let 4 ag 04/Let	• <b>4 x</b> \$\$	100 % * ) 888 Докурана (2) Ошибос 0 1 Предупрежаений. 0 (О) Сообщений. 0 1 (Р) 50 Решск по хурналу (Р)	использовать HTP протокой @ Arpuбуты @ UsServer OPC UA Ce Атрибут	нет • а реер Значение
C Tainers's Serventoe  Touce  Touce Touce Touce Touce  Touce  Touce  Touce  Touce  Touce  Touce  Touce  Touce  Touce  Touce  Touce Touce Touce Touce Touce Touce Touce  T	<b>+ 4 x</b> - م	100 % × 368 Диаграния Сд. Журнал С Ошибок 0 1/1 Предпреженний: 0 0 Сообщений: 0 1 1 10 № 10 Понск по журналу ро Текст Влемент Файл Проест	Использовать HTP протокой @ Атрибуты @ UaServer OPC UA Ce Атрибут	нет ураср Значение
C         Панель заячентое           Панель         36 Аслекt           i         B ACnett           i         M ACN Source           i         M ACN Source           i         M ACN Source           i         M ACN Source	• <b>4 x</b> \$2	100 % ~ ) Вії Дигарання Таблица Су Хурнал ~ а > С Ошибос 0 // Предупреждений: 0 0 Сообщений: 0 1 0 % Текст Элемент Фала Просет	использовань НТР протокой Фарибуты Ф Избегчег ОРС UA Се Агрибут	нет ряср Значение
Панель злементов           Поисс           Image: Comparison of the second	• <b>4 x</b>	100 %, * ) 201 Дограма Таблица Догурама Таблица До Хорма • а х 10 Ошибос 0 1 Предупреждений. 0 0 Сообщений. 0 1. 10 № 0 Понск по мурналу р Текст Элечент байл Проест	использовань НТР протокой Ф Агрибуты Ф Избегчет ОРС UA Ce Агрибут	нет реор Злачение
Control         Descention           Rosce         Image: Control           Image: Control         Image: Control	× # × - q	100 % ~ 341 Диаграмка Сар Журнал С Ошибок 0 / 1 Предупреждений: 0 / Сообщений: 0 / 1 Ро 100 Понск по журналу р Текст Элемент Файл Прост	Montandean HTP mportona Arputyrni Ø Juderner OPC UA Ce Arputyr	Her peep 3ravenue
Панель злечентов           Полос           Image: Comparison of the service of t	• 4 х <i>Р</i> = ^ Сожения	100 % ~ 1 888 Дигограмая Таблица Су Хурная - а > СО Quedoce 0 // в Предпрежаений 0 ОС Сообщений 0 Ц Со № Ранск по журналу до т Тект Элемент Файл Проест	Monandearn HTP mportoad Arpedyne Usserver OPC UA Ce Arpedyn Arpedyn Activiate Windo	Her prop Judvenue





		iiiii		3	
6.3	Постройте решение		, перейдите к развёртыванию		и примените
	конфигурацию к линуксово	му серве	epy.		

...

AstraServer	1.0.0.d3 🎮	от 🖌	(9)	

-

**6.4** Теперь в папке конфигурации на рабочем столе появились две подпапки с файлами конфигурации для Windows и Linux:

- 1	Name	^	Date modified	Туре	Size
	EDU-ASTRA		24.11.2022 17:32	File folder	
Я	EDU-WIN		16.11.2022 10:35	File folder	

теперь есть возможность загружать конфигурации для обоих серверов

**6.5** Откройте OpcExplorer, введите IP-адрес Linux машины, подтвердите ввод нажатием клавиши Enter и нажмите на иконку монитора слева – Запрос проверки связи.

Проект	Помощь				
1- 2	10.0.0.50			•	Ø
атель		•	ņ	х	0
			ρ	Ŧ	1
- H	and the data set of the second	0			

**6.6** Нажмите на кнопку «Добавить источник» и введите данные для подключения как на изображении ниже (в соответствии с проектом DevStudio), нажмите OK.

Добавить источни	ик	×
Спецификация	🕑 UA (tcp)	•
Сервер	Astra_UA	
Порт		62544
	(	ОК Отмена





Internal

53

6.7 В поле «Источник данных» выберите нужный источник и подключитесь к нему (кнопка

«Подключиться к исполняющему компоненту»

**6.8** Добавьте в окно Инспектор *Parameter* из объекта *Sandbox*. При изменении значения сигнала, меняется и качество. Подробная информация описана в документации (Документация → SePlatform.Server → Сигналы SePlatform.Server).

).

Мы можем увидеть, что источником нового сигнала является Linux сервер







# 7. Модификация проекта SePlatform.DevStudio

# Работа с атрибутами

Познакомимся атрибутами объектов и параметров.

7.1 Перейдите в *SandBox* при помощи Обозревателя решений DevStudio. Перетяните сюда *Логический объект* 

из Панели элементов. Назовите его Calculator.

Demo_Dev - SePlatform.Development Studio Файл Правка Вид Проект Помощь			- 0 ×
0 0 · 1 0 10 8	0 C loal		Système
🕞 Обозреватель решений 🔹	a x SandBox" + X	🗢 😭 Свойства	• ‡ ×
C Obcoperations permitted Forcex A G Demo_Dev A G Demo_Dev A G Demo_Dev A G Demo_Devomx A G Devomx A G Devomx	X Sandbox**X     Personeter     Calculator     Calculator     Calculator     Calculator	<ul> <li>              € Свяйства          </li> <li>             Севиста         </li> <li>             Собщие         </li> </ul> <li>             Уровень доступа         </li> <li>             Область доступа         </li> <li>             Собласть доступа         </li> <li>             Канана         <ul> <li>             Добласть доступа         </li> </ul> </li> <li>             Добласть доступа         <ul> <li>             Добласть доступа         </li> <li>             Добласть доступа         </li> </ul> </li> <li>             Добласть доступа         <ul> <li>             Добласть доступа         </li> <li>             Добласть доступа         </li> </ul> </li> <li>             Добласть доступа         <ul> <li>             Добласть доступа         </li> </ul> <ul> <li>             Добласть доступа         </li> </ul> <ul> <li>             Добласть доступа         </li> </ul> <ul> <li>             Добласть доступа         </li> <li>             Добласть доступа         </li> <li>             Добласть доступа         </li> </ul> </li> <li>             Добласть доступа         <ul> <li>             Добласть доступа         </li>             Добласть доступа         </ul></li> <li>             Добласть доступа         <ul>             Добласть доступа         </ul></li> Добласть доступа <li>             Добласть доступа         <ul>             Добласть доступа         </ul></li> Добласть доступа	
Панель элементов	a x 100% * HE	Атрибут	Значение
Поиск	р - Диаграмма Таблица		
1 () Om.PLC	🕞 Журнал 👻 🕫	×	
() Om.Server	🖸 Ошибок 0 1 Предупреждений: 0 🚯 Сообщений: 0 🛓 🖸 мр. Поиск по журналу 🔎		
Вычисления     Вычисления     Событие     Логический объект     Событие     Событие     Параметр     Собет	Текст Элемент Файл Проект	Activate Wind	
		Go to Settings to a	
	Журнал Формулы		
Загрузка решения завершена	Bcero: 2, из них выделено: 1		10.40
P Type here to search	A 🖾 O 🛱 📙 🐖 🖳 🔍 💶 🖃	~ ê 달	teng 10:40

7.2 Перейдите внутрь и добавьте сюда *З параметра* типа Uint4: Var\_a, Var\_b, Result.







#### 7.3 Добавьте атрибут Описание и Начальное значение для Var\_A:

о выделите *Var\_A*, кликните ПКМ по Панели атрибутов → Добавить новые атрибуты



 $\circ$  выберите атрибут Om.System  $\rightarrow$  Attributes  $\rightarrow$  Описание



о Вописании пропишите «Value A».







#### 7.4 Добавьте для *Var\_b* и *Result* атрибут описания: Value B и Result.

**Подсказка**: можно сразу добавить атрибуты сразу к нескольким параметрам. Для этого надо выделить их, ПКМ → добавить атрибуты. Так же можно написать общую часть названия

	Общие	
	Уровень доступа	публичный
	Область доступа	глобальная
→ <sup>1</sup> Var_b	Направление	вход
	Тип	uint4
→ <sup>®</sup> Result	Имя	
	@ Атрибуты	- a ×
		•
	Атрибут	Значение
00% v St	Описание	Value
диа рамма технице		

**7.5** Постройте решение, перейдите к развёртыванию и примените конфигурацию к линуксовому серверу.

EDU-ASTRA	_			
AstraServer	1.0.0.d4 🥕	1.0.0.d4 or 01.12.2022 11:10:36	$\checkmark$	\$

**7.6** Перейдите в *OpcExplorer*, добавьте в Инспектор весь объект *Calculator*. Здесь отобразилось начальное значение у параметра *Var\_A* и атрибуты описания для каждого из параметров

Ф., Инспектор 1 - SePlatform.Service - OP	CExplorer			_	·			-	×
Файл Вид Проект Помощь	•	I Astra_UA	• )	<b>€</b> •	7 ) (1		3		
Собозреватель	• ¤ ×	10.0.0.50 12	SandBox.Calculator.Var_	b th V	нспектор 1 🖶 🗙		-		Ŧ
Поиск	,o •	Тип	Сигнал	Значение	Качество	Время	Источник		
и 🚱 Astra_UA на 10.0.0.50		M SandBox.0	Calculator.Result	0	плохое: 32 - Waiting For Initial	01.12.2022 11:14:21	Astra UA		
D Server		SandBox.	Calculator.Var_a	123	хорошее: 192 - Good	01.12.2022 11:26:22	Astra_UA		
I D NativeTypes	-	SandBox.	Calculator.Var_b	0	плохое: 32 - Waiting For Initial	01.12.2022 11:14:21	Astra_UA		
SandBox									
P arameter									
S 777001									
Service     Service									
777001									

Таким образом в *DeveloperStudio* можно добавлять атрибуты не только параметрам, но и объектам.

Подсказка: можно очистить список нажав на иконку «ластик» , чтобы управлять отображаемыми параметрами можно их выделить, а затем добавить/убрать при помощи







# Работа с логикой

Теперь попробуем построить небольшую логику в проекте, научим объект *Calculator* вычислять сумму двух переменных разными способами. Вычисления сервер производит с помощью модуля логики. В данном модуле используется скриптовый язык SePlatform.OM. Язык SePlatform.Om позволяет описывать исполняемые элементы скриптов (процедуры и формулы), которые запускаются в процессе работы компонентов *SePlatformPlatform*. Скрипты можно создавать в процессе работы со следующими программными продуктами: *SePlatform.Server*, *SePlatform.HMI*, *SePlatform.DevStudio*. Подробное описание можно найти в документации в разделе **SePlatform.Om**.

В сервере выполнить операции можно различными способами. Один из способов – это заставить параметры самостоятельно вычислять свои значения.

**7.7** Выберите *параметр* Result и перейдите во вкладку Формулы (здесь указываются однострочные скрипты, не требующие точки с запятой на конце).

Damo Day - SaPlatform Dayalonment St.	udio	×		- 7	×
Calls Desers Rus Depart Development St	Jaio			- 0	~
	8	6 teal ·프플, 4 , 2 2 2 4 4 , 15 3 5 5 5 도수리 파우브 부수의 고린 태우		<b>6</b> S	Bystême
Обозреватель решений	• # ×	Calculator ** X Demo Dev 12	😭 Свойства		• # ×
Поиск	<i>,</i> 0 -	Demo Dev.Domain.EDU-ASTRA.AstraServer.Astra Application.SandBox.Calculator Логический объект      Demo Dev.omx      Result     .	<ul> <li>Result Параметр</li> </ul>		
4 💰 Demo Dev			Общие		
4 🔄 Demo_Dev			Уровень доступа	публичный	
🔺 🗁 Файлы		A) (D) Var_a	Область доступа	гаобальная	
Demo_Dev.omx		+>TE Result	Name and Activity of the second secon		
Oпределения		AJ (2) Var_b	паправление	вход	
4 🏜 Domain			Тип	uint4	
EDU-ASTRA			Имя	Result	
• Server • System			<ul> <li>@ Атрибуты</li> <li>Веки!! Парамето</li> </ul>		• 4 ×
			Aroufur	2	
		100 8	Оприбут	Desult	
Су Панель элементов	• • •	Паларания Таблица	Описание	Result	
	~				
🛛 🍠 Событие		jg Xypean ····································			
4 О Параметр		Ошибок: 0 ! Предупреждений: 0 О Сообщений: 3 ± 40 Поиск по журналу Р -			
0					
bool     int1		Текст Элемент Файл Проект			
0 bool 0 int1		Текст Злемент Файл Проект Компиляция Demo, Dev была успешно выполнена ранее.			
bool     int1     uint1     int2					
) bool () int1 () int2 () int2 () uint2		Tecr         Злемент Фала         Проект           0         Компиляция Demo_Der была успешно выполнена ранее.         0         0         Построение Demo_Der была успешно выполнена ранее.         0         Виполнено применение к AttraServer по адресу local.AstraNode конфигурации версии 1.0.0.04         0			
bool     int1     uint1     uint2     uint2     int4		Тест         Злемент Фала         Проект           Компиляция Demo_Dev была успешно выполнена ранее.         Вогороние Demo_Dev была успешно выполнена ранее.         Виполнено применение к AdtraServer по адресу localAttraNode конфигурации версии 1.0.0.d4	Activate Windo		
bool     int1     int2     int4     int4		Tecr         Злемент Фалл         Проект           ©         Комплияция Demo, Der была успешно выполнено ранее.         Построение Demo, Der была успешно выполнено ранее.         Виполнено применение к AttraServer по адресу local AttraNode конфигурации версии 1.00.d4	Activate Windo		





**7.8** Во вкладке Формулы введите скрипт, помещающий в переменную *Result* сумму переменных: Var\_a + Var\_b (пользуйтесь подстановкой значений из выпадающего списка).

**Подсказка**: если в поле формулах нет возможности ввода, то необходимо сменить направление переменной на «Выход» либо «Отсутствует»

Demo_Dev - SePlatform.Development Studio			– Ø ×
Файл Правка Вид Проект Помощь			
● ⊕ • ↑ C 🕆 ⊨ 🖨 🕈 ै	◇ [sol · 프 프 _ ] ◇ ] 알 & A A , ] 또 또 한 한 돈 후 피 ㅠ 프 ㅋ 후 프 ㅋ 파 ( ㅋ 후 퍼 ( ㅋ 주 ) + ㅋ ) = = = = = = = = = = = = = = = = = =		Systême platform
🗔 Обозреватель решений 🔹 🖣 🗙	Calculator" # X Demo_Dev 12	😭 Свойства	<b>▼</b> # ×
Поиск 🔎 –	Demo Dev.Domain.EDU-ASTRA.AstraServer.Astra Application.SandBox.Calculator Логический объект 🕬 Demo Dev.omx • Result •	O Result Параметр	•
4 💰 Demo_Dev		Общие	
4 🔃 Demo_Dev		Уровень доступа	публичный
🔺 🦢 Файлы	A. C. Var,a	Область доступа	глобальная
💽 Demo_Dev.omx		Направление	OLOGICIEVEL
Определения     Определения	⇒u var_o	Tue	oleytetojet
✓ as Domain		ТИП	uint4
EDU-ASTRA		Имя	Result
- PLC - Server - System		@ Атрибуты • Result Параметр	• # ×
		Атрибут	Значение
🖓 Панель элементов 👻 🖣 🗙	100% * WH	Описание	Result
Поиск О -	Диаграмма Таблица		
Событие	<i>f<sub>e</sub></i> Формулы → ӊ х		
4 О Параметр	Result		
1 bool	1 Var a + Var b		
(1) int1			
1 uint1			
2 int2			
12 uint2			
10 int4			
U) uint4	Kyohan Oopmi/hai		
Применение к AstraServer по appecy local ArtraNode и	percent and a second 10.04 Beero: 3 as her suseseen: 1		
	beer of a management of a		

Около переменной **Result** появился знак функции <sup>С Result</sup>, который говорит о том, что данная переменная содержит в себе какую-то формулу. На нее можно навести курсор мыши, чтобы узнать подробности о функции.

**Подсказка:** После введения формулы, переключайтесь на журнал формулы, чтобы у вас была возможность видеть информацию об ошибках

- 7.9 Постройте решение, перейдите к развёртыванию и примените конфигурацию к линуксовому серверу.
- **7.10** Перейдите в *OpcExplorer*, дождитесь переподключения Astra\_UA. Проверьте работу калькулятора: введите значения для *переменных A и B*.

Q. Инспектор 1 - SePlatform.Service - OP	CExplorer				× .			_	×
Файл Вид Проект Помощь									
🔾 💭 🥀 ▾ 😻 10.0.0.50	•	Astra_UA		• 🖦 🍇 •	+ - 7 )	• • •			
🚼 Обозреватель	• # ×	10.0.0.50	SandBox.Calc	ulator.Var_b 🛍 🛛	Инспектор 1 🕂 🗙				Ŧ
Поиск	,o •	Тип	Сигнал	Значение	Качество	Время	Источник		
▲ Ө Astra_UA на 10.0.0.50		1 Sand	Box.Calculator.Var_a	2	xopowee: 192 - Good	01.12.2022 12:04:27	Astra UA		
E Server		Sand	Box.Calculator.Var_b	3	хорошее: 192 - Good	01.12.2022 12:04:40	Astra UA		
IntiveTypes		10 Sand	Box.Calculator.Result	5	xopowee: 192 - Good	01.12.2022 12:04:40	Astra UA		
<ul> <li>         SandBox         SandBox          </li> <li>             Q             2 Calculator         </li> </ul> <li>             Q             Parameter         <ul> <li>             Q             777001         </li> </ul> </li> <li>             Service         <ul> <li>             777001         </li> </ul> </li>									





**Важно!** Начальные значения, которые указаны в атрибутах параметров не являются триггерами для запуска вычислений. Вычисления запускаются во время выполнения.

Следующий способ выполнения операций в сервере – производить вычисления при помощи обработчика событий.

**7.11** В панели элементов разверните вкладку Вычисления и перетяните на рабочее поле элемент *Обработчик события*, (Handler)



**7.12** Добавьте в объект *Calculator* ещё один *параметр* типа Uint4, назовите его Result\_from\_Handler (результат из обработчика).







**7.13** Выделите *Handler*, перейдите во вкладку Формулы и запишите скрипт, который в переменную *Result\_from\_Handler* помещает сумму *переменных а и b* (Как на рисунке ниже). Обратите внимание, что в конце необходимо указать «;».

**Подсказка**: если дважды кликнуть по Handler появится диалоговое окно, в котором также можно писать скрипты.

Calculator* + X Demo_Dev +	-
Demo Dev.Domain.EDU-ASTRA.AstraServer.Astra Application.SandBox. Calculator Jor	ический объект 💷 Demo_Dev.omx 🔻 Handler 🔻
→ 😗 Var_a → 😗 Var_b	∫x Result → 😭 Result_from_Handler → (
100 % ~ 埼旺	
Диаграмма Таблица	
<i>ј<sub>ж</sub></i> Формулы	<b>▼</b> ∓ ×
🕼 Handler	
1 Result_from_Handler = Var_a + Var_b;	

#### 7.14 Вернитесь к журналу, обновите страницу (F5) для отрисовки связей.

alculator* 🕫 🗙	Demo_Dev 12	Handler 2			
Demo Dev.D	omain.EDU-ASTR	RA.AstraServer.As	tra Application.SandBox. Calculator Логический объект 📫	Demo_Dev.omx	<ul> <li>Handler</li> </ul>
			00		
→) 🕜 Var_a			^ <sup>≉</sup> ⊠ <u>Handler</u> ≕	f.c.	Result 🏵 🧐
> War b			Var_b Result_from_Handler	Resul	t_from_Handler 🎯
				1	
100 % Y 15H <			¢		

**7.15** Сам по себе обработчик не работает, его нужно активировать. Запускаться он будет при изменении параметров *Var\_a* и *Var\_b*.

Чтобы реализовать его работу наведите мышкой на **переменную а**, дождитесь возникновения окружностей справа и слева прямоугольника, наведите мышкой на правую окружность – курсор превратится в прицел, захватите эту окружность и тяните связь к левому верхнему углу **обработчика** (на нём тоже появится окружность).

Calculator" + X Demo_Dev 12 Handler 12	Ŧ	🔐 Свойства	<b>~</b> ↓ ×
🤗 <u>Demo Dev.Domain.EDU-ASTRA.AstraServer.Astra Application.SandBox, Calculator</u> Логический объект 📫 Demo_Dev.omx 🔹 Var_a	•	<ul> <li>Триггер</li> <li>Общие</li> </ul>	
A Contraction of Partic A		Источник события	Var_a
Var_b Result from Handler		Событие	Обновление сигнала
DØ Var_b □ Var_a Result_from Handler ↔			
	~		
100% * 税任 <	Ť		
Диаграмма Таблица			

Данные связи называются *триггерами* обработчика. При нажатии на оранжевую стрелку появится панель свойств данного триггера и в свойстве «Событие» можно выбрать, при каком действии, произошедшем с сигналом, будет срабатывать *Обработчик события*.

Важно! Нельзя «триггерить» обработчик от его выходных параметров, сервер такое вычисление не





произведёт.

**7.16** Постройте решение, перейдите к развёртыванию и примените конфигурацию к линуксовому серверу.

**7.17** Перейдите в *OpcExplorer*, дождитесь переподключения Astra\_UA и добавьте в инспектор недостающий сигнал *Result\_from\_Handler*. Проверьте работу калькулятора: введите значения для переменных а и b.

Файл Вид Проект Помощь									
🥥 💭 🕂 🗸 🚺 10.0.0.50	-	🕑 Astra_UA	-	n 🙀 -	+ - 7 🕟	0 🔍 🥒 🖉			
🕂 Обозреватель	• ‡ ×	10.0.0.50 ±	SandBox.Calculate	r.Var_b ± И	нспектор 1 🛥 🗙				Ŧ
Поиск	<u>ب</u> م	Тип	Сигнал	Значение	Качество	Время	Источник		
и 🖉 Astra_UA на 10.0.0.50		SandBox.0	Calculator.Var_a	22	хорошее: 192 - Good	01.12.2022 12:47:27	Astra UA		
E Server		M SandBox.	Calculator.Var_b	23	хорошее: 192 - Good	01.12.2022 12:45:35	Astra_UA		
NativeTypes		3 SandBox.	Calculator.Result	45	хорошее: 192 - Good	01.12.2022 12:47:27	Astra UA		
<ul> <li>SandBox</li> </ul>		3 SandBox.0	Calculator.Result_from	n 45	хорошее: 192 - Good	01.12.2022 12:47:27	Astra UA		
4 🕑 Calculator									
Result									
Result_from_Handler									
Var_a									
Var_b									
5 777001									
S 777030									
\$ 777021									
D marameter									
§ 777001									
Service									

Обратите внимание, что если ввести данные в Переменную b, значение **Result\_from\_Handler** не изменится, т.к. триггером является изменение значения переменной а. Если ввести значение для Var\_a, то **Result\_from\_Handler** посчитает сумму.

**Подсказка:** после повторной загрузки проекта в OPC Explorer необходимо отключаться и подключаться вновь, чтобы отобразились актуальные данные

<ul> <li>Подключиться к и</li> <li>Выбор ко</li> </ul>	сполняющему компоненту	>
Выбор компонент	Выберите исполняющие компоненты для по	одключения
Проверка соединения Параметры	Фильтр по имени	<u>,</u>





Internal

62

# 8. Работа с компонент Systeme Platform платформы – SePlatform. Historian

*SePlatform.Historian* – компонент Systeme Platform платформы для сбора и сохранения информации о технологическом процессе.

Возможности SePlatform.Historian:

- ✓ Сбор и хранение оперативных значений параметров технологического процесса;
- 🗸 Сбор и хранение истории событий и тревог технологического процесса;
- Предоставление исторических данных клиентам.

Перед началом работы, необходимо установить компонент *SePlatform*. *Historian* на машину с OC Linux.



8.1 Перейдите в BM Linux, скопируйте дистрибутив из сетевой папки на вашу BM







8.2 Вставьте файл в папку с дистрибутивами

Адрес: 🦰 Компьютер > Домашняя (0:63:0x0:0x0!:) > distr

8.3 Откройте терминал Fly и переместитесь в папку distr



8.4 Удостоверьтесь, что Вы находитесь в директории

Подсказка: С помощью команды Is в Linux можно посмотреть какие файлы находятся в папке

📷 distr : bash — Терминал Fly
Файл Правка Настройка Справка
🕰 🚨 📋 🔝 🗸 🍼
se@astrapc:~\$ cd /home/se/distr se@astrapc:~/distr\$ ls access.txt
ARM_AccessPoint.txt
HKM_WorkStation.txt
ldan.txt
seplatform.accesspoint-1.0.x+fix29135.b14.r87226.deb
seplatform.historian-1.0.11+b2.r87074.deb
seplatform.hmi.alarms-ru_RU-1.1.0+b1.r5878.x86_64.deb
SePlatform.HMI.charts-1.0.0+b6.r85537-x64.deb
seplatform.hmi.commonlib-ru_RU-1.0.3+b1.r5865.x86_64.deb
Seplatform.nml.desktop-ru_ku-1.0.l+b/l.r0/239.x86_64.deb
seplatform hmi security 1.0.0 pig.1 00007-x04.000
seplatform.hmi.setpoints-ru RU-1.0.1+b3.r5772.x86 64.deb
seplatform.hmi.tables-ru_RU-1.2.0+b6.r85536.x86_64.deb
seplatform.hmi.trends-ru_RU-1.0.x+b78.r5463.x86_64.deb
SePlatform.Security-x64-1.0.0+b11.r86668.deb
seplatform.server-1.0.x+fix29135.b14.r87226.deb
sewastrapc:~/distr\$

**8.5** Для установки модуля SePlatform.Historian введите команду: sudo dpkg —i SePlatform.historian\*\*\*\*\*\*\*.deb.







#### 8.6 Для конфигурирования SePlatform. Historian перейдите в папку

/opt/SePlatform/SePlatform.Historian и откройте файл SePlatform.Historian.Server.xml при помощи Редактора Kate.



На машинах с OC Windows *SePlatform.Historian* может работать как через DCOM, так и по TCP. Так как данный проекты кроссплатформенный, нужно отключать DCOM.

**8.7** Для отключения DCOM в строку с тэгом SePlatform.Historian.Server необходимо дописать EnableDCOM="0", тем самым отключив поддержку DCOM.



**8.8** Спуститесь ниже к тэгу **<Base>**. Перед закрытием данного тэга необходимо раскомментировать строчку для определения базы данных. Удалите комментарий и в атрибуте **Alias** введите название для данной базы – «History».







#### **8.9** Сохраните файл **SePlatform.Historian.Server.xml** и закройте.

Система может запросить права администратора. Для этого укажите пароль вашей BM Linux – 12345678

🛒 Требуе	ется аутен	тификаци	я — Аген	т PolicyK	it1 от KDE	-	. 🗆	×
**]	Для сох права а	ранения дминис	і этого / тратор	докуме а	нта необ	бходим	ИЫ	
	Приложе дополнит аутентифі	ние пытаето ельных прі ікация.	ся выполн ивилегий.	нить дейс Для этог	твие, котор о требуется	ое требу	ет	
	Пароль:	•••••						
🕕 Подро	обности >>	]			Да	От	мена	

#### 8.10 Выйдите в Терминал и введите команды для перезапуска

службы

sudo systemctl restart seplatform-historian.server.service



и отображения статуса.

sudo systemctl status seplatform-historian.server.service

💽 se : sudo — Терминал Fly
Файл Правка Настройка Справка
se⊎astrapc:∞⊅ sudo systemcti status sepiatrorm.nistorian.server.service
<ul> <li>Septerforministorian set verses verses de la cromanistorian de verses and de la contracta de la contra Contracta de la contracta de la contracta</li></ul>
Active: active (running) since Thu 2022-112-01 15:40:56 MSK: 4 days ago
Main PID: 8461 (seplatform.hist)
Tasks: 15 (limit: 4915)
CGroup: /system.slice/seplatform.historian.server.service
└─8461 /opt/SePlatform/SePlatform.Historian/seplatform.historian.server &
new RE 19,99,36 patrons conlatform biotorias conver[0461], [EELB blob data]
gek 80 12.23.30 dstrapt sepiationministorianiserver[8401], toito uto uto uto ataj
ger 06 12:24:36 astrapt senlatform historian server[8461]: [6518 blob data]
дек Ø6 12:24:36 astrapc SePlatform.Historian.Server[8461]: [616В blob data]
дек 06 12:25:36 astrapc seplatform.historian.server[8461]: [651B blob data]
дек 06 12:25:36 astrapc SePlatform.Historian.Server[8461]: [616B blob data]
дек 06 12:26:36 astrapc seplatform.historian.server[8461]: [651B blob data]
gek 06 12:26:36 astrapc SePlatform.Historian.Server(8461): [6168 blob data]
gek 06 12:27:36 astrapc seplatform.historian.server184611; 16518 blob dataj
gek 06 12:2/36 astrapc seriattorm.Historian.server18461]: [6166 6100 6ata]
THES T-TO/TO (END)





За сохранение истории отвечает атрибут истории. Добавим нескольким параметрам возможность сохранять историю и просмотрим оперативные и исторические графики с помощью *SePlatform.Trends*. Вернитесь в Windows → *DevStudio*, и перейдите в машину Linux. В панели элементов выберите и перетащите на рабочую область блок SePlatform.Historian

Теперь необходимо настроить проект автоматизации для того, чтобы описать, где находится SePlatform.Historian.

#### 8.11

	🗇 local 🔹 🗄 🚆 🕺 🚳 👷 🥦 🐿 🛍 ⊨ 羊 丰	<u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u></u>	臣 徂 周 ◆ 。		Système platform
🛛 Обозреватель решений 🛛 👻 🗘 🗙	Calculator* Demo_Dev ta Handler ta EDU-ASTRA* 4 X		Ψ	😭 Свойства	- q >
Тоиск 🖉 -	I Demo_Dev.Domain, EDU-ASTRA Ysen SePlatform.Domain => Demo_Dev.omx	Historian	- 😥	间 Database База данны	x SePlatform Historian
🖌 👶 Demo_Dev 🔗				Общие	
a Demo_Dev				Псевдоним	default
и 🕜 Файлы 💿 Пото Откорти				Тип использования	Для значений и событий
<ul> <li>Дето_Devicing</li> <li>Определения</li> <li>Определения</li> </ul>	Ge AstraServer #2 12			Только именованные множества	Нет
# Domain	Historian H			Имя	Database
4 Q. AstraServer	A assence				9
AeServer		netAdapter 💷 😽			
A Ab barry the Contract	Un caserver	50			
Astra Application	1000	30			
Astra_Application	TcpServer				
	TcpServer				
<ul> <li>✓ ✓ Astra_Application</li> <li>✓ ✓ SandBox</li> <li>✓ ✓ Calculator</li> <li>○ Parameter</li> </ul>					
<ul> <li>✓ ✓ Astra_Application</li> <li>✓ SandBox</li> <li>✓ Calculator</li> <li>○ Parameter</li> <li>☑ HistoryModule</li> </ul>					
<ul> <li>we_kstra_uppication</li> <li>✓ QandBox</li> <li>✓ Calculator</li> <li>○ Parameter</li> <li>♥ HistoryModule</li> <li>♥ Teseverer</li> </ul>				@ Атрибуты	• a >
<ul> <li>▲ W Astra - ppiration</li> <li>▲ Ø SandBox</li> <li>F Ø Calculator</li> <li>○ Parameter</li> <li>♥ HistoryModule</li> <li>֎ TcpServer</li> <li>₩ UsServer</li> <li>₩ UsServer</li> </ul>				@ Атрибуты 付 Database База данны	د به ع x SePlatform.Historian
<ul> <li>▲ W. Astra - ppiration</li> <li>▲ Ø SandBox</li> <li>▶ Ø Calculator</li> <li>⇒ Parameter</li> <li>ঊ HistoryModule</li> <li>ঊ TopServer</li> <li>֎ UaServer</li> <li>W EtherretKäpter</li> </ul>				@ Атрибуты <b>]</b> Database База данны Атрибут	т ф. з x SePlatform.Historian Значение
				@ Атрибуты Database База данны Атрибут	¥ д з x SePlatform.Historian Значение
We kata "Application     Sandox     Galuator     Gal	TopServer Jun - Wei Johnson Telonus			@ Атрибуты Database База данны Атрибут	⇒ q. з x SePlatform.Historian Значение
	Trademan Acarponus Tafonus Carponus Tafonus		- 4 ×	<ul> <li>Атрибуты</li> <li>Датаbаse База данны Атрибут</li> </ul>	≂д) x SePlatform Historian Значение
A (L, Star, Application     Sandor     A (L, Star, Application     A (L, Star, Ap		ñe	т Ф х скложурналу р -	@ Атрибуты Database База данны Атрибут	₩ ĝi j x SePlatform Historian Bravenine
A W, Sang Application     Sandbox     A Sandbox     Calculator     Parameter     WithoryModule     SepletormServer     SepletormServer     SepletormServer     SepletormServer	Image: Second State	ракования про Злемент Файл Про	τ <b>φ</b> (x το χριαλή β τ φτ   Λ	Фарибуты Database База данны Атрибут	← g э sePlatform Historian Значение
Augusta Application     Sendor     Calculator     Calculator		ром Злемент Файл Про Остор	cr. ng xypesay D = ee D = Dev	<ul> <li>Атрибуты</li> <li>Датаваче База данны: Атрибут</li> </ul>	т д ) x SePlatform Historian Злачение
A we star	Image: Second	Demo Demo Demo	r 0 x cr no xypetay → P + en _Dev Dev	<ul> <li>Атрибуты</li> <li>Датавазе База данны Атрибут</li> </ul>	¥ 8 2 x SePlatform Historian 3начение
Augusta Application     Augusta     A		лемент Файл Про Элемент Файл Про Demo Demo	• 8 х соло хурналу — Д = еет — Л "Dev "Dev "Dev	<ul> <li>Atpudytu</li> <li>Database Sasa garrue Aspudyt</li> </ul>	v 9 ) SePlatform-Historian Злачение
A W, KARS, Application     Sendor     Sendor     Sendor     Sendor     Sendor     Sendor     Sendor     Sendor     Sendorminitator     Sendor	TopServer      T	nov Злемент Файл Про Demo Demo Demo Demo	cc no xypesty cc no xypesty Dev Dev Dev Dev Dev	<ul> <li>Атрибуты</li> <li>Датаваяе База данние Агрибут</li> </ul>	v (k 2) SePlatform Historian 3inakenne
Augusta Application     Sandor     Calculator     Calculator		Jaressent Galls. Jaressent Galls. Demo Domair Den Demo Domair Den Demo Domair Den Demo	• Ф Х сс по журналу — До = "Дек "Дек "Дек "Дек "Дек	<ul> <li>Атрибуты</li> <li>Олавланска данны</li> <li>Агрибут</li> </ul>	← () s SePlatform Historian 3 havenue

#### 8.12 В поле «Псевдоним» укажите «History»

😭 Свойства	<b>-</b> ↓ ×						
间 Database База данных SePlatform.Historian							
Общие							
Псевдоним	History						
Тип использования	Для значений и событий						
Только именованные множества	Нет						
Имя	Database						





8.13 Укажите базу данных, с которой будет работать модуль истории. Перейдите Astra Server
 → в окне модуля нажмите «+» и выберите из выпадающего списка базу данных, с которой будет работать модуль. После чего нажмите «Да»



**8.14** Нажмите мышью на базу данных и в окне свойств укажите Каталог очереди данных: \tmp\QuQue



Важно! Данный каталог выбирается только для нужд обучения. При разработке реального проекта необходимо выбрать другой каталог.





Дa

Отмена

**8.15** Перейдите Sandbox → Calculator и настройте сохранение данных для переменной Result\_from\_Handler.



Для этого в меню файл выберите → Вид → История





8.16 В появившемся окне поставьте галочку «Сохранять в историю»

История	I.		-	ųΧ
→ Result_fr	rom_Handl	er Парамет	p	+
Сохран	ять истори	ю		
Зона неч значени	нувствител ю	ьности по	7.51	\$
Зона неч времени	чувствител 1, мс	ьности по		•
Сохр	анять с ме	ткой време	ни сервера	
Атрибуты	История			

- 8.17 Постройте решение и залейте конфигурацию
- **8.18** Перейдите в OPC Explorer, кликните дважды ЛКМ по параметру значение и отметите галочкой «Генерировать значения» Выберите из выпадающего списка ниже генератор «пила».

🔾 🔍 🔨 🖌 🙀 10.0.050	- 0	Astra_UA	- 4	b 🔩 -	+ - 7		• • • • • •		
Обозреватель	• # × 10	.0.0.50 ±	SandBox.Calculator.Va	er_b ≭⊐ Ин	спектор 1 📲 🗙				
	Tu Tu	n	Сигнал	Значение	Качести	0	Время	Источник	
Astra_UA Ha 10.0.0.50		SandBox.	Calculator.Var a	0	1/10x0e: 32 - Waiti	ng For Initial	06.12.2022 13:19:13	Astra_UA	 
<ul> <li>Server</li> <li>NativeTypes</li> </ul>		SandBox.	Calculator.Result	0	noxoe: 32 - Waiti	ng For Initial	06.12.2022 13:19:13	Astra UA	
▲	0	SandBox.	Calculator.Result from	0	noxoe: 32 - Waiti	ng For Initial	06.12.2022 13:19:13	Astra UA	
▷       13 Result         ▷       13 Var_b         ▷       13 Var_b         ○       177001         ○       177002         ○       177021         ▷       177021         ▷       10 Parameter         ○       777001         ○       777021         ▷       10 Parameter         ○       777001         ○       777001			Изменить значен ns=1;s=SandBox.Ca Value A 0 Генерировать зна пила Функция for (byte i = 0; i < 2 yield return i; for (byte i = 255; i ) yield return i;	ие   culator.Var_= ачения 55; i += 5) 0; i -= 5)	v ]				





## 8.19 Проделайте то же самое для переменной Var\_b и укажите другое имя «Пила2»

				· •					
Почек	• # ^	10.0.0.50 12	SandBox.Calculator.Va	r_b 12					•
	-	Тип	Сигнал	Значения	Качество	Время	Источник		
Astra_DA Ha 10.0.0.00		SandBox	Calculator.Var.b	0	xopowee: 192 - Good	06 12 2022 13:25:04	Astra UA		_
Server		SandBox	Calculator.Result	125	неопределенное: 88 - Sub-Nc	06.12.2022 13:25:04	Astra UA	 	
▲		A SandBox	Calculator.Result from	125	неопределенное: 88 - Sub-Nc	06.12.2022 13:25:04	Astra UA		
▶ ① Result         ▶ ① Result_from_Handler         ▶ ① Var_a         ▶ ① Var_b         ⑤ 777001         ⑤ 777020         ⑥ 777020         ⑥ 777021         ▷ ⑦ Parameter         ⑤ 777001         ⑤ 777001         ◎ 777021         ▷ ⑦ Parameter         ⑤ 777001         ▷ Service         ☑ 777001			✓ Изменить значени ns=1:s=SandBox.Ca Value B 0 ✓ Генерировать зна пила2 Функция for (byte i = 0; i < 25; yield return i; yield return i;	не Iculator.Va 55; i += 5) 0; i -= 5)	r_b	<			li A

# 8.20 Теперь значения генерируются, а история сохраняется в Result\_from\_Handler

Q, Инспектор 1 - SePlatform.Service - OP	CExplorer							-	×
Файл Вид Проект Помощь									
🔾 💭 🐧 - 🙀 10.0.050	•	Astra_UA	• 🕀	- 🙀 -	4 - 7	) 🖲 🖲 🖉	20		
1 Обозреватель	• a ×	10.0.0.50 12	SandBox.Calculator.Var	b th	Инспектор 1 🖷 🗙				Ŧ
Поиск	<i>.</i>	Тип	Сигнал	Значени	качество	Время	Источник		
▲ 🚱 Astra_UA на 10.0.0.50		SandBox.Ca	lculator.Var_a	40	xopowee: 192 - Good	06.12.2022 14:29:4	5 Astra UA		
D Server		SandBox.Ca	lculator.Var_b	40	хорошее: 192 - Good	06.12.2022 14:29:4	5 Astra UA		
NativeTypes		SandBox.Ca	lculator.Result	80	xopowee: 192 - Good	06.12.2022 14:29:4	5 Astra UA		
4 🍠 SandBox		G SandBox.Ca	lculator.Result from	85	xopowee: 192 - Good	06.12.2022 14:29:4	5 Astra UA		
4 📀 Calculator		-							
P 03 Result									





8.21 Для просмотра исторических данных кликните ЛВК по параметру
 *Result\_from\_Handler*. В открывшемся окне есть возможность указать интервал времени, за которое необходимо просмотреть историю – кнопка Загрузить историю значений за выбранный интервал времени. Выберите «За последние 10 минут».



Также можно открыть график сигнала. Для этого в Обозревателе кликните ПКМ по нужному параметру и выберите. Показать график сигнала.






# 9. Работа с событиями

В проекте есть возможность настроить генерацию событий и тревог при наступлении определённых условий. Например, при превышении уровня можно уведомлять об этом пользователя не только цветовой индикацией на мнемосхеме, но и с помощью событий в журнале. В том числе, со звуковым сопровождением. Журналы событий затем можно анализировать при разборе инцидентов. Сначала, необходимо настроить генерацию событий в сервере.

# Настройка генерации событий в SePlatform.DevStudio

Для генерации событий необходимо, чтобы в составе сервера был модуль **OPC AE Server**, который будет генерировать события и передавать их клиентским приложениям. В конфигурации по умолчанию уже присутствует данный модуль.

 ASUA	
	UaServer
	TroSenver

Теперь необходимо сконфигурировать сигналы, которые будут генерировать события. Для примера будем использовать подусловие отклонении значения сигнала суммы двух сигналов.

9.1 Перейдите в DevStudio, нажмите Вид → События для отображения соответствующей вкладки на панели атрибутов (в правой нижней части экрана). Перейдите в логический объект Calculator, выделите параметр Result, откройте вкладку События и поставьте галочку около Генерировать события.

Iculator 12 ×		Свойства
emo. Domain. astra. Astra Server. Astra A	oplication.SandBox, Calculator Логический объект 💷 Demo.omx	🦪 События
-		O Result Параметр
∫	Result	🗌 Агрегировать события
-		🗹 Генерировать события
O 🤬 Var_A		О Перечисление
	A Contraction of the second se	andler О По уровню
○ 🚳 Var_B	Vor A	О Отклонение
		Атрибуты События Исто

**9.2** Выберите событие По уровню, укажите значение и сообщение, которое будет отображаться при возникновении данного события.

🦪 Событ	тия						<b>-</b> ₫ ×
O Resul	tПа	раметр					
Arper	гира	вать соб	ытия				
🖌 Генер	рира	овать соб	ытия				
ОПере	ечи	ление					
• По у	ров	ню					
О Откл	поне	ение					
1		Подусл	овие	Значение	Сообщение	Важность	Деактивирующее
	~	Выше у	ровня т	150	. Высокий результат	1	
6	~	Норма	*		. Нормальный результат	1	
6	~	Ниже у	ровня 👻	100	. Низкий результат	1	
Атрибуть	ы	События	История				1

**9.3** Поднимитесь на уровень выше (в логический объект SandBox), задайте атрибут Описание для логического объекта *Calculator* – Калькулятор





Internal

73

- **9.4** Поднимитесь ещё на уровень выше (внутрь AstraApplication), задайте атрибут Описание для логического объекта *SandBox* Песочница.
- 9.5 Постройте решение, перейдите к развёртыванию и примените конфигурацию к линуксовому серверу.

# Просмотр событий

В OpcExplorer есть несколько способов наблюдать события.

1. Перейдите в *OpcExplorer*, дождитесь переподключения сервера.

2. Так как события настроены внутри логического объекта *Calculator*, нажмите на него в Обозревателе. События по уровню генерируются тогда, когда порог пересекается, а не каждый раз, когда значение выше порога.



Есть ещё один способ просмотра событий в *OpcExplorer*. Кликните на вкладку Проект → Добавить события. Перетяните в открывшееся окно объект *Calculator* из обозревателя.

# 8. Модификация проекта SePlatform.DevStudio (часть 2)

Далее углубимся в возможности *SePlatform.DevStudio*. Начнём с обмена данными между объектами. Изучим функционал объектных ссылок. Для корректной работы демонстрационного проекта отключите генераторы с *Var\_A* и *Var\_B OpcExplorer*.

# Передача данных между объектами (ссылки)

Для начала в сервер нужно добавить ещё объект, с которым будет происходить обмен данными.

1. Перейдите в *SandBox* и из панели элементов перетяните *Логический объект*, назовите его Calc\_from\_Link – калькулятор, который будет считать данные по ссылке.

2. Перейдите внутрь *Calc\_from\_Link*, в панели элементов раскройте папку Ссылки и перетяните элемент *Объектная ссылка* на рабочую область.





🗸 Demo - DevStudio		
Файл Правка Вид Проект Справ	ка	
● ● - t C 🕆 🗃 🖶 🗗	ి ి	локальная 🔹 🞬 🚆 📢 🍦 😰 🦂 😜 🖕
🗔 Обозреватель решений	- ų ×	Calc_from_Link* + ×
Поиск	,o –	Demo.Domain.astra.AstraServer.AstraApplicati
🖓 Панель элементов	<b>-</b> ∓ ×	
Поиск	– <b>م</b>	
F { } Om.PLC		
A Server		
Банисления		
🔺 🗁 Ссылки		
🔊 Ссылка		
Объектная ссылка		
🛞 Инициализатор ссылки		
Логический объект		
🖻 🍠 Событие		
Параметр		
🧇 Сокет		

3. В свойстве Объект этой ссылки выберите <u>Calculator</u>, находящийся в **AstraServer**.

Calc_from_Link* += ×	🚰 Свойства		*	<b>ņ</b> :
Demo.Domain.astra.AstraServer.AstraApplication.SandBox.Calc f	from Link GubjectReference Объектная сс	ылка		
	Объект			[
Ссылка на	Исполняющий компонент	🎎 Укажите путь к элементу	×	
<del>й</del> ф	Только чтение	Поиск	<i>p</i> -	
	Аспектная	🔺 📑 astra		
	Имя	AstraServer	[	
		SandBox     SandBox     Calc_from_Link     Calculator     Iserring00		
		Да	Отмена	

Но у этого объекта не указаны те объекты, с которыми он может взаимодействовать. То есть не указаны входные и выходные параметры. Для описания коммуникации используется такое свойство, как Направление у параметров. Если у параметра нет направления, значит, он не может участвовать в коммуникациях. Чтобы связи были доступны, нам необходимо у параметра поменять свойство Направление.

Перейдите в *Calculator*, выделите *переменную А* и в свойстве Направление укажите <u>Вход</u>. Сделайте то же самое для *переменной B*, а для параметра *Result\_from\_Handler* укажите направление <u>Выход</u>. Обновите страницу для отрисовки (F5).

J	Calculator* + ×		:
4	🕨 <u>Demo.Domain.astra.AstraServer.AstraApplication.SandBox. Calculator</u> Логический объект 🛒 🛛 Der	emo.omx	• •
	✓ ✓ ✓ Kesult       ✓ ✓ Kesult       ✓ ✓ Handler Summa =       ✓ ✓ Lander Summa =       ✓ ✓ ✓ Lander Summa =       ✓ ✓ Lander Summa =       ✓ ✓ Lander Summa =       ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓		Result_from_Handler 🎯 🕼

Обратите внимание, теперь параметры рисуются в трех разных зонах: входная зона – слева, в середине элементы, которые не участвуют в направлении, справа – выходная зона.





4. Вернитесь в *Calc\_from\_Link*, щёлкните ПКМ по ссылке и выберите Экспонировать входы и выходы. Студия сама создаст параметры с теми же именами и типами и нарисует соответствующие связи.

Calculator* +> X	
🤌 Demo.Domain.astra.AstraServer.AstraApplication.SandBox, Calculator Логический объект 🛒	Demo.omx 🔹
Image: Second Secon	Result_from_Handler 🤿 🕲

Есть ещё один способ создать ссылку на объект. Вместо того, чтобы перетягивать элемент **Объектная ссылка** из панели элементов, Вы можете вытянуть объект, на который хотите сослаться из Обозревателя решения.

Важный момент! Из обозревателя решений перетягиваются ССЫЛКИ на объекты, а из панели элементов – ЭКЗЕМПЛЯРЫ.

5. Постройте решение, перейдите к развёртыванию и примените конфигурацию к линуксовому серверу.

Теперь в проекте есть описание связи между объектами и параметрами объектов. Таким же образом можно обмениваться данными между объектами не только в рамках одного сервера, но и из внешних исполняющих компонентов. Например, с контроллерами, либо с другими серверами вводавывода.

#### Подключение к исполняющему компоненту через DeveloperStudio

*DevStudio* может как *OpcExplorer* напрямую подключаться к серверу. Для этого существует кнопка **Подключиться к исполняющему компоненту**. Укажите, к какому серверу подключаться (AstraServer) → нажмите Далее → дождитесь сообщения «Готов к отладке» → Далее → Поставьте галочку около Изменение значений → Готово. И *DeveloperStudio* напрямую подключится к серверу и будет получать значения.







Cale	c_from_Link +¤ ×	<							-
🤌 <u>De</u>	mo.Domain.ast	ra.AstraServer.AstraA	pplication.SandBox. Cal	lc from	Link Логический объект 🛋	Demo.omx	•	Result_from_Hand <b>*</b>	astra.AstraServer.4 🔻
+	≫us Var_A ≫us Var_B (	45	SubjectReference       Ссылка на Calculator       Э щ Var_A       45       Э щ Var_B	106	Result_from_Handler (>) (6)			106 Result_from_H	landler →國

При изменении входных параметров, их значения отправляются в *Calculator*, там происходят вычисления и результат передаётся в выходной *Result\_from\_Handler* внутри *Calc\_from\_Link*.

Далее в проекте будет описание связи с внешним исполняющим компонентом – сервером вводавывода, имитирующего логику ПЛК.

# Добавление внешних исполняющих компонентов и реализация передачи данных между машинами

Для того, чтобы получить данные с сервера, эмулирующего работу ПЛК, необходимо на отдельной машине развернуть конфигурацию сервера. Для этого необходимо установить компоненты на вторую машину, и развернуть проект Imitator из SePlatform.DevStudio на SePlatform.Server. После развёртывания можно приступать к дальнейшему прохождению материала.



Для того, чтобы описать информационную модель обмена данными между компонентами, в проекте необходимо описать компоненты, с которыми будет происходить обмен данными, и интерфейсы обмена. В проекте мы будем описывать сервер, эмулирующий ПЛК, как внешний исполняющий компонент, находящийся на другой машине.

Сервер коммуницирует с устройствами, а устройства относятся к среде исполнения. Среда исполнения находится в *SePlatform.Domain*. Здесь описаны машины, то есть узлы, объединённые в одну сеть. И здесь же можно разместить какие-то внешние устройства, которые отвечают за вычислители.

- 1. Перейдите в **Domain** при помощи Обозревателя решений.
- 2. Перетяните сюда из панели элементов *Компьютер*, назовите его PLC, дайте комментарий.







77

Элемент *Компьютер* отличается от элемента *Узел SePlatform.Domain* тем, что у элемента *Компьютер* нет свойства адрес, которое нужно для описания **SePlatform.Net.Agent** для того, чтобы определить, как идентифицировать конкретный узел в среде развёртывания.

3. Перейдите в *Компьютер PLC*, выделите *Ethernet Adapter* и в свойстве Адрес задайте ему IPадрес машины, на которой развёрнут проект Imitator. Также задайте ему Сеть <u>Ethernet</u>.

PLC* += X	👻 🚰 Свойства	
<u>Demo.Domain. PLC</u> Компьютер	🛚 🐝 💵 EthernetAdapter Адаптер Ethernet	
	🔊 Общие	
PP	Адрес	10.0.2.16
EthernetAdapter	Сеть	Ethernet
	Имя	EthernetAdapter

4. Находясь внутри *PLC* перетяните из панели элементов *Исполняющий компонент*, назовите его Imitator, дайте комментарий.

🎇 Demo - DevStudio	
Файл Правка Вид Проект Справка	
💽 ⊙ - € ≎ 📅 🐸 🛃 🖨 🔿 🖒	локальная 🔹 🛗 🚆 👷 🖓 🌑 🖕 🏪 🐂 🐂 🛤 💷 🗐
🗔 Обозреватель решений 🔍 🔻 🛪 🗡	< PLC* += X
Поиск 🖉 -	📳 Demo.Domain. PLC Компьютер 💷 Demo.omx
Панель элементов	Imitator
Поиск р -	10.0.2.16
Исполняющий компонент	
Резервный исполняющий компонент	歳 Комментарий 🛛 🗙
🥼 Дублирующий исполняющий компоне	» Сервер, имитирующий работу ПЛК.
📖 Адаптер Ethernet	
📄 Папка обмена	
Последовательный порт	
Сервер истории Microsoft SQL Server	Да Отмена
🕖 Сервер истории PostgreSQL	
📖 Система ПОРТАЛ	
Мастер ПК	

5. Для того, чтобы описать всё, что будет происходить на этом сервере, перейдите в *Imitator* и перетяните сюда из панели элементов *Приложение*, назовите его Application\_PLC и задайте комментарий.

🕽 Обозреватель решений	- 4 ×	Imitator* 🕫 🗙			
Поиск	P - Q	Demo.Domain.PLC.	mitator Исполняющий ком	понент 🛒 Demo.	omx
📝 Панель элементов	- 4 ×				
			3 Комм Дерево об (источника	ентарий ъектов и параметров і данных)	ж имитатора ПЛК
🖻 🚞 МЭК 61850				Да	Отмена

Внутри этого *Приложения* необходимо указать объекты и параметры, с которыми будет происходить взаимодействие.





6. В проекте Imitator есть датчик, и от него мы будем получать значения. Для осуществления этого перейдите в *Application\_PLC* и добавьте сюда *Логический объект*, назовите его Sensor\_in\_PLC (датчик со стороны источника).

7. Перейдите внутрь *Sensor\_in\_PLC* и расположите здесь *параметр* типа float. Назовите его Pressure и задайте ему направление <u>Выход</u>.



На этом описание источника с точки зрения объектно-ориентированного программирования завершено. Теперь перейдём к описанию объекта на сервере ввода-вывода (потребитель, AstraServer).

8. Перейдите в *AstraServer* → *SandBox*, перетяните сюда из панели элементов *Логический объект*, назовите его Sensor\_in\_Server (датчик со стороны сервера ввода-вывода).

9. Перейдите внутрь *Sensor\_in\_Server* и из Обозревателя решений перетяните сюда объект *Sensor\_in\_PLC*, создавая тем самым ссылку на объект, размещенный в *PLC*. Заметьте, что свойство Объект уже заполнено.

魏 Demo - DevStudio			– 0 ×
Файл Правка Вид Проект Справка			
● ● • € ♀ 🔁 🖴 🖶 🖨 💍 🌣	локальная 🔹 🖓 🛗 🛫 📢 👷 🖓 🍕 🖕 🧮 🐂 🐂 ⊨ ≐		k? @
🔄 Обозреватель решений 🔷 👻 Д 🗙	Sensor_in_Server* * ×	🚰 Свойства	
Поиск 🔎 -	Demo.Domain.astra.AstraServer.AstraApplication.SandBox, Sensor in Server.AstraApplication.SandBox	G_Sensor_in_PLC Объектная ссылка	•
🔺 🎎 Demo 🔨 🔨		🔿 Общие	
4 🐼 Demo	Carl Sensor in PIC	Объект	Domain.PLC.Imitator.Application_PLC.Sensor_in_PLC
Файлы Файлы	Ссылка на Domain.PLC.Imitator.Application_PLC.Sensor_in_PLC	Исполняющий компонент	
и и Спределения и и Спределения	Pressure 🏵 🚯	Только чтение	Нет
astra		Аспектная	Нет
<ul> <li>AstraServer</li> </ul>		Имя	_Sensor_in_PLC
AeServer		0	
AstraApplication			
💝 HistoryModule			
Stephene Ste			
Un Uaberver			
Historian			
de Ethernet			
Iearning00			
A I PLC	Y		
EthernetAdapter			
4 🤽 Imitator			
Application_PLC			
+ PLC V			

10.Кликните по ссылке *\_Sensor\_in\_PLC* ПКМ → Экспонировать входы и выходы.





Demo.Dor	nain.astra.AstraServer.AstraApplic	cation.SandBox. Se	nsor in Serv	er Логический с	объект 💷 🛛 🛛	emo.c
	G_Sensor_in_PLC		١ آ			
	Ссылка на <u>Domain.PLC.Imitator.App</u>	lication_PLC.Sensor_in_l	PLC		Pressure	()€
		Pressure 🕀	0			
	Переименовать	F2				
ę	Комментарий	Ctrl+*				
	Удалить комментарий	Ctrl+Alt+*				
1	Перейти к определению	F12				
	Изменить тип элемента		•			
×	Удалить	Delete				
	Экспонировать входы и выходы					
	Экспонировать отдельные сигна.	лы				
	Инициализировать ссылку					

На этой объектно-ориентированный подход закончен. Осталось описать транспортный уровень.

11.Перейдите в *Application\_PLC* и из панели элементов перетяните сюда *Kapmy adpecos Modbus*. Кликните на значок папки около надписи Файл и задайте имя файла MB\_Map.

🗔 Обозреватель решений	<b>▼</b> ↓ ×	Sensor_in_Server* Application_PLC* += ×
Поиск	,o -	🖧 Demo.Domain.PLC.Imitator. Application PLC Приложение 💷
🖓 Панель элементов	- 4 ×	
Поиск	0870-5 10870-5 (шлк 11850 ис. А Е IDA	Sensor in PLC Pressure Device Pressure Device Pressure Dev

12.Зайдите внутрь *карты адресов ModbusAddressМар* и укажите параметры, указанные на изображении ниже (данные адреса идентичны адресам из проекта Imitator).

5	Sensor_in_Server* Application_	PLC*	ModbusAddressM	lap* +¤ ×						Ŧ
88	Demo.Domain.PLC.Imitator.Applic	ation PL	C. ModbusAddress	<u>sMap</u> Карта адре	сов Modbus	₽Ĩ				Ī
	Сигнал	Тип	Привязка 🔺	Сегмент	Адрес	Номер бита	Номер записи в файле	Метка времени	Размер строки	
K	•	-	-	-	•	-	•	•	•	
۵	Sensor_in_PLC.Pressure	float	непосредственно	Input Registers	48			Нет		

Сохраните карту и обязательно закройте её (карта обновляется при открытии).

13.Для описания того, кто эту карту адресов будет обслуживать, перейдите в *Imitator* и перетяните из панели элементов *Станцию Modbus TCP*. В свойстве Карта адресов выберите ту карту <u>ModbusAddressMap</u>.





🗔 Обозреватель решений	• 4 × 5	Imitator* + ×	<del>,</del>	🚰 Свойства		
Поиск	0-0	Demo.Domain.PLC. Imitator Исполняющий компонент 📲		🖓 ModbusTcpSlave Станц	я Modbus TCP	
Панель элементов Поиск	• # ×	Application PLC 💷 🌸		Параметры станции Номер станции		
D 🔁 BACnet	<u>^</u>	A Sensor in DIC	🎯 ModbusTcpSlave 🖃 🥥 🍟	Номер ТСР порта		502
Modbus		ModhusAddressMap	Станция Modbus TCP	Кодировка текста		windows-1251
Опросчик Modbus RTU Станция Modbus TCP Станция Modbus TCP Станция Modbus TCP				<ul> <li>Изменение порядка б Байт в слове</li> <li>Слов в сигналах TMF4, TRF</li> </ul>	айт 4, TMF8	Нет —
				Слов в сигналах ТМС, TR4		Нет
				Общие Карта адресов		
🖻 🧰 МЭК 60870-5				Имя	🎎 Укажите путь к эл	лементу Х
M3K 61850					Поиск	<u>ہ</u> م
Приложение	> ~				B PLC     A lmitator     A Applicat	tion PLC

С точки зрения транспортного уровня мы сейчас описали источник. Но серверу с этим источник надо взаимодействовать. Нужно добавить модуль, который будет работать с этой станцией.

14. Перейдите в AstraServer и перетяните сюда из панели элементов Опросчик Modbus TCP.



15.Постройте решение, перейдите к развёртыванию и примените конфигурацию к линуксовому серверу.

Для наблюдения результата перейдите в *SandBox*.

)em	10.Domain.astra.As	traServ	er.Astra	aApplication. SandBox Логичес	кий объект 🗐	Demo.om
	O 😡 Paramet	er				
	→ <sup>1</sup> Var_A → <sup>1</sup> Var_B	10 0	0	Result_from_Handler →	-10 P	in <u>Server</u>
	Calc from I	.ink				
	→ 🕜 Var_A	0	0	Result_from_Handler 🏵 🔢		
	Nor B	0				





# Объектно-ориентированный подход

В SePlatform.DevStudio используется объектная модель, основанная на объектно-ориентированной парадигме, но имеющая отличительные особенности, связанные со спецификой области промышленной автоматизации. Основные понятия, используемые в SePlatform.DevStudio:

#### Объект

Объекты в SePlatform.DevStudio обычно описывают какие-либо реальные автоматизируемые объекты, либо мнимые. Объекты являются ключевой единицей информации при описании решения. Логические объекты ранее были добавлены в демонстрационный проект.

#### Тип

Тип (класс, шаблон) является описанием набора похожих друг на друга объектов, имеющих одинаковую роль в решении, а также общий набор данных и правил их обработки. Использование типов позволяет структурировать большое количество объектов, а также облегчает их изменение и добавление новых объектов. Далее в проекте опишем пример простого логического типа, и работу с его экземплярами.

#### Использование типов

В логическом типе *SandBox* (песочница) есть объект *Calculator*. Давайте попробуем создать логический тип, который будет повторять ту же самую логику, а потом его растиражируем несколько раз, чтобы понять принцип работы типизации.

Логические типы описываются за пределами среды исполнения, то есть за пределами **Domain** – пространство **Определения**.

1. При помощи Обозревателя решений перейдите в пространство Определения.



2. Перетяните в *Определения* из панели элементов *Пространство имён* (по сути это просто папка для хранения объектов). Назовите его Types, задайте комментарий.







3. Перейдите внутрь *Types* и добавить ещё одно *Пространство имён* с именем SandBox\_Type, задайте комментарий.



4. Перейдите внутрь *SandBox\_Type* и из панели элементов добавьте сюда *Логический тип*. Назовите его Calc\_Type и задайте комментарий.



5. Перейдите внутрь *Calc\_Type* и создайте структуру *Calculator*, используя обработчик. Помните про направление у параметров и триггеры для обработчика.





Imitator* SandBox* Calc_Type* + ×		
💊 <u>Demo.Types.SandBox Type. Calc Type</u> Логический тип 🍯	Demo.omx	▼ Handler ▼ 👌
→ ② ② Var_A	Var_A Handler ≡ Var_A Result	Result 🔿 🚺
<i>f</i> ж Формулы		<b>→</b> ‡
🏂 Handler		
1 Result = Var_A + Var_B;		

Логический тип добавлен и описан. Теперь можно приступать к тиражированию экземпляров данного типа.

6. Перейдите в *SandBox* на *AstraServer* и из панели элементов перетяните сюда 3 экземпляра *Calc\_Type*. Назовите их Calc\_1, Calc\_2 и Calc\_3.

🕎 Обозреватель решений	<b>-</b> ↓ ×	Imitator* SandBox* += × Calc_Type*
Поиск	P -	🥐 <u>Demo.Domain.astra.AstraServer.AstraApplication, SandBox Логический объект 🛸</u> Demo.omx 🔹 Calc_3 🔹 astra.AstraSer
🕜 Панель элементов	- # ×	
Онисс           ▲ {} Турез           ▲ {} SandBox_Type           ✔ Calc_Type           ✔ Calc_Type           ↓ Om.PLC           ▷ {} Om.Server           ▲ Burucceния           ☑ Обработчик события           ☑ Обработчик события           ☑ Прасисание           ☑ Собълки           ✔ Логический объект           ☑ Событие           ☑ Событие           ☑ Параметр           ☑ Соскет	Q *	Image: Sensor in Server         20       Pressure         20       Var, A         20       Var, A         20       Var, A         20       Var, A         20       Var, B         20       Var, B         20       Var, A         20       Var, B         20       Var, A         20       Var, B         20       Var, B         20       Var, B         20       Var, B         20       Var, B

7. Постройте решение, перейдите к развёртыванию и примените конфигурацию к линуксовому серверу.

Для наблюдения результата перейдите в *SandBox*.

o.Domain.astra.AstraServe	r.AstraAp	<u>plication. SandBox</u> Логич	еский объект 🛋	Demo.or	nx	• Parameter		<ul> <li>astra.AstraS</li> </ul>
O 😰 Parameter		Sensor in Server 🛋	<ul> <li><u>Calc 1</u> Тип: <u>Турез.Sa</u></li> <li>→ 10 Var_A</li> </ul>	indBox_Type 0 0	≝ <u>Calc Type</u> Result → 🕑	<ul> <li><u>Calc 2</u> Тип: <u>Types.Sa</u></li> <li>→ 14 Var_A</li> </ul>	andBox Type 0 0	L <u>Calc Type</u> Result → 🙆
✓ Calculator           →D 10         Var_A         10           →D 10         Var_B         0	0	Result_from_Handler 🏵 🧐	<ul> <li>Var_B</li> <li>Calc 3 Тип: Types.Sa</li> </ul>	0 IndBox Type	.Calc Type	→ 🕼 Var_B	0	)
✓ <u>Calc from Link</u> ↔ <u>War A</u> 0     ↔ <u>War B</u> 0	0	and the sult_from_Handler → 🕑	→ (A) Var_A → (A) Var_B	0 0	Result → 🚺			

Так работает типизация. Вы один раз описали тип и много раз его переиспользовали. Экземпляры не зависят друг от друга. Логика у них одна и та же, но данные свои.

#### Использование ссылок в типах

Ранее в проекте описывалась передача данных от одного объекта к другому. В случае необходимости обмена данными от одного типового объекта к другому используется другая методика. Для этого в логическом типе можно описать типовую схему передачи данных от одного типа объекта к другому. Для этого в логическом типе используются ссылки и представления типов.





Опишем логические типы со стороны источника (PLC) и сервера ввода-вывода (AstraServer).

1. Перейдите в пространство *Определения* → *Types* → *SandBox\_Type*. Создайте здесь *Логический тип*, назовите Sensor\_in\_PLC\_Type и дайте комментарий.



Данный датчик будет иметь возможность передавать информацию о своём давлении и может принимать команду на изменение состояния имитации.

2. Перейдите внутрь *Sensor\_in\_PLC\_Type*. Добавьте на вход *параметр* типа Bool с именем SetImit и на выход *параметр* типа float с именем Pressure.

SandBox* Sensor_in_PLC_Type* += ×	
💊 Demo.Types.SandBox Type. Sensor in PLC Type Логический тип 💷 🚽	Demo.omx
→ 🕒 SetImit	Pressure 🏵 🚺

Теперь необходимо описать этот же датчик, но со стороны сервера ввода-вывода.

3. Перейдите в пространство имён *SandBox\_Type*. Создайте здесь *Логический тип* с именем Sensor\_in\_Server\_Type, задайте комментарий.



Для того, чтобы связать их между собой, необходимо указать свойство Представляемый тип.

4. Выделите *Логический тип Sensor\_in\_Server\_Type*, в свойстве Представляемый тип укажите <u>Sensor\_in\_PLC\_Type</u>. То есть он представляет тип, описанный со стороны ПЛК.





{} Demo.Types, SandBox, Type Пространство имён ⊯       Demo.ormx         ◆ Sensor_in_Server_Type Логический тип         ◆ Obugue         ◆ Obugue         ◆ Sensor_in_PLC_Type ™         ◆ Sensor_in_Server_Type	SandBox* SandBox_Type* + ×	🚰 Свойства	
AChect     Achec	{) <u>Demo.Types</u> <u>SandBox Type</u> Пространство имён ⊯ Demo.omx	Sensor_in_Server_Туре Логический тип Общие Базовый тип	
✓     ✓     Sensor_in_PIC_Type       ✓     Sensor_in_Server_Type       ✓     Sensor_in_Server_Type       ✓     Sensor_in_Server_Type	→C (1) Var_A Result → (1)	Представляемый тип	
◆ Sensor in PLC Type **         ◆ Sensor in Server Typ **         ● <tr< td=""><td></td><td>Имя</td><td>Sensor_in_Server_Type</td></tr<>		Имя	Sensor_in_Server_Type
	<ul> <li>Sensor in PLC Type *1</li></ul>	Ykaxkime nyte k site     Forck     P +=2PLC     P +=2PLC     P +=2Pster     = System     { } Types     { } SandBox_Type     Sensor_in,     \$ Sensor_in,     \$ Sensor_in.	менту X

5. Перейдите внутрь типа *Sensor\_in\_Server\_Type*, и для того, чтобы описать взаимодействие с логическим типом, описанным со стороны ПЛК, необходимо добавить ссылку на этот логический тип. (ссылки добавляются из Обозревателя решений). Перетяните сюда из Обозревателя решений *Sensor\_in\_PLC\_Type*. Нажмите по этой ссылке ПКМ → Экспонировать входы и выходы.

Тоиск 🔎 -	♦ <u>Demo.Types.SandBox Type</u> , <u>Sensor in Server Type</u> Логический тип	<ul> <li>_Sensor_in_PLC_Type</li> </ul>
<ul> <li>Image: PLC</li> <li>Image: PLC</li></ul>	<ul> <li>Sensor in PLC Type</li> <li>Sensor in PLC Type</li> <li>Course as <u>Bensor in PLC Type</u></li> <li>SetImit</li> <li>SetImit</li> </ul>	→ Pressure → 🕻

Свойство Представляемый тип гарантирует то, что в дальнейшем будет ссылка на объект, который и является экземпляром типа *Sensor\_in\_PLC\_Type*.

Теперь необходимо описать экземпляры в источнике и описать экземпляры в сервере.

6. Перейдите в *Application\_PLC* и перетяните сюда из Панели элементов 2 экземпляра *Sensor\_in\_PLC\_Type*, назовите их Sensor\_in\_PLC\_1 и Sensor\_in\_PLC\_2.

Application_PLC* + X	
<u>Demo.Domain.PLC.Imitator.</u> Ap	<u>plication_PLC</u> Приложение 🛒
✓ Sensor in PLC ■ Pressure → ④	ModbusAddressMap Карта адресов Modbus Файл: MB_Map.xml
Sensor in PLC 1	
→ B SetImit	Pressure (> (4)
Sensor in PLC 2 Tun: Types.SandBox Typ	⊫ <u>∓</u> be.Sensor in PLC_Type
→ 🚯 SetImit	Pressure 🏵 🙆

7. Перейдите в *SandBox* на *AstraServer* и перетяните сюда из Панели элементов 2 экземпляра *Sensor\_in\_Server\_Type*, назовите их Sensor\_in\_Server\_1, Sensor\_in\_Server\_2.





O 🔞 Parameter	Sensor in Server	Calc 1 Тип: Types SandBox Type Calc Type	e
	Pressure 🏵 🙆	→ <sup>(1)</sup> Var_A Result	- : 09
Calculator	<u>اً</u>	→ 🕜 Var_B	
→) 🔞 Var_A	Result_from_Handler 🏵 🥶	Calc 2	e
Vui_D		→ <sup>(1)</sup> Var_A Result	t OP
Calc from Link		→ 🕜 Var_B	
→Dug Var_A →Dug Var_B	Result_from_Handler 😁 🤒	Calc 3	1
Sensor in Server 1	<b>™</b> ()	→ @ Var_A Result	e t 🕀
Тип: <u>Types.SandBox_Ty</u>	pe.Sensor in Server Type	→	

Обратите внимание, экземпляры добавляются сразу с ошибкой. Всё потому, что ссылка внутри этого объекта должна быть проинициализирована. Для этого нужно указать свойство Представляемый объект.

8. Выделите Sensor\_in\_Server\_1, в свойстве Представляемый объект укажите <u>Sensor\_in\_PLC\_1</u> (Датчик №1 на стороне сервера ввода-вывода (потребитель) представляет данные с датчика №1 со стороны ПЛК (источника)). Сделайте то же самое для Sensor\_in\_Server\_2.

X == *		🚰 Свойства			
Domain.astra.AstraServer.AstraApplication. SandBox Логиче	еский объект 🖷 Demo.omx	✓ Sensor_in_S Общие	ierver_1 Логический объект		
O Parameter	Calc 1	Тип		Types.SandBo	x_Type.Sensor_in_Server_Type
	Тип: <u>Types.SandBox_Type.Calc_Type</u>	Уровень досту	па	публичный	
	→ 🙆 Var_A Result → 🙆	Область досту	па	глобальная	
<u>Calculator</u>	→ 🙆 Var_B	Аспект			
→ 🐨 Var_A Result_from_Handler → 😡	🧭 <u>Calc 2</u>	Представляем	ый объект		
≫ 1 Var_B	Тип: <u>Types.SandBox_Type.Calc_Type</u>	Имя	🎎 Укажите путь к элементу	×	er_1
Calc from Link	Var B		Поиск	,o -	
≫ 🐵 Var_A Result_from_Handler → 😡			Þ 📳 astra		
≫ 🔞 Var_B	Calc 3      Tug: Tuge: SandRey: Tuge Calc Tuge		Ilearning00		
Sensor in Server 1 📑 🗿	Min. Types.sanubox. Type.cat_Type		PLC     Minister		
Тип: Types.SandBox Type.Sensor in Server Type	Nor P		A Application_PLC		
→ 🚯 SetImit Pressure → 🚯			Sensor_in_PLC		
			Sensor_in_PLC_1		-
Sensor in Server 2		Представляем	Sensor_in_PLC_2		
		oobent, koropi	Да	Отмена	

9. Щёлкните ПКМ по пустому полю → Инициализировать все ссылки. Нажмите F5 для обновления страницы.

Если зайти внутрь логического объекта, ссылка **\_Sensor\_in\_PLC\_Type**, которая вела к какому-то типу, сейчас ведёт к конкретному объекту через инициализатор. Он указывает, на какой именно объект будет вести эта ссылка. И то, что вы указали в свойстве Представляемый тип гарантирует то, что ссылка будет вести к объекту, внутри которого точно есть набор параметров SetImit и Pressure.





<u>Demo.Domain.astra.AstraServer.AstraApplication.SandBox, Sensor in Server 1</u> Логический объект 🗐 Demo.omx	•
азовый тип: <u>Types.SandBox Type.Sensor in Server Type</u> Sensor in PLC_Type Ссылка на <u>Sensor in PLC_Type</u> ЭО 1) SetImit Pressure ЭО	Pressure 🎯 🕻
Domain.PLC.Imitator.Application_PLC.Sensor_in_PLC_1	

Теперь опишем транспортный уровень.

10.Перейдите в *Application\_PLC*, затем в *Kapmy адресов ModbusAddressMap*. Заполните карту также, как на изображении ниже. Сохраните карту и закройте её.

A	Application_PLC* 1 ModbusAddressMap* 부 ×									
% <u>-</u>	🐉 Demo.Domain.PLC.Imitator.Application PLC. ModbusAddressМар Карта адресов Modbus 폐									
	Сигнал	Тип	Привязка	Сегмент	Адрес	Номер бита				
¥	•	-	-	-	-	-				
6	Sensor_in_PLC.Pressure	float	непосредственно	Input Registers	48					
₿	Sensor_in_PLC_1.SetImit	bool	непосредственно	Coils	7	0				
6	Sensor_in_PLC_1.Pressure	float	непосредственно	Input Registers	50					
B	Sensor_in_PLC_2.SetImit	bool	непосредственно	Coils	8	0				
4	Sensor_in_PLC_2.Pressure	float	непосредственно	Input Registers	54					

11. Постройте решение, перейдите к развёртыванию и примените конфигурацию к линуксовому серверу.

Для наблюдения результата перейдите в *SandBox*. Если отправить команду SetImit = True, то имитация запустится, иначе – остановится.

L	🔾 🥶 Parame	ter (	)	Senso	r in Server		<u>Calc_1</u> Тип: <u>Types.Sa</u>	andBox	Type.	Calc_Type	=
				(30	Flessule Or	*	Var_A	0	0	Result C	» (
6	Calculator					· ) >	Var_B	0			
Э	Oug Var_A	10	0	Result_fr	om_Handler	به 🕼	Calc 2				
(1	Oug Var_B	0					Тип: <u>Types.S</u>	andBox	Type.	Calc_Type	
6	Calc from	Link				-	Var_A	0	0	Result C	» (
-	0 War_A	0	0	Result_fr	om_Handler	»@	Var_B	0			
7	O 🕜 Var_B	0					Calc 3	andRov	Turne	Cale Type	
6	Sensor in S	erver 1				*	Var A	0	0	Result (	<b>⇒</b> (
	Тип: <u>Types.Sa</u>	ndBox Typ	e.Senso	or in Server Ty	<u>ype</u>	+	Var B	0			
-	Представляе		ator.apr	100	Pressure (	→ <b>(</b> )					





## Применение аспектов. Использование сокетов

Постоянно строить представления для объектов неудобно, особенно когда этих объектов много. Для удобства построения представления и описания связи на различных уровнях с помощью мастера, используются **Аспекты**. **Аспекты** позволяют описать одни и те же объекты с разных сторон (точек зрения). То есть с разных уровней.

В **DevStudio** в решении описывается информационная модель обмена данными между компонентами. Следовательно, в проекте одни и те же объекты могут быть описаны на разных уровнях (в разных аспектах). Например, в демонстрационном проекте датчики будут описаны на среднем уровне (ПЛК) и на уровне сервера ввода-вывода (IOS). Для того, чтобы описывать объекты на разных уровнях, необходимо сначала добавить в проект уровни, чтобы в дальнейшем построить представления объектов с помощью специального мастера.

С этого момента начинается работа над демонстрационным проектом. В проекте будут описаны датчики и задвижки. Данные по датчикам будут передаваться по Modbus, а по задвижкам – по МЭК. Датчиков и задвижек будет 2 вида: простые и расширенные.

Простые датчики будут содержать на входе команду управления SetImit а на выходе сообщать о величине давления и о своём состоянии: нормальное, аварийное и предаварийное. Расширенные датчики будут содержать в себе всё то, что было у простых (наследование), а также будут дополнены уставками на чтение и на запись.

Простые задвижки на вход будут принимать команды открыть, закрыть и остановить, а на выходе будут сообщать информацию о своём состоянии (нормальное, аварийное или предаварийное) и о положении (открыта, закрыта, остановлена, в движении). Расширенные задвижки также будут наследовать структуру простых задвижек и будут дополнены параметром, характеризующим процент открытия задвижки.

В данной методичке будет описан процесс работы над созданием Простой задвижки и Расширенного датчика. Работа над частями проекта будет самостоятельной.

1. Перейдите в пространство Определения с помощью Обозревателя решений.

2. Перетяните сюда из Панели элементов **Пространство имён**. Назовите его Aspects, задайте комментарий.

<u>Demo</u> Содержимое модуля 🗐	Demo.omx
Ccuinka Ha Om.System	Appec: CentralNode
<pre>{} Types ==  {} {} SandBox_Type</pre>	{ } Aspects 🖬 👬 🎎 Комментарий 2 Папка с точками зрения, которые используются в проекте.
\	





3. Перейдите внутрь *Aspects* и перетяните сюда из панели элементов 2 экземпляра элемента *Аспект*. Назовите их PLC и IO\_Server, задайте комментарий.

Aspects* + X	
{ } <u>Demo. Aspects</u> Пространство имён	🛋 Demo.omx
РІС В Комментарий Х Точка зрения на типы и объекты со стороны ПЛК (источника). Да Отмена	<ul> <li>Ю_Server</li> <li>Комментарий Х</li> <li>Точка зрения, описывающая типы и объекты со стороны сервера ввода- вывода (потребителя).</li> <li>Да Отмена</li> </ul>

Начнём создание Простой задвижки. Её структура представлена на изображении ниже.



Как можно заметить, команды, положения и состояния сгруппированы в один элемент – сокет. *Тип сокета* – это группа параметров, которую можно использовать сколь угодно раз и где угодно.

Комментарий: сокет State общий для датчиков и задвижек.

Systêr

4. Перейдите в пространство имён *Туреs* и перетяните сюда из Панели элементов *Тип сокета*. Назовите его State, задайте комментарий.





5. Перейдите внутрь *сокета State* и добавьте сюда 3 *параметра* типа Bool: Normal, Warning и Failure.

6. Добавьте каждому из параметров атрибут Описания (нормальное состояние, предаварийное состояние, аварийное состояние) и генерацию события по установке (Объект перешёл в нормальное, предаварийное или аварийное состояние).

State* + ×	₹	🚰 Свойства						<b>-</b> ↓ ×
🧇 <u>Demo.Types. State</u> Тип сокета 💻 Demo.omx 🔹 🍕	ᢟ	○ Normal П	араметр					•
-		🔿 Общие						
		Тип				bool		
		Имя				Normal		
O 🕕 Warning		<b>.</b>				·		
		О Атрибуты						₹ џ ×
0 0 Failure		O Normai II	араметр					· ·
-	1	Атрибут				Значение		
		Описание	1.			Нормальное состояние		
		Атрибуты	События   И	1стория				
		🤞 События						- <b>∓</b> ∓ ×
		○ Normal П	араметр					
		🗌 Агрегиро	вать событи	19				
		🖌 Генерирс	вать событи	ия				
			Подуслови	ve (	Сообщение		Важность	Деактив
		✓	Установка		Объект перешёл н	в нормальное состояние	1	
			Снятие				1	
		Атрибуты	События И	Істория		and a second s	-periorperio	

7. Вернитесь в Пространство имён *Туреѕ* и создайте здесь новое *Пространство имён* с именем Valves. Задайте комментарий.







8. Перейдите внутрь *Valves* и создайте здесь *Тип сокета* с именем CMD, задайте комментарий.



9. Перейдите внутрь *сокета СМD* и перетяните сюда из Панели элементов 3 экземпляра *параметра* типа Bool. Назовите их Open, Close и Stop.

10.Добавьте каждому из *параметров* атрибут Описания (команда открыть, закрыть, остановить) и генерацию события по установке (подана команда на открытие, закрытие и остановку задвижки).

CMD* + X	<b>₹</b> 🔗 c	войства					<b>~</b> џ ×
Demo.Types.Valves, CMD Тип сокета      Demo.omx	😸 🔿 C	р <mark>еп П</mark> а	раметр				
	( A	трибуть	1				- ų ×
O 🚯 Open	00	р <mark>е</mark> п Па	раметр				
	Атри	бут		Значение			
	Опи	сание		Команда	Открыть		
O O Stop	Атри	буты	События Истор	ия	- hasten		
	d c	обытия	1				- û ×
	00	) <mark>pen</mark> Па	раметр				
	- A	прегиро	овать события				
		енерир	овать события				
		1	Подусловие	Сообщение		Важность	Деактивир
			Установка	. Подана команда на открыти	задвижки	1	1 E
			Снятие				1 E
	Атри	буты	События Истор	я	- handere		4

11.Вернитесь в Пространство имён *Valves* и создайте ещё один *Тип сокета*. Назовите его StateValve и задайте описание.







12.Зайдите внутрь *Типа сокета StateValve* и добавьте сюда 3 экземпляра *параметра* типа Bool. Назовите их Opened, Closed, Stoped и inProgress.

13.Добавьте каждому из **параметров** атрибут Описания (задвижка открыта, закрыта, остановлена и в движении) и генерацию события по установке (задвижка открылась, закрылась, остановилась и пришла в движение).

StateValve* - ×	🗢 🚰 Сво	ойства				
Demo.Types.Valves. StateValve Тип сокета  Demo.omx	🛞 🔾 Op	ened [	lараметр			
	🧿 Атр	оибуты				
O Dened	O Op	ened [	lараметр			
	Атриб	ут			Знач	ение
	Описа	ние			Заде	ижка открыта
O 😗 Stoped	Атриб	уты	События Истор	ия	hory	taun a hondr
O 🕒 inProgress	🦪 События					
	0 Op	ened l	Тараметр			
	Ar	региро	вать события			
	🗹 Ге	нерир	овать события			
			Подусловие	Сообщение		Важность
		~	Установка	. Задвижка откры	лась	1
			Снятие			4
	Атриб	уты	События Истор	я		de case se le materie

Теперь перейдём к созданию логических типов. Сначала опишем логический тип со стороны источника (PLC), а после – со стороны потребителя (IO\_Server).

14.Перейдите в Пространство имён *Valves* и создайте здесь *Логический тип*. Назовите его Valve\_PLC, задайте комментарий.







15.Выделите Логический тип Valve\_PLC и в свойстве Аспект укажите PLC (точка зрения источника).



16.Перейдите в Логический тип *Valve\_PLC* и перетяните сюда из Панели элементов на вход *сокет CMD*, а на выход 2 *параметра* типа Uint4. Назовите их State и StateValve.







17.Перейдите в **Пространство имён Valves** и создайте здесь **Логический тип**. Назовите его Valve\_IOS, задайте комментарий.



19.Выделите *Логический тип Valve\_IOS*, в свойстве Аспект укажите <u>IO\_Server</u>, в свойстве Представляемый тип – <u>Valve\_PLC</u> (задвижка со стороны сервера ввода-вывода (потребителя) представляет данные с задвижки, описанной со стороны ПЛК (источника)).

√ Valves* +¤ ×	🗧 😁 Свойства	
{} Demo.Types. Valves Пространство имён 🛒 Demo.omx	• 🎇 💊 Туре Логический тип	
	🔿 Общие	r.
🗇 CMD 🗉 🎇 🗇 StateValve 🗉 😤	Базовый тип	
O B Open O B Opened	Аспект	Aspects.IO_Server
O 🕕 Close O 🕒 Closed	Представляемый тип	Valve_PLC
O 🚯 Stop 🛛 O 🚯 Stoped	Имя	Valve_IOS
O 🕒 inProgress		
таксто сврска со ч	<u>ve</u>	
State Valve (* 19)		
( Subscop		

20.Зайдите внутрь **Логического типа Valve\_IOS**. Для того, чтобы описать взаимодействие с типом со стороны ПЛК, необходимо добавить ссылку, но теперь ссылку будем добавлять немного иначе. Щёлкните ПКМ по пустому рабочему полю → Добавить → Аспектную ссылку. Выберите представление, на которое необходимо добавить ссылку – Valve\_PLC → Далее → Далее. Выберите элементы ссылки, которые требуется экспонировать – Сокет СМD → Готово.





Valve_IOS* +¤ X						
<u>Demo.Types.Valves. Valve IOS</u> Логический тип <a>Demo.omx</a>						
► <u>CMD</u> Turi: <u>CMD</u> → ① Open → ① Close → ① Stop		Valve_PLC Ссылка на CCbIлка на CMD CMD CDpen Close Stop	alve PLC State ᠿ ᠿ StateValve ᠿ ᠿ			

21. *State* и *StateValve* будем разбивать на булевский параметры. Для этого из Панели элементов на выход перетяните *сокеты State* и *StateValve*. Теперь от *параметра State* типа Uint4 протяните связи к каждому булевскому параметру из *сокета State*. То же самое сделайте и для *параметров сокета StateValve*.

emo.Types.Valves, <u>Valve_IOS</u> Ло	гический тип 🖷 Demo.omx	_Valve_PLC	
CMD Turn: CMD D Open D Close D Stop	→ Valve_PLC Ccurrxa Ha Valve_PLC Ccurrxa Ha Valve_PLC State → ① StateValve → ① StateValve → ①		State       ◆         Turn: State       Normal → 1         Warning → 1       →         Failure → 1       →         StateValve       ✓         Opened → 1       Closed → 1         Stoped → 2       →         inProgress → 3       →

**DevStudio** ругается на эту связь, так как нельзя из целочисленного параметра передать информацию булевскому параметру. Чтобы это исправить, необходимо каждую из этих связей преобразовать.

22.Выделите по очереди каждую из связей и в свойстве Преобразование установите <u>Получить бит</u>. Свойство Номер бита каждой связи укажите в соответствии с изображением.



Объектно-ориентированный подход с точки зрения типизации описан. Теперь необходимо добавить экземпляры со стороны ПЛК и со стороны сервера ввода-вывода, т.е. сначала источник, потом потребитель.

23. Перейдите в *Aplication\_PLC*. Добавьте сюда *Логический объект*, назовите его root. Добавьте ему комментарий и атрибут Описание.





🗔 Обозреватель решений 🔹 म 🗙	Valve_IOS* Application_PLC* += ×	🚰 Свойства
Поиск 🔎 –	🖧 <u>Demo.Domain.PLC.Imitator. Application PLC</u> Приложение 🔍 Demo.omx 🔹 🔆	Object Логический объект
Панель элементов + 4 ×		@ Атрибуты
Valve_PLC <aspects.plc></aspects.plc>	Sensor in PLC      ModbusAddressMap     Pressure      Kapra agpecce Modbus	<ul> <li>Оbject Логический объект</li> <li>Атрибут</li> <li>Значение</li> </ul>
Valve_IOS <aspects.io_servi () Om.PLC () Om.Server</aspects.io_servi 	Qaikr: MB_Map.xml 200       ✓ Sensor in PLC 1       THM: Types.Sensor in PLC Type       THM: Types.Sensor in PLC Type	Описание Демо-проект Атрибуты События История
Вычисления     Карты адресов     Ссылки     Физический объект     Логоческий объект	ЭС Settinit         Pressure         ЭСП           У Sertori In Diff.         III         III           У Sertori In Diff.         IIII         IIIII           У Di Settinit         Pressure         IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	
Параметр     Обытие     Опраметр     Ороо     Опт	Да Отиена	
© int2 © uint2 © int4		

- 24. Зайдите внутрь *root* и создайте здесь *Логический объект*. Назовите его Valve и добавьте ему атрибут Описание, в нём запишите Задвижки.
- 25. Зайдите внутрь *root* и перетяните сюда из Панели элементов 3 экземпляра типа *Valve\_PLC*. Назовите их Valve\_1, Valve\_2, Valve\_3. Задайте каждому из экземпляров атрибут описания (Задвижка №1, Задвижка №2, Задвижка №3).



26. Перейдите в *AstraApplication*, кликните ПКМ по пустому рабочему полю → Создать здесь представления объектов. Выберите полностью объект root из PLC → Далее → Далее → Укажите аспект новых представлений (Aspects.IO\_Server, так как вы находитесь внутри сервера ввода-вывода) → Готово.



Если зайти внутрь любой из задвижек со стороны сервера ввода-вывода, то можно увидеть, что ссылка внутри объекта проинициализирована (ведёт к конкретному объекту со стороны ПЛК, связанному с ним одним представлением).





овый тип: <u>Types.Valves.Valve_IOS</u>		
▶ <u>CMD</u> (>) Тип: <u>CMD</u> Э) © Open ⇒) © Close ⇒) © Stop	Image: State PLC         Ccunxa Ha Valve PLC         Ccunxa Ha Valve PLC         State PLC         State PLC         Image: PLC </th <th><ul> <li>State ← Twn: State</li> <li>Normal → 0</li> <li>Warning → 0</li> <li>Failure → 0</li> <li>StateValve</li> <li>Twn: StateValve</li> <li>Opened →</li> <li>Closed →</li> <li>Stoped →</li> <li>inProgress →</li> </ul></th>	<ul> <li>State ← Twn: State</li> <li>Normal → 0</li> <li>Warning → 0</li> <li>Failure → 0</li> <li>StateValve</li> <li>Twn: StateValve</li> <li>Opened →</li> <li>Closed →</li> <li>Stoped →</li> <li>inProgress →</li> </ul>

Объектно-ориентированный подход описан, теперь необходимо описать транспортный уровень. Информация о задвижках передаётся по 104 МЭКу.

27. Перейдите в *Application\_PLC*, перетяните сюда из Панели элементов *Kapmy adpecoв MЭK*. Нажмите на файлик внутри и создайте файл с именем IEC\_Map.



28. Зайдите внутрь *Карты адресов lecTmAddressMap*. Для заполнения карты адресов воспользуйтесь уже готовым файлом, содержащим в себе все адреса. Для этого нажмите на кнопку Загрузить из файла и выберите файл iec\_export\_Map\_Valves.xlsx. Если имена всех параметров и объектом совпадают с тем, как описано в методичке, то все необходимые адреса импортируются. Сохраните карту и закройте её.





🎎 Demo - DevStudio					,	
Файл Правка Вид Проект Справка				Загрузить из	файла	
🕒 🔿 🕇 C 🔁 🔁 🖶 🖨 🐧 🖒	лока	льная 🔹 🔛 🖶 📮 🔽 🖻	🛛 🏷 🛛 🔤	a 🔁 🛓 🏶 🦂	€ <b>0</b> -	* * * * *  = +
🗔 Обозреватель решений 🔹 🔻 🗙	F	Application_PLC* lecTmAdd	ressMap*	-¤ ×		
Поиск 🔎 –	8	Demo.Domain.PLC.Imitator.App	olication	PLC. lecTmAddre	ssMap Ka	рта адресов МЭК 60870-5
🖏 EthernetAdapter 🔷		Сигнал	Тип	Привязка	Адрес	Протокольный тип
🖻 🧃 Historian	X	-	-	-	•	-
📥 Ethernet	۵	Sensor_in_PLC.Pressure	float	не привязан		
learning00	0	Sensor_in_PLC_1.SetImit	bool	не привязан		
FthernetAdapter	۵	Sensor_in_PLC_1.Pressure	float	не привязан		
▲ Q Imitator	0	Sensor_in_PLC_2.SetImit	bool	не привязан		
Application_PLC	۵	Sensor_in_PLC_2.Pressure	float	не привязан		
😪 lecTmAddressM	0	root.Valves.Valve_1.CMD.Open	bool	непосредственн	15	45: C_SC_NA_1
S ModbusAddres	0	root.Valves.Valve_1.CMD.Close	bool	непосредственн	16	45: C_SC_NA_1
P Q root A Second in BLC	B	root.Valves.Valve_1.CMD.Stop	bool	непосредственн	17	45: C_SC_NA_1
Sensor_in_PLC	•	root.Valves.Valve_1.State	uint4	непосредственн	33	7: M_BO_NA_1
Sensor in PLC 2	•	root.Valves.Valve_1.StateValve	uint4	непосредственн	1	7: M_BO_NA_1
WodbusTcpSlave	0	root.Valves.Valve_2.CMD.Open	bool	непосредственн	18	45: C_SC_NA_1
- PLC	0	root.Valves.Valve_2.CMD.Close	bool	непосредственн	19	45: C_SC_NA_1
C Saniar	0	root.Valves.Valve_2.CMD.Stop	bool	непосредственн	20	45: C_SC_NA_1
	•	root.Valves.Valve_2.State	uint4	непосредственн	2	7: M_BO_NA_1
	•	root.Valves.Valve_2.StateValve	uint4	непосредственн	3	7: M_BO_NA_1
	₿	root.Valves.Valve_3.CMD.Open	bool	непосредственн	21	45: C_SC_NA_1
	₿	root.Valves.Valve_3.CMD.Close	bool	непосредственн	22	45: C_SC_NA_1
	B	root.Valves.Valve_3.CMD.Stop	bool	непосредственн	23	45: C_SC_NA_1
	•	root.Valves.Valve_3.State	uint4	непосредственн	4	7: M_BO_NA_1
	•	root.Valves.Valve_3.StateValve	uint4	непосредственн	5	7: M_BO_NA_1

Раз появилась новая карта адресов, необходимы также соответствующие станция и опросчик.

29. Поднимитесь на уровень выше (в *Imitator*), перетяните сюда из Панели элементов *Станцию МЭК 60870-5-104*. В её свойствах укажите Номер станции <u>1</u> и выберите карту адресов <u>lecTmAddressMap</u>.



30. Перейдите в AstraServer и перетяните сюда из Панели элементов Опросчик Мэк 60870-5-104.







99

#### 31. Постройте решение, перейдите к развёртыванию и примените конфигурацию к линуксовому серверу.

Для наблюдения результата перейдите в *AstraServer*  $\rightarrow$  *root*  $\rightarrow$  *Valves*.

10. Domain. astra. Astra Se	rver.AstraApplication.root. \	<u>/alves</u> Логический объе	<mark>ект 🛋 Demo.om</mark> x	<ul> <li>Valve_3</li> </ul>	<ul> <li>astra.AstraServe</li> </ul>
Valve 1 <aspects. Тип: <u>Types.Valves.Va</u> Представляет: <u>PLC.I</u></aspects. 	IO_Server>	Valve 2 <aspects. Тип: <u>Туреs.Valves.Va</u> Представляет: <u>PLC.I</u></aspects. 	IO_Server> IVe IOS mitator.Application_PLC.root.Va	Valve 3 <aspects.ii Тип: <u>Туреs.Valves.Valv</u> Представляет: <u>PLC.In</u></aspects.ii 	O_Server> III ve_IOS nitator.Application_PLC.root.Valve
A 🕨 CMD	▲ State 🐗	A 🕨 CMD	▲ State 🐗	🔺 🕨 CMD	▲ State 🖛
🔊 🕕 Open 🗹	✓ Normal → 🕕	🔊 🚯 Open 🗌	✓ Normal → 🕕	→ 🕒 Open 🗸	✓ Normal → 🕕
→ 🕒 Close 🗌	🔲 Warning 🏵 🕕	→ 🕒 Close 🔽	□ Warning → 🕕	🔊 🕒 Close 🗌	🗌 Warning 🏵 🕒
🔊 🕒 Stop 🔲	🔲 🛛 Failure 🏵 🕕	🔊 🕒 Stop 🔲	🗌 🛛 Failure 🕀 🕕	🔊 🕕 Stop 🔲	🗌 🛛 Failure 🏵 🕕
	▲ StateValve <		🔹 StateValve 🐗		▲ StateValve 🐗
	✓ Opened → 🕒		□ Opened → ①		✓ Opened →
	Closed 🕀 🕒		Closed 🔿 🕕		Closed 🕀 🕒
	□ Stoped → 🕕		Stoped 🕀 🕒		Stoped 🕀 🕒
. ce	inProgress 🕀 🕒		inProgress 🏵 🕒		🔲 inProgress 🏵 🕕

Самостоятельное задание. Создать тип датчика со стороны ПЛК и со стороны сервера ввода вывода, разместить экземпляры в папке *root* в ПЛК, создать представления объектов на *AstraServer*, импортировать карту связывания (данные по датчикам передаются по Modbus, поэтому ничего нового создавать не нужно. Следует лишь дополнить уже существующую карту новыми адресами). Структура типа датчика описанного со стороны сервера ввода-вывода и конечный результат представлены на изображениях ниже.



#### Наследование

Добавим в проект расширенные датчики и задвижки, используя наследование. Наследование позволяет описать ещё один тип, используя уже ранее описанный тип, дополнив его новыми параметрами. Для начала подготовим набор параметров, который будет использован.





Начнём с расширенного датчика. Структура типа, описанного со стороны сервера ввода-вывода представлена ниже.



1. Перейдите в *Пространство имён Sensors* (Определения → Types → Sensors). Перетяните сюда из панели элементов *Тип сокета*. Назовите его Presets и добавьте комментарий.



2. Перейдите внутрь *сокета Presets* и добавьте сюда 4 экземпляра *параметра* типа float. Назовите их HiHiLimit, HiLimit, LoLimit, LoLoLimit. Добавьте для всех атрибуты описания (Верхняя аварийная уставка, верхняя предаварийная уставка, нижняя предаварийная уставка, нижняя аварийная уставка) и единицы измерения (МПа). Также поставьте галочку около Сохранять историю.

🗔 Обозреватель решений	- 4 ×	Presets* += X	🗧 🚰 Свойства	
Поиск	,o -	🗇 <u>Demo.Types.Sensors. Presets</u> Тип сокета 💷 Demo.omx 🔹	😸 🔿 НіНіLimit Параметр	
🕜 Панель элементов	<b>-</b> ₽ ×		Общие	
Поиск	,o -		ФТРИБУТЫ	
🖻 🍠 Событие			O HiHiLimit Параметр	
<ul> <li>Параметр</li> </ul>			Атрибут	Значение
U) bool		O 🚺 LoLimit	История	Enable="True" ServerTime="False
10 int8		🔿 🙆 LoLoLimit	Описание	Верхняя аварийная уставка
10 uint8			Атрибуты События История	
( float			Alphoylar Coolina Alcopia	

3. Перейдите в *Пространство имён Sensors,* скопируйте и вставьте *Тип сокета Presets*. Измените название на WritePresets и комментарий.





Sensor PLC	<aspects.plc> 🛋 🌸</aspects.plc>	Sensor IOS <asp Представляет: Sen</asp 	ects.IO_Server> 🗉 🎇
Securit	Pressure 🕞 🚺	→ 🕕 Setlmit	Pressure 🗩 🕼
🖗 <u>Presets</u> 🛋 🛞	WritePresets	*	Normal 🏵 🕕
O 🙆 HiHiLimit	O 🙆 HiHiLimit		Warning 🕀 🕕
	O ( HiLimit	🎎 Комментарий	×
O 🙆 LoLoLimit		Группа параметров с устав	ками на

4. Перейдите внутрь *Сокета WritePresets*. Перед имеющимся именем каждого параметра добавьте букву «w», а перед описанием – слово «команда».

WritePresets* += X	🚰 Свойства	
🧇 <u>Demo.Types.Sensors. WritePresets</u> Тип сокета 💷 🛛 Demo.omx 💽 😤	🔿 wHiHiLimit Параметр	
	Общие	
O 🙆 wHiHiLimit	ФТРИБУТЫ	
O 🙆 wHiLimit	O wHiHiLimit Параметр	
	Атрибут	Значение
	История	Enable="True" ServerTime="False"
O 🙆 wLoLoLimit	Описание	команда Верхняя аварийная уставка
	Атрибуты События История	

Начнём создавать логические типы.

5. Перейдите в *Пространство имён Sensors*, перетяните сюда из Панели элементов *Логический тип*. Назовите его ExSensor\_PLC, задайте комментарий.

. <u>Types. Sensors</u> Логический тип 💷 Demo.om)	ExSensor_PLC
<ul> <li>Sensor PLC <aspects.plc> ■ SetImit State → Pressure → Pressure → Pressure → Sensor IOS <aspects.io_server> ■ Pressure → Tipegcrae.naer: Sensor PLC → O SetImit Pressure → Normal → Warning → Failure → I Failure → I SetImit → State ≪ Normal → Failure →</aspects.io_server></aspects.plc></li></ul>	ExSensor PLC       Image: Constraint of the image: Cons

6. Выделите *Логический тип ExSensor\_PLC*, в свойстве Аспект укажите <u>PLC</u>, Базовый тип – <u>Sensor\_PLC</u>.

Свойство Базовый тип отвечает за то, что именно вы будете наследовать, какую структуру и какие данные будете переносить.







7. Перейдите в *Логический тип ExSensor\_PLC*. Перетяните сюда из Панели элементов на вход *Тип сокета WritePresets*, а на выход – *Presets*.

📮 Обозреватель решений	• 4 × /	ExSensor_PLC* += ×		
Поиск	P - Q	Demo.Types.Sensors. ExSensor PLC Логический тип 🖷	Demo.omx	•
🔐 Панель элементов	<b>~</b> ↓ ×	Базовый тип: Types.Sensors.Sensor PLC		*****
Поиск <b>4 {}</b> Demo	<u>م</u> ^	→ i Setlmit		Pressure 🕞 🙆
A { } Types				State 🕞 🔞
Figure 1 SandBox_Type				State of g
🧇 State				
Valves				
▲ {} Sensors		WritePresets		Tun: Presets
🖉 ExSensor_PLC <	Aspects.PLC:	N. 0. 1999 - 3		
🖉 Sensor_IOS <as< td=""><td>pects.IO_Ser</td><td></td><td></td><td>HIHILIMIT CO</td></as<>	pects.IO_Ser			HIHILIMIT CO
Sensor_PLC < As	spects.PLC>	30 WHILIMIT		HiLimit 🗇 😳
Presets		→) 😳 wLoLimit		LoLimit 🔿 🕼
		→ W wLoLol imit		LoLoLimit 🏵 🙆

Входные и выходные параметры со стороны ПЛК описаны. Теперь необходимо описать логический тип со стороны сервера ввода-вывода.

8. Перейдите в *Пространство имён Sensors*. Перетяните сюда из Панели элементов *Логический тип*. Назовите его ExSensor\_ISO, задайте комментарий.

<u>Demo.Types.</u> Sensors Пространство имён	Demo.omx	<ul> <li>ExSensor_IOS</li> </ul>	
<ul> <li>Sensor PLC <aspects.plc> III</aspects.plc></li></ul>	<ul> <li>►<u>XSensor PLC</u> <aspects.plc> ■ Базовый тил: <u>Sensor PLC</u></aspects.plc></li> <li>SetImit</li> <li>WritePresets</li> <li>WrHiHiLimit</li> <li>WHILimit</li> <li>W vLoLumit</li> <li>WLOLDLimit</li> </ul>	<ul> <li>ExSensor IOS</li> <li>Комментарий</li> <li>Тип расширенного датчика со стороны сервера ввода-вывода (потребитель), наследующий логику простого датчика)</li> <li>Да Отмена</li> </ul>	

9. Выделите *Логический тип ExSensor\_IOS*. В свойстве Аспект укажите <u>IO\_Server</u>, Представляемый тип – <u>ExSensor\_PLC</u>, Базовый тип – <u>Sensor\_IOS</u>.





Sensors* + X						₹	🚰 Свойства	
() <u>Demo.Types. Sensors</u> Пространство имён 🛒	Demo.omx		▼ Ex	Sensor_IOS		- *	ExSensor_IOS Логический	і тип
Sensor PLC <aspects.plc> =      Pressure →      SetImit      Pressure →      Normal →      Warning →      Failure →      Discontinue →      Pressure →</aspects.plc>	ExSensor PLC <aspects. Essoewin mrs Sensor PLC     Sensor PLC     WritePresets     WritePresets     WriteInitit     WiteInitit     WiteInitit     WiteInitit     Welcolumit     Loco</aspects. 	PLC> =1 State → ① essure → ② Presets ← iLimit → ③ bLimit → ③ bLimit → ③	<ul> <li><u>ExSensor IOS</u> «Aspet Basebuil runt <u>Sensor IO</u> Представляет: <u>ExSenso</u></li> <li>SetImit</li> </ul>	tts:IO_Server> = 25 25 Pressure () Pressure () C State () Normal () Warning () Failure () S	Presets      Set     G HiHiLimit     G LoLimit     G LoLimit      G LoLimit      G LoLimit      G whiHiLimit     G whILimit     G whILimit     G whILimit     G whILimit     G whILimit     G whILimit		Общие Базовый тип Аспект Представляемый тип Имя	Sensor IOS Aspects.IO Server ExSensor_PLC ExSensor_IOS

10.Перейдите внутрь *Логического типа ExSensor\_IOS*. Щелкните ПКМ по пустому рабочему полю → Добавить → Аспектную ссылку. Представление, на которое необходимо будет ссылаться – *ExSensor\_PLC*, Далее → Далее, Элементы ссылки которые требуется экспонировать – *WritePresets* и *Presets*.

emo.Types.Sensors. ExSensor IOS Логи	ческий тип 💷 Demo.omx	_ExSensor_PLC	
зовый тип: <u>Types.Sensors.Sensor IOS</u> € ]] SetImit	a> _Sensor_PLC Ссылка на <u>Sensor_PLC</u> → Э 3 SetImit State Э Ф Pressure Э Ф		→ State Trin: State Normal → B Warning → B Failure → B
WritePresets Twn: WritePreset	a LexSensor_PLC Cosuka Ha ExSensor PLC Cosuka Ha ExSensor PLC → ③ SetImit State → → ④ WritePresets → ④ WHiHiLimit → ④ ④ WHILimit → ④ ④ WLoLimit → ④ ● WLOLIMIT		Pressure Pressure Presets Tun: Presets HiHiLimit HiLimit LoLimit LoLoLimit (

Так работает наследование. Логику родителя поменять нельзя, но зато можно добавить что-то новое. Но если что-то изменится у родителя, все наследники унаследуют эти изменения.

Разместим экземпляры на стороне источника и потребителя.

11.Перейдите в *Application\_PLC* → *root* → *Sensors*. Перетяните сюда из Панели элементов 3 экземпляра типа *ExSensor\_PLC*. Назовите их ExSensor\_1, ExSensor\_2, ExSensor\_3. Каждому из них добавьте атрибут Описание (Расширенный датчик №4, 5 и 6).





ors* +¤ X		🗧 🚰 Свойства	
o.Domain.PLC.Imitator.Application	PLC.root. Sensors Логический объект 🛒 Demo.omx	🔹 ExSensor_4 🔄 🔹 🖌 🖌 🔹	кт
		Общие	
Sensor 1 <aspects.plc></aspects.plc>	Sensor 2 <aspects.plc></aspects.plc>	ts.PLC> III	Types.Sensors.ExSensor_PL
Тип: Types.Sensors.Sensor_PLC	Тип: Types.Sensors.Sensor_PLC Tип: Types.Sensors	Sensor PLC Уровень доступа	публичный
→ 🕕 SetImit State → 😰	→ 🕕 SetImit State → 👘 → 🕕 SetImit S	tate 🗇 🔞 Область доступа	глобальная
Pressure 🏵 🕻	Pressure 🗇 🙆 Pres	sure Э	Aspects.PLC
		Представляемый объект	
Tun: <u>Types.Sensors.ExSensor PLC</u>	Tun: Types.Sensors.ExSensor PLC Tun: Types.Sensor Sensor Sensor PLC	Aspects.PLC> MMR MRR	ExSensor_4
🔊 🕕 Setlmit 🛛 State 🏵 🕻	→ 🚯 SetImit State → 🔞 → 🚯 SetImit	State 🗇 🔞	
∧ IN WritePresets Pressure → ●	A 🕨 WritePresets Pressure 🎯 🙆 🔺 🕨 WritePre	esets Pressure 🏵 🔞 🧳 ExSensor_4 Логический объе	ĸt
🔊 🕼 wHiHiLimit 🔷 Presets ◀	🔊 🚳 wHiHiLimit \land Presets 🖛 🔊 🚳 wHiHiLir	nit 🔨 Presets 🐗 Атрибут	Значение
🗩 🙆 wHiLimit 🛛 HiHiLimit 🏵 🕻	→ 🕼 wHiLimit HiHiLimit → 🕼 → 🕼 wHiLimit	НіНіLimit Э	Расширенный датчик №4
Ð 🙆 wLoLimit 🛛 HiLimit 🔾	→ 🕼 wLoLimit HiLimit → 🕼 → 🕼 wLoLimi	t HiLimit 🗇 🙆	
🔊 🕼 wLoLoLimit 🛛 LoLimit 🏵 🕻	🔊 🕼 wLoLoLimit 🛛 LoLimit 🏵 🕼 🖈 🕼 wLoLoLi	mit LoLimit 🍛 🙆	
LoLoLimit 🗇 🕻	LoLoLimit 🕀 🚺	LoLoLimit 🕀 🙆	

12.Перейдите в **AstraApplication**. Щёлкните ПКМ по пустому месту на рабочем поле → Создать здесь представления объектов. Выберите всю папку **root** из PLC → Далее → Далее → Аспект новый представлений – Aspects.IO\_Server → Готово.

13.Перейдите в *Application\_PLC*, откройте *Kapmy адресов ModbusAddressMap*, импортируйте файл mb\_export\_map\_Sensors.xlsx. Сохраните и закройте карту адресов.

14. Постройте решение, перейдите к развёртыванию и примените конфигурацию к линуксовому серверу.

Для наблюдения результата перейдите в *AstraServer*  $\rightarrow$  *root*  $\rightarrow$  *Sensors*.

Самостоятельное задание. Создать тип расширенной задвижки со стороны ПЛК и со стороны сервера ввода вывода, разместить экземпляры в папке *root* в ПЛК, создать представления объектов на *AstraServer*, импортировать карту связывания (данные по датчикам передаются по МЭК, ничего нового создавать не нужно. Следует лишь дополнить уже существующую карту новыми адресами). Структура типа расширенной задвижки описанной со стороны сервера ввода-вывода и конечный результат представлены на изображениях ниже.

ExValce_IOS* += ×			🗧 🚰 Свойства	
Demo.Types.Valves. ExValce_IOS	Логический тип 💷 Demo.omx	: 💽 🛃	OpenPrc Параметр	
Базовый тип: Types.Valves.Valve IOS			🔿 Общие	
E CMD	Valve PI C	State State	Уровень доступа	публичный
Тип: <u>CMD</u>	Ссылка на <u>Valve_PLC</u>	Тип: <u>State</u>	Область доступа	глобальная
Den Open	A 🕨 CMD State 🕞 🔞	Normal > 1	Направление	выход
→ 🕕 Close	→ 🕕 Open StateValve → 🚳	Warning 🏵 🕕	Тип	uint4
🔊 🕕 Stop	→ 🕒 Close	Failure 🏵 🕒	Имя	OpenPrc
			Атрибуты	
عام _ExVa	alve_PLC	→ OpenPrc → 😡	Э OpenPrc Параметр	
Ссылк	ta Ha ExValve_PLC	00	Атрибут	Значение
^ ► CI	MD State 🗇 🕼		Описание	Процент открытия задвижи
*) <b>[</b> ]0	pen State valve 🐨 🐨		Единицы измерения	%
+⊃ [3 Cl  →] [3 St	op opening of the second s			L





Valves + × ✓ Demo.Domain	.astra.AstraServer.AstraAppli	cation.root. Valves Логическ	кий объект 💷 Demo.o	mx • ExValve 6	▼ astra.AstraServer.A
✓ Valve 1 < Aspect           Тип: Турез. Valves           Представляет: Р           ▲ В Ореп           ⇒ В Ореп           ⇒ В Сюзе           ⇒ В Stop	tts.IO_Server> Kalve IOS ICImitator.Application PLC.root Val Varning () Failure () Failure () Closed () Closed () Stoped () Stoped () ()	<ul> <li>✓ Valve 2 <aspects.io_si Twin: Types.Valves.Valve I/ Представляет: PLC.Imitat</aspects.io_si </li> <li>▲ ▷ CMD</li> <li>&gt; ① Open</li> <li>&gt; ② Close</li> <li>&gt; ③ Stop</li> </ul>	erver> S S S S S S S S S S S S S S S S S S S	Valve 3 <aspects.io_se open="" plcimitate="" stop<="" th="" tipeg.crass.net:="" tipeg.valves.valve.io="" twn:="" types.valves.valve.io=""><th>Ver&gt; S State Normal S Variation PLC.root.Valve Normal S Variation Variation StateValve Vopened S Closed S Stoped S Stoped S Stoped S S S S S S S S S S S S S S S S S S S</th></aspects.io_se>	Ver> S State Normal S Variation PLC.root.Valve Normal S Variation Variation StateValve Vopened S Closed S Stoped S Stoped S Stoped S S S S S S S S S S S S S S S S S S S
ExValve 4 <aspe Тип: <u>Турез.Valves.</u> Представляет: <u>PL0</u></aspe 	inProgress →      i	ExValve 5 <aspects.io_s< td="">           Тип: <u>Турез.Valves.ExValce_IC</u>           Представляет: <u>PLC.Imitator</u></aspects.io_s<>	inProgress (*)	ExValve 6 <aspects.io_s Тип: <u>Types.Valves.ExValce.In</u> Представляет: <u>PLC.Imitator</u></aspects.io_s 	inProgress () erver> OS :Application_PLC.root.Valves.
	A State     V     Normal → 3     Warning → 3     Failure → 3     StateValve ←     Opened → 3     Opened → 3				State      State      Vormal      O     Varning      O     Failure      O     StateValve      Opened      O     Opened      O
	Closed → 1 Stoped → 1 inProgress → 1 100 OpenPrc → 1		Closed → 0     Stoped → 0     inProgress → 0     OpenPrc → 0		Closed → 1 Stoped → 1 inProgress → 1 OpenPrc → 1

Конфигурация сервера ввода-вывода, взаимодействующим с имитатором ПЛК готова, теперь можно приступить к разработке мнемосхем.

# 9. SePlatform.HMI

Для того, чтобы организовать взаимодействие пользователей с технологическими данными, необходимо обеспечить графический интерфейс. Для разработки и визуализации человеко-машинного интерфейса в составе Systeme Platform платформы есть компонент *SePlatform.HMI*. В демонстрационном проекте мы визуализируем получаемые сервером данные. Перед началом работы необходимо установить *SePlatform.HMI* на машину в ОС Windows и Linux:



Установите *SePlatform.HMI* на Windows при помощи запуска дистрибутива, а на Linux через командер: перейдите в PuTTY, удостоверьтесь, что Вы находитесь в директории с дистрибутивами и после этого введите команду: sudo dpkg –i SePlatform.hmi\*\*\*\*\*\*\*.deb.





#### Создание проекта

Для начала, создадим простой проект, и запустим его.

- 1. Запустите *SePlatform.HMI*. Пуск  $\rightarrow$  SePlatform  $\rightarrow$  SePlatform.HMI.
- 2. Нажмите Файл  $\rightarrow$  Создать проект  $\rightarrow$  Назовите его «DemoHMI»  $\rightarrow$  Сохранить.

Проект создан, в нём уже присутствует форма по умолчанию MainForm. Для начала создадим пустую форму и поработаем с несколькими базовыми элементами *SePlatform.HMI*.

# Добавление экранной формы

Создадим пустую экранную форму.

1. В Библиотеке компонентов раскройте список Проект → нажмите ПКМ по Экранные формы → Создать → Визуальные → Форма.

Библиот	тека компонентов		8	× Maci	ер создания элеми	ентов	
Фильтр					ер создания элем	CHIOD	
>	Оби	цие элементы		Новый	элемент	(	Создание нового элемента
~	Прое	кт <demohmi></demohmi>		Значен	ия атрибутов	Т	Гип элемента Тип
Гло	обальные объекты					5	Базовый тип элемента
Тип	ы						
> Экр	оанные формы	Сортировка	•				> 🍰 AP · · · · · · · · · · · · · · · · · ·
		Создать	Ctrl+T				> 🍪 SQL
		Новая папка					> 🛞 Базовые
							Визуальные
							Вес элемента разметки
							Выпадающий список
							□ ← Выравнивание элемента разметки
							А Текст
							📐 Точка трансформации
							🗹 Флажок
							🛅 Форма
							🔁 Фрейм
							<ul> <li>Эллипс</li> </ul>
							> 🛞 Встроенные
							> 🚷 Графики
							> 👶 Окружение .
							ОК Отмена

Создана пустая экранная форма. Теперь можно работать с её свойствами.

2. Откройте созданную экранную форму двойным нажатием ЛКМ.

3. Нажать ЛКМ в структуре объекта на Form\_1. В редакторе свойств появятся свойства выбранного объекта.

_				
Ии	мя	Описани	e	
	Form_1	Тип на о	снове Форма	
Þe,	дактор свойств			
2				
Св	войство		Харак	Значение
	Отображаемое и	мя		Form_1
	🔎 Модификатор до	оступа		📑 Приватный
>	68 X		RW	0
>	<b>(B)</b> Y		RW	0
>	🔞 Z-значение		RW	0
>	🔞 Угол поворота		RW	0
>	🔞 Масштаб		RW	1
>	Видимость		RW	true
>	🔞 Непрозрачность		RW	1
>	Включено		RW	true
>	Фокус ввода		RW	<не определен
>	💧 Всплывающая п	одсказка	RW	
>	🔞 Ширина		RW	400
>	🔞 Высота		RW	300

4. Для того, чтобы в RunTime форма открывалась с нужным названием, в свойстве Заголовок окна введите <u>SandBox</u>. Завершите ввод нажатием клавиши Enter.





5. В свойстве Высота укажите 600.

Структура объекта						
Имя Описание						
Torm_1	Тип на основе Фо	рма				
Редактор свойств						
۵,						
Свойство		Харак	Значение			
> 🔨 Ширина окна	RW	1920				
> 📵 Высота окна			975			
> S Заголовок окна			SandBox			
> В Показывать заголовок			true			
> 📵 Показывать кнопку "свернуть" 🛛 🤉 🛚 R W true			true			
Редактор свойств События						

- 6. В свойстве Размеры окна укажите Автоподбор.
- 7. В свойстве Положение окна укажите По центру монитора.

Редактор свойств				
Свойство	Харак	Значение		
<ul> <li>и Размеры окна</li> <li>и Положение окна</li> </ul>	R W R W	Автоподбор По центру монитора		
<ul> <li>Ф Режим обработки закрытия окна</li> <li>Путь к файлу иконки</li> </ul>	R W R W	Закрыть		
Редактор свойств События				

Теперь форма стандартного размера, и её можно запустить, но прежде установите форму главной: ПКМ в левой части экрана по Form\_1 → Установить форму главной.



8. Нажать Проект -> Показать главную форму в RunTime.








В результате получим пустую запущенную экранную форму. На данный момент, на экранной форме нет никакой логики и элементов.



Пустая экранная форма создана, добавим на экранную форму элементы.

#### Добавление элементов

Познакомимся с некоторыми возможностями *SePlatform.HMI* на примере калькулятора. Добавим на экранную форму два поля ввода и обработчики, которые при нажатии на кнопку выведут сумму двух переменных в текстовое поле различными способами.

1. Перетяните на экранную форму из Визуальных элементов Библиотеки компонентов 2 **поля ввода**, **кнопку** и элемент **Текст**. Назовите в соответствии с изображением.

Form_1*			Структура объекта			
Edit	Button	Text		Имя ~ 🛅 Form_1 ~ Графические об	Описание Тип на основ бъекты	ве Форма
				■ TextEdit ■ TextEdit ■ Button_( A Text_Res	_VarA Поле ввода _VarB Поле ввода Calculate Кнопка sult_Om Текст	

2. В свойстве Текст *Поля ввода VarA* укажите <u>Переменная A</u> (сделайте то же и для *Поля ввода VarB), кнопки BattonCalculate* – <u>Вычислить на Ом</u>, элемента *Text\_Result\_Om* – <u>Результат на Ом</u>. Отредактируйте шрифт и выравнивание текста.

Переменная А	Вычислить на Ом	Результат на Ом
Переменная В		

Вычисление суммы двух переменных будет происходить по нажатию на Кнопку Вычислить на Ом.

3. Выделите **Кнопку Вычислить на Ом** нажатием ЛКМ → перейдите во вкладку События → нажмите на событие ButtonPressed ПКМ → Добавить обработчик → Выполнить код → Редактировать.





Структура объекта			
Имя	Описание		
✓ ➡ Form 1	Тип на осно	ве Форма	
<ul> <li>Графические объекты</li> </ul>			
ebl TextEdit VarA	Поле ввола		
abl TextEdit VarB	Поле ввода		
Nthe Button Calculate	Кнопка		
A Text Result Om	Текст		
A lext_kesuit_off	TEKCI		
События			
0			
Имя		Характеристики	Обработчик
MouseMove			
MouseClick			
MouseRightClick			
MouseDoubleClick			
MousePress			
MouseEnter			
Mousel eave			
TouchMove			
TouchPress			
TouchRelease			
FocusChanged			
ButtonPressed			
До	бавить обработ	гчик	
The Bo	гавить	Ctrl+V	

Откроется окно, в котором можно редактировать код обработчика.

4. В коде обработчика ввести:

🔀 Исхо	дный код		×
1	<pre>Text_Result_Om.Text = String.ToString(String.ToUint4(TextEdit_VarA,</pre>	0)	+
	<pre>String.ToUint4(TextEdit_VarB, 0));</pre>		
150% ~			
🕘 0 Ош	ибок 🚵 0 Предупреждений 🕕 0 Сообщений		
Опис	вние Положение		
<u>Я</u> зык: Оп	OK	Отме	на

В коде обработчика можно писать скрипты как на языке SePlatform.Om, так и на JavaScript. В данном примере используется SePlatform.Om. Так как все свойства, с которыми происходит работа, имеют свой тип данных, в скрипте необходимо использовать функции, которые явно приводят строковые типы текстовых свойств к целочисленным для выполнения операции сложения. Если оставить данную операцию без преобразования типов, то в результате получится просто объединение (конкатенация) двух строк.

На языке JavaScript данный код может иметь вид: Result.Text = +Var\_A.Text + +Var\_B.Text;

5. Нажмите ОК, сохраните и запустите проект.

После запуска можно увидеть, что при нажатии на *Кнопку Вычислить на Ом* результат отображает сумму двух переменных.

🔀 SandBox		_		×
10	Вычислить на Ом		30	
20				





В разработке проектов предпочтительнее использовать язык SePlatform.Om так как он компилируемый (т.е. перед запуском проекта, обработчики будут проверяться на ошибки) и имеет строгую типизацию (что экономит память, улучшает быстродействие, и позволяет избежать ошибок). JavaScript

интерпретируемый (перед запуском нет проверки на ошибки, что повышает риск появления ошибок в режиме исполнения), имеет динамическую типизацию (что также способствует возникновению ошибок).

# Добавление функций

Попробуем написать самописную функцию, которая будет выполнять вычисления за нас, где не нужно выполнять переводы в форматы, достаточно вызвать метод, который будет всё вычислять самостоятельно.

1. Перетяните на экранную форму из Библиотеки компонентов *Кнопку* и элемент *Текст*. Назовите их Button\_Calculate\_Func и Text\_Result\_Func.

2. В свойстве Текст *кнопки Button\_Calculate\_Func* введите <u>Вычислить функцией</u>, *Text\_Result\_Func* – <u>Результат функцией</u>. Настройте шрифт и выравнивание текста.

переменная	А Вычислить на Ом	Результат на Ом
Переменная	В	
	Вычислить функцией	Результат функцией

3. Для того, чтобы добавить функцию в Структуре объекта нажмите ПКМ по форме Form\_1 → Добавить функцию.

Структура объекта			8
			AL
Имя		Описание	
✓ ➡ Form 1			
<ul> <li>Графические объекты</li> </ul>		Переименовать	F2
TextEdit_VarA		Перейти к типу	F12
■ TextEdit_VarB		Создать	Ctrl+M
Button_Calculate		Добавить функцию	
A Text_Result_Om Button_Calculate_Fu		Скопировать полный путь	
A Text_Result_Func	÷	Вырезать	Ctrl+X
		Копировать	Ctrl+C
	Ē.	Вставить	Ctrl+V
		Удалить	Del

4. В Структуре объекта переименуйте функцию – Summ (функция, которая вычисляет сумму).

5. В Структуре объекта щелкните дважды по функции. Откроется редактор функции.

6. В редакторе функции добавьте 2 аргумента А и В, установите им **Тип** аргумента string. В поле Действие выберите **Выполнить код**. В поле **Тип** возвращаемого значения выберите string. Нажмите Редактировать.

-					
ип возвраща	емого зна	ачения:		5 string	
Имя аргум	ента	Ссылка		Тип аргумента	
А			S string		
В			Sstring		
цействие :	Выполн	ить код			
					Редактирова





7. В открывшемся окне введите код:

<i>∫</i> ∗ Исхо	дный код			×
1	return	<pre>String.ToString(String.ToUint4(A, 0) + String.ToUint4(B,</pre>	0));	
150% ~				
🕘 0 Оши	бок 🔬 0	Предупреждений () 0 Сообщений		
Описа	ние	Положение		
Язык: Om	-	OK	Отм	ена

#### 8. Нажмите ОК.

Теперь нужно где-то вызвать эту функцию. Функция будет вызвана при нажатии на *Кнопку Вычислить функцией*.

9. Выделите *Кнопку Вычислить функцией*, перейдите к её событиям. ПКМ нажмите на ButtonPressed → Выполнить код → Редактировать.

10.В открывшемся окне введите код:

🕎 Исходный код	×
<pre>1 Text_Result_Func = Summ(TextEdit_VarA, TextEdit_VarB);</pre>	
150% ~	
О Ошибок	
Описание Положение	
	ОК Отмена

11. Нажмите ОК, сохраните и запустите проект. Теперь вычисления производятся и при помощи созданной функции.



### Добавление внешних переменных

Чтобы каждое изменение свойства объекта приводило к возникновению события, применяются уведомляющие поля. Уведомляющие поля удобны, когда требуется оперативная реакция на любое изменение важного параметра.

1. Для того, чтобы создать **Уведомляющее поле**, нажмите ПКМ на **Form\_1** в структуре объекта → Создать. В поле Тип элемента выберите Уведомляющее поле. В поле Базовый тип элемента выберите Уведомляющий uint4. Нажмите ОК.





Новый элемент	Создание нового элемента					
Значения атрибутов	Тип элемента Уведомляющее поле					
	Базовый тип элемента					
	В Уведомляющий bool					
	🔞 Уведомляющий float4					
	🔞 Уведомляющий float8					
	🚺 Уведомляющий int1					
	12 Уведомляющий int2					
	🔯 Уведомляющий int4					
	10 Уведомляющий int8					
	9 Уведомляющий string					
	🕕 Уведомляющий timestamp					
	🔟 Уведомляющий uint1					
	😳 Уведомляющий uint2					
	🥶 Уведомляющий uint4					
	u Уведомляющий uint8					
	😡 Уведомляющий variant					

**Уведомляющее поле** выбранного типа добавится в Структуру объекта (группа Данные) и в список свойств объекта.

- 2. В Структуре объекта назовите это **Уведомляющее поле** Variable\_A.
- 3. Создайте ещё одно **Уведомляющее поле** с именем Variable\_В.

Эти **Уведомляющее поля** были добавлены через форму Form\_1, значит в свойствах Form\_1 появились новые пункты: Variable\_A и Variable\_B. И в их Вычисляемом значении можно указать, что именно поместить в это свойство.

Структура объекта				
Имя		Описан	ие	
✓		Тип на с	основе	Форма
> Графические об	ъекты			
> Данные				
> Функции				
Редактор свойств				
۵				
Свойство			Харак	Значение
✓ 🥶 Variable_A			RW ۶	<не определено>
Вычисляемое зн	ачение			<не определено>
> 🚾 Variable_B			RW 9	<не определено>
Редактор свойств С	обытия			

4. Поместим в Вычисляемое значение этих **Уведомляющих полей** код для того, чтобы эти 2 переменные вычислялись автоматически:





Структура объекта			é
			[AL]
Имя		Описание	
✓		Тип на основе	Форма
> Графические о	бъекты		
> Данные			
> Функции			
Редактор свойств			ť
۵,			
Свойство		Харак	Значение
∨ 🥶 Variable_A		RW ۶	<не определено>
Вычисляемое з	начение	~	(f) String.ToUint4(TextEdit_VarA, 0)
🗸 🥶 Variable_B		RW ۶	<не определено>
Вычисляемое з	начение	~	(f) String.ToUint4(TextEdit_VarB, 0)
Радактор свойств	События		

5. Добавьте на экранную форму из Библиотеки компонентов Визуальный элемент *Текст*, назовите ero в Структуре объекта Text\_Result\_Self\_Calc, в свойстве Текст введите <u>Результат самостоятельный</u>. Настройте шрифт и выравнивание текста.

Переменная / Переменная	а Вычислить на Ом В	Результат на Ом
	Вычислить функцией	Результат функцией
		Результат самостоятельный

6. Выделите элемент *Text\_Result\_Self\_Calc*, перейдите к его вычисляемому значению свойства Текст и введите код:

Структура объекта		
Имя		Описание
✓		Тип на основе Форма
<ul> <li>Графические объекты</li> </ul>		
■ TextEdit_VarA		Поле ввода
I TextEdit_VarB		Поле ввода
Button_Calculate		Кнопка
A Text_Result_Om		Текст
Button_Calculate_Fit	unc	Кнопка
A Text_Result_Func		Текст
A Text_Result_Self_Cal	c	Текст
🗸 Данные		
🖽 Variable_A		Уведомляющий uint4
🚥 Variable_B		Уведомляющий uint4
✓ Функции		
<i>f</i> ≭ Summ		Функция
Редактор свойств		
Q		
Свойство	Харак	Значение
✓ ⑤ Текст	R W ⊙	Результат самостоятельный
Начальное значение		<не определено>
Вычисляемое значение	~	(f) String.ToString(Variable_A + Variable_B
Редактор свойств События		

7. Сохраните проект, запустите. Теперь в *Текстовом поле Результат самостоятельный* вычисление производится автоматически.



🔀 SandBox		_		×
45	Вычислить на Ом	Резуль	тат на	Ом
18				
	Вычислить функцией	Результа	ат функ	цией
			63	

### Работа с элементами АР

**Элементы АР** – набор компонентов для взаимодействия с сигналами определенного типа по протоколу TCP. Компонент не визуальный (не отображается на форме) и виден только в области Структуры объекта.

# Добавление элементов АР определённого вида

В проекте *DevStudio* внутри **SandBox** есть **Calculator**, где он также может выполнять вычисления. Значит, можно сделать так, чтобы вычисления производил не *SePlatform.HMI*, а *SePlatform.Server*, то есть будем записывать данные в сервер, а потом эти данные с сервера считывать как раз через *SePlatform.HMI*.

1. Добавьте на экранную форму Form\_1 ещё одну *Кнопку* (с именем Button\_Calculate\_Server и надписью внутри Вычислить сервером) и элемент *Текст* (с именем Text\_Result\_Server и надписью внутри Результат сервером). Настройте шрифт и выравнивание текста.



Для работы с сервером следует использовать **Элементы AP**. Для того, чтобы подключиться к серверу к определенному сигналу, нужно добавить **Элемент AP** соответствующего типа. В проекте в *DevStudio* параметры **Var\_A**, **Var\_B** и **Result** внутри **Calculator** типа Uint4, соответственно и в *SePlatform.HMI* нужно добавлять **Элементы AP** типа uint4 из библиотеки компонентов.

Calculator + X	
🤌 <u>Demo.Domain.astra.AstraServer.AstraApplication.SandBox, Calculator</u> Логический объект 🖷 Demo.omx 🔹	Handler_Summa
x x x x x x x x x x	Result_from_Handler 🏾 🏵 🚯

2. Перетяните на экранную форму **Form\_1** Элемент АР uint4 из Библиотеки компонентов (раздел AP). В структуре объекта назовите его Ap\_Var\_A.



Библиотека компонентов 🗗 🗙	Form_1* 🔟			Структура объекта	
Фильтр			^		
✓ Общие элементы ^	Переменная А	Вычислить на Ом	Результат на Ом	Имя У Графические объекты	Описание
📑 Браузер источника АР 🗐 Запрос алармов	Переменная В			TextEdit_VarA	Поле ввода Поле ввода
<ul> <li>Запрос значений множества сигналов АР</li> <li>Запрос значений элемента АР</li> <li>Ипрос значений элемента АР</li> </ul>	 В	ычислить функцией	Результат функцией	Button_Calculate A Text_Result_Om Button_Calculate_Func	Кнопка Текст Кнопка
두 Источник АР 북 Элемент AP uint1 북 Элемент AP uint2	 *		Результат самостоятельный	A Text_Result_Func A Text_Result_Self_Calc A Text_Result_Senger	Текст Текст Текст
🐇 Элемент AP uint4 🖌 🐇 Элемент AP uint8	В	ычислить сервером	Результат сервером	M Button_Result_Server ✓ Данные	Кнопка
				Variable_A	Уведомляющий uint4 Уведомляющий uint4
				불 Ap_Var_A	Элемент AP uint4

Это не визуальный компонент, отображается только в данных Структуры объекта.

Теперь этому **Элементу АР** нужно показать, как правильно подключаться к серверу. То есть ему нужно указать IP-адрес машины, на которой крутится сервер (в нашем случае – машина с OC Linux) и то, по какому пути ему нужно будет подключиться.

3. Выделите в Структуре объекта *Элемент Ар\_Var\_А* и щелкните ПКМ по его свойству Источник → Указать объект.

Структура объекта				
Имя		Описа	ние	
Графические объекты				
🛩 Данные				
🥶 Variable_A		Уведол	иляющий uint4	
🥶 Variable_B		Уведок	иляющий uint4	
👙 Ap_Var_A		Элеме	нт AP uint4	
Редактор свойств				
٩				
Caeŭerre	Vanar	2		
Своиство	ларак	эначен	ие	
Отображаемое имя		Ap_Var_	А	
Кардинальное число		1		
🖏 Источник	RW→	< He on		
> (S) Путь	RW	< H	Сослаться	
> 🥶 Значение	R W 🗇	< F	Указать объект	
		-	Вырезать	Ctrl+X
		Da	Копировать	Ctrl+C
		1		<b>C</b> 1 1
		1.1	RCTORNTL	$(\pm r) \pm V$

4. В свойстве Источник есть пункт Хост. Здесь необходимо указать IP-адрес машины, на которой крутится сервер.





Структура объекта	
Имя	Описание
<ul> <li>Графические объекты</li> <li>Данные</li> </ul>	
🥶 Variable_A	Уведомляющий uint4
\cdots Variable_B	Уведомляющий uint4
Ap_Var_A	Элемент AP uint4
Редактор свойств	
۵,	
Свойство	Харак Значение
医 Отображаемое имя	Ap_Var_A
🔞 Кардинальное число	1
🜱 🖏 Источник	R ₩ → 🎆 Источник АР
🖏 Родительский источник	R ₩ → <не определено
🖳 Источник учетных данны	их <b>R W →</b> <не определено
🥶 Таймаут операций с БД	R ⊆ <не определено:
國 Таймаут установления со	единения R ⊆ 🛛 <не определено
🥶 Максимальный размер п	ула буфер R ⊆ 🛛 <не определено
> (S) Хост	R W 10.0.2.14
> 🥶 Порт	R W 4388
> 🥶 Порт истории	R W 4950
> 😮 Путь	RW
Auguru	D W Aslas

5. В свойстве Источник есть пункт Порт. Здесь должен быть тот же самый номер порта, что и у *Модуля TCP Server* внутри *AsrtaServer* в проекте *DevStudio*.

∕ AstraServer ± ×			🚰 Свойства	
AtraServer All	pha.Server 🛋 🛛 Demo.omx	* 👬	TcpServer TCP Server	
	К	N	🔿 Параметры модуля	
- Sta ModbusTepMaster =			Номер ТСР порта	4388
Опросчик Modbus TCP	Модуль истории	OPC AE Cepsep	Анонимные клиенты могут изменять значения сигналов	Да
<b>T</b>	Un Historian.Database	UaServer	Очередь данных	500000
		ОРС ИА Сервер	Очередь сообщений	100000
			Таймаут очистки застоявшихся читателей, мин.	1
Опросчик МЭК 60870-5-104	🗟 AstraApplication 🖷 🌟	( TcpServer	Активность	Дa
÷	🥥 SandBox	ICP Server	Отображаемое имя	12
	🔗 root			-
Опросчик МЭК 60870-5-104 Ф	SandBox     Foot	TCP Server	Активность Отображаемое имя	Да





Структура объекта			
Имя	Описан	ие	
<ul> <li>Графические объекты</li> </ul>			
∨ Данные			
Wariable_A	Уведом.	ляющи	й uint4
Variable_B	Уведом.	ляющи	й uint4
🚆 Ap_Var_A	Элемен	т AP uir	nt4
Редактор свойств			
۹			
Свойство		Харак	Значение
Отображаемое имя			Ap_Var_A
🔞 Кардинальное число			1
✓ ₩ Источник		RW→	關 Источник АР
🖏 Родительский источник		R W →	<не определено>
🖳 Источник учетных данных		RW→	<не определено>
🥶 Таймаут операций с БД		R⊆	<не определено>
🥶 Таймаут установления соедин	нения	R⊆	<не определено>
🥶 Максимальный размер пула	буфер	R⊆	<не определено>
> 🔕 Хост		RW	10.0.2.14
> 🚾 Порт		RW	4388
> 🚾 Порт истории		RW	4950
> 🖲 Путь		RW	
Редактор свойств События			

- 6. В свойстве Активность укажите <u>True</u>.
- 7. Сверните свойство Источник.

В свойстве Путь необходимо указать путь, по которому этот элемент сможет достучаться до конкретного сигнала в сервере. Тег сигнала можно найти в *OpcExplorer*.

8. Откройте *OpcExplorer*, в Обозревателе найдите *Параметр Var\_A*, щелкните ЛКМ. Откроется окно, в котором прописан путь до этого сигнала, скопируйте его и вставьте в свойство Путь элемента *Ap\_Var\_A*.

		Структура объекта	
Coospesarene         ◆ ↓ >           Douck         P →           Image: Construction of the server         P ⊕         Server           P ⊕         Server         P ⊕         ObjectTypes           P ⊕         ObjectTypes         P ⊕         Server           P ⊕         Server         P ⊕         Server	Инспектор 1 SandBox.Calculator.Var_A 社 × SandBox.Calculator.Var_A в Astra UA на 10.0.2.14 Переменная А 10 Следить за значением Значение Качество Время	Имя У Графические объекты Данные Variable_A Variable_B <u>Ap_Var_A</u> Редактор свойств	Описание Уведомляющий uint4 Уведомляющий uint4 Элемент AP uint4
<ul> <li>Galc_3</li> <li>Calc_from_Link</li> <li>Calculator</li> <li>Calculator</li> <li>Casult</li> <li>Result_from_Handler</li> <li>Calculator</li> <li>Var_A</li> <li>Var_B</li> </ul>		Свойство © Отображаемое имя © Кардинальное число > 🛄 Источник > © Путь > © Значение	Харак Значение Ap_Var_A 1 R W → III Источник AP R W ✓ SandBox.Calculator.Var_A R W ⊙ <не определено>

9. Проделайте действия 2-8 для Var\_B и Result.





Описание Данные III Variable_A Уведомляющий uint4 III Variable_B Уведомляющий uint4	Структура объекта		
Ap_var_o Элемент AP uinte AP	Имя ✓ Данные Фанаble_A Фанаble_B – #Ap_Var_A – #Ap_Var_B – #Ap_Var_B	Опис Уведо Уведо Элем Элем Элем	ание мляющий uint4 мляющий uint4 ент AP uint4 ент AP uint4 ент AP uint4
	Редактор свойств		
р свойств	Свойство	Харак	Значение
рр свойств Свойство Харак Значение Во Харак Значение Свойство Харак Значение Свойство	Отображаемое имя		Ap_Var_B
рр свойств  Свойство Харак Значение Отображаемое имя Ар_Var_B Свойство Харак Значение Свойство	<ul> <li>Кардинальное число</li> <li>Мсточник</li> </ul>	RW→	🖬 Источник АР
рр свойств  Свойство  Харак Значение Отображаемое имя Ар_Var_B Кардинальное число 1 Источник R W → II Источник AP  Кардинальное число 1	> 🖲 Путь	RW~	SandBox.Calculator.Var_B

Результат вычисления будем сразу выводить в *текстовое поле Результат сервером*.

10.Выделите в Структуре объекта *Text\_Result\_Server*, перейдите к его Вычисляемому значению свойства Текст. Введите код:

Структур	а объекта					
Имя				Описание		
	A Text_F	Result_Om		Текст		
	btn Butto	n_Calculate_Fun	c	Кнопка		
	A Text_F	Result_Func		Текст		
	A Text_F	Result_Self_Calc		Текст		
	A Text_F	Result_Server		Текст		
	btn Butto	n_Result_Server		Кнопка		
Редактор	свойств					
Q						
Свойство	)		Харан	Значение		
✓	екст		R ₩ <	🔉 Результат сервером		
H	ачальное зн	начение		<не определено>		
Be	ычисляемо	е значение	~	(f) Ap_Result.ValueAsString		
Редакто	р свойств	События				

Значение в сервер мы будем записывать при нажатии на Кнопку Вычислить сервером.

11.В Структуре объекта выделите *Кнопку Button\_Calculate\_Server*, перейдите во вкладку События, щелкните ПКМ по ButtonPressed → Добавить обработчик → Выполнить код → Редактировать. Введите код:

🔀 Исхо	дный код							×
1	Ap_Va	ir_A	= Strin	g.ToUint4	(TextEd:	it_VarA,	0);	
2	Ap_Va	ir_B	= Strin	ng.ToUint4	(TextEd:	it_VarB,	0);	
150% ~								
0 0 Out	ибок 4	🛛 0 Пред	упреждений	і) 0 Сообщений	i			
Описа	ние			Γ	Іоложение			

12.Нажмите ОК, сохраните проект, запустите. Проверьте соответствие значений сигналов *Var\_A*, *Var\_B* и *Result* в Инспекторе *OpcExplorer*.





🙀 SandBox	_		×					
52 Вычислить на Ом 132	Результ	ат на С	Эм	🔍 Ин Файл 🔾	аспектор 1 - ОрсЕхріо Вид Проект Сп	ргег правка 10.0.2.14	• 🕑 Astra_UA	
Вычислить функцией	Результат	г функі	цией	Тип	<b>Інспектор 1</b> + × Сигнал	Значение	Описание	Источник
	1	84		14 14	SandBox.Calculator. SandBox.Calculator.	<u>Var A</u> 52 <u>Var B</u> 132	Переменная А Переменная В	<u>Astra_UA</u> <u>Astra_UA</u>
Вычислить сервером	1	84		•	SandBox.Calculator.	<u>Result</u> 184	Результат	<u>Astra_UA</u>

### Добавление источников данных

*Источник АР* – компонент для связи с источником данных по протоколу TCP. Компонент невизуальный (не отображается на форме) и виден только в области Структура объекта.

В случае, если придется изменить параметры подписки на сигналы, например, порт или хост, то придется вносить изменения в каждый **Элемент АР**. Это неудобно, для унификации есть возможность создать **Источник АР**, где будут указаны все необходимые свойства для подключения.

1. Перетяните на экранную форму из Библиотеки компонентов *Источник АР* (раздел АР). В Структуре объекта переименуйте его в ApSource\_Main.

Библиотека компонентов 🛛 🗗 🗙	Form_1* 🗵	Структура объекта	
Фильтр	^		
Общие элементы	Переменная А	Имя	Описание
✓ ♣ AP	вычистить на ом результат на ом	A Text_Result_Om	Текст
📑 Браузер источника AP	Переменная В	Button_Calculate_Func	Кнопка
Запрос алармов		A Text_Result_Func	Текст
💣 Запрос значений множества сигналов АР	Вычислить функцией Результат функцией	✓ Данные	
💼 Запрос значений элемента АР		Wariable_A	Уведомляющий uint4
💭 Источник AP 🛛 🛶 📃		U Variable_B	Уведомляющий uint4
Очередь активных алармов	Результат	Ap_Var_A	Элемент AP uint4
> A OPC	Самостоятельный	Ap_Var_B	Элемент AP uint4
A 501		👙 Ap_Result	Элемент AP uint4
A Consults		ApSource_Main	Источник АР
> Вазовые	вычислить сервером Результат сервером	✓ Функции	

2. Выделите *ApSource\_Main* в Структуре объект, в свойстве Хост введите IP-адрес машина, на которой находится сервер, в свойстве Порт должен быть тот же самый номер порта, что и у *Модуля TCP Server* внутри *AsrtaServer* в проекте *DevStudio*, Активность – <u>true</u>.

Структур	оа объекта				
Имя			Описан	ие	
~ <u>7</u>	анные				
	🥶 Variabl	e_A	Уведом.	ляющи	ій uint4
	🥶 Variabl	e_B	Уведом.	ляющи	ій uint4
	🚆 Ap_Var	_A	Элемен	т AP uiı	nt4
	🚆 Ap_Var	_B	Элемен	т AP uiı	nt4
	🚆 Ap_Re	sult	Элемен	т AP uiı	nt4
	💭 ApSou	rce_Main	Источн	ик АР	
~ ¢	ункции				
Свойств	0			Харак	Значени
> S>	бост			RW	10.0.2.14
> 🚾 r	Тарт			RW	4388
> 😐 r	Торт истории	1		RW	4950
> 5 Г	Туть			RW	
> 🖪 4	Активность			RW	true
> 🐽 r	Териод перег	юдписки		RW	0
> 5	1мя клиента			RW	HMI
				RW	HMI
> (5)	пдентификат	ор клиента			





Теперь элементы *Ap\_Var\_A*, *Ap\_Var\_B* и *Ap\_Result* должны получать данные о хосте и порте от этого источника.

3. В Структуре объекта выделите *Ар\_Var\_А*, нажмите ПКМ по свойству Источник → Сброс значения.

4. В структуре объекта выделите **Ар\_Var\_А**, нажмите ПКМ по свойству Источник → Сослаться → **ApSource\_Main**.

Структура	объекта	
Имя		Описание
∨ Да	нные	
	🥶 Variable_A	Уведомляющий uint4
	🥶 Variable_B	Уведомляющий uint4
	👙 Ap_Var_A	Элемент AP uint4
	👙 Ap_Var_B	Элемент AP uint4
	🐥 Ap_Result	Элемент AP uint4
	🐺 ApSource_Main	Источник АР
	нкции	
Редактор	свойств	
Q		
Свойство		Харак Значение
S Or	гображаемое имя	Ap_Var_A
🔞 Ka	рдинальное число	1
ШИ	точник	R W → III. ApSource_Main_
> S П	/ть	R W - SandBox.Calculator.Var_A
> 🥶 Зн	ачение	R W 😳 <не определено>
Редактор	свойств События	_

- 5. Повторите действия 3, 4 для *Ар\_Var\_В* и *Ар\_Result*.
- 6. Сохраните проект, запустите. Проверьте соответствие значений сигналов *Var\_A*, *Var\_B* и *Result* в Инспекторе *OpcExplorer*.

# Использование глобальных объектов. Каскадирование источников

В проекте может быть использовано много экранных форм и различных объектов, использующих общие ресурсы. Например, подключение к ОРС серверу. Для того, чтобы на каждую экранную форму не добавлять Источники АР, где прописан полный путь, и в случае, если необходимо внести изменения, не редактировать их в каждой форме по-отдельности, можно использовать Глобальные объекты.

1. В библиотеке компонентов во вкладке Проект щёлкните ПКМ на Глобальные объекты → Создать. Откроется Мастер создания элементов. В поле Тип элемента выберите Глобальный объект, Базовый тип элемента - Не задан. Назовите его Global.

F. 6-			Новый элемент	Созлание нового элемента
фильтр	нентов	• •	Значения атрибутов	Гип элемента Глобальный объект 🔻
>	Общие элементы			Базовый тип элемента
✓ Глобальные с	Проект <demohmi> объекты</demohmi>	]		> 💩 AP
Типы У Экранные	Сортировка 🕨			> & SQL
For	Создать Ctrl+T			> 🐣 Security
Ma	Новая папка			> 🖓 Визуальные
				> 🚳 Встроенные > 🖓 Графики
				> 🛞 Окружение
				Не задан
				ОК Отмена

- 2. Откройте глобальный объект Global двойным нажатием мыши.
- 3. Перейдите в форму Form\_1, вырежьте отсюда источник ApSource\_Main и вставьте его в Global.



Структура объекта	
Имя	Описание
✓ {♣} Global	<базовый тип не задан>
∨ Данные	
ApSource_Main	Источник АР

Теперь на этот источник можно ссылаться из любого места в проекте при помощи специального слова **unit**.

Так как у элементов *Ap\_Var\_A*, *Ap\_Var\_B* и *Ap\_Result* есть одинаковая часть пути (SandBox.Calculator), можно эту часть вложить в ещё один *Источник AP*.

4. Перейдите в форму Form\_1, перетяните из Библиотеки компонентов *Источник AP*, назовите его ApSource\_Calc.

5. Выделите в Структуре объекта *ApSource\_Calc*, щёлкните ПКМ по свойству Родительский источник → Сослаться → unit.Global.ApSource\_Main. В свойстве Активность укажите <u>true</u>, а в свойстве Путь укажите одинаковую часть пути: SandBox.Calculator.

Структура объекта	
Имя	Описание
✓ Данные	
Wariable_A	Уведомляющий uint4
Wariable_B	Уведомляющий uint4
👙 Ap_Var_A	Элемент AP uint4
🐇 Ap_Var_B	Элемент AP uint4
🐇 Ap_Result	Элемент AP uint4
ApSource_Calc	Источник АР
✓ Функции	
Редактор свойств	
Q	
Свойство	Харак Значение
Отображаемое имя	ApSource_Calc
🔞 Кардинальное число	1
🖏 Родительский источник	R W → Щunit.Global.ApSource_Mair
> 🖲 Путь	R W SandBox.Calculator
> В Активность	R W true
Редактор свойств События	

6. В структуре объекта выделите элемент *Ар\_Var\_А*, щёлкните ПКМ по свойству Источник → Сослаться → ApSource\_Calc. В свойстве Путь оставьте только индивидуальную часть пути: Var\_A (то есть Вы каскадируете источники, собираете путь по кусочкам).





Структура объекта			
Имя		0	писание
✓ Данные			
🥶 Variab	le_A	Ув	едомляющий uint4
🥶 Variab	le_B	Ув	едомляющий uint4
👙 Ap_Va	r_A	Эл	емент AP uint4
🐇 Ap_Va	r_B	Эл	емент AP uint4
🐇 Ap_Re	esult	Эл	емент AP uint4
🐺 ApSo	urce_Calc	И	точник АР
Функции			
Редактор свойств			
٩			
Свойство		Харак	Значение
Отображаем	ое имя		Ap_Var_A
🔢 Кардинально	е число		1
🛄 Источник		RW→	ApSource_Calc
> 医 Путь		R W ~	Var_A
> 🥶 Значение		R W 🛇	<не определено>
Редактор свойств	События		

7. Проделайте действия из пункта 6 с элементами *Ар\_Var\_В* и *Ар\_Result*.

Структура объекта	
Имя	Описание
✓ Данные	
🥶 Variable_A	Уведомляющий uint4
🥶 Variable_B	Уведомляющий uint4
👙 Ap_Var_A	Элемент AP uint4
👙 Ap_Var_B	Элемент AP uint4
🐇 Ap_Result	Элемент AP uint4
ApSource_Calc	Источник АР
✓ Функции	
Редактор свойств	
Q	
Свойство	Харак Значение
🜖 Отображаемое имя	Ap_Var_B
🔞 Кардинальное число	1
🛄 Источник	R W → 🛄 ApSource_Calc
> 🜖 Путь	R W 🗸 Var_B
> 🥶 Значение	R W 😳 <не определено>
Редактор свойств События	

8. Сохраните проект, запустите. Проверьте соответствие значений сигналов *Var\_A*, *Var\_B* и *Result* в Инспекторе *OpcExplorer*.

В результате сервер также будет производить вычисления. Но для элементов OPC, которые используются в проекте, используется источник, указанный в глобальных объектах. Таким образом, можно облегчить работу с большим количеством экранных форм и объектов в проекте. При внесении изменений не будет необходимости редактировать каждый источник, достаточно поменять свойства в глобальном объекте.

# Типизация

В проекте может быть много типовых элементов, например датчиков, которые могут быть использованы на различных экранных формах. Для того, чтобы постоянно не рисовать датчик на мнемосхеме и не добавлять логику в каждый из них, можно один раз создать типовой датчик, затем тиражировать его экземпляры на мнемосхемах. Если необходимо будет внести изменения, то нужно будет всего лишь отредактировать тип датчика, и все экземпляры унаследуют эти изменениями. Это позволяет избежать ошибок и упрощает проект.





1. В библиотеке компонентов во вкладке Проект щелкните ПКМ на Типы → Создать. Откроется Мастер создания элементов. В поле Тип элемента выберите Тип, в поле Базовый тип выберите визуальный компонент Прямоугольник.

Новый элемент	Созлание нового элемента
Значения атрибутов	Тип элемента Тип
,,	Базовый тип элемента
	> 🛞 Security
	🗸 🛞 Визуальные
	Ползунок
	Прямоугольник
	🛅 Радиальный градиент
	ок Системная кнопка

Графические типы можно создавать на основе любого графического элемента.

2. Перейдите внутрь созданного типа (Rectangle\_1) двойным нажатием мыши.

3. Перетяните на тип **Rectangle\_1** из библиотеки компонентов элемент **Текст**.

4. Перейдите на форму Form\_1 и перетяните из Библиотеки компонентов 3 экземпляра типа Rectangle\_1.

	Rectangle_1* Form_1	* 🗙	
Библиотека компонентов	-		
Фильтр	Переменная А	Вычислить на Ом	Результат на Ом
Проект <demohmi></demohmi>			
✓ Глобальные объекты {	переменная в	<u>5</u>	
✓ Типы		Вычислить функцией	Результат функцией
			Результат самостоятельный
		Вычислить сервером	Результат сервером
	Text	Text	Text

- 5. Перейдите в тип **Rectangle\_1** и перетяните элемент **Текст** в нижнюю часть прямоугольника.
- 6. Перейдите на форму Form\_1. Все экземпляры типа приняли эти изменения.

Переменная / Переменная	А Вычислить на Ом В	Результат на Ом
	Вычислить функцией	Результат функцией
		Результат самостоятельный
	Вычислить сервером	Результат сервером
Text	Text	Text





Изменения внутри типа влечёт за собой изменения у всех экземпляров. Но каждый экземпляр можно сделать уникальным. Можно переопределить свойство родительского элемента, на основе которого построен логический тип. Из этого следует следующая особенность типизации — возможность указывать какие-то уникальные свойства. Например, можно сделать так, чтобы у каждого экземпляра в текстовом поле размещалось своё уникальное название.

7. Перейдите к типу **Rectangle\_1**. Создайте здесь **Уведомляющее поле**: щелкните ПКМ по **Rectangle\_1** в Структуре объекта → Создать. Тип элемента – Уведомляющее поле, Базовый тип элемента Уведомляющий string. Назовите его name\_type.

Структура объекта	
Marg	Описание
✓	Описание Тип на основе Прямоугольник
<ul> <li>Графические объекты</li> </ul>	
A Text_1	Текст
S name_type	Уведомляющий string

8. Выделите элемент *Text\_1* и в Вычисляемое значение его свойства Текст поместите уведомляющее поле <u>name\_type</u>.

Структура объекта		
Имя		Описание
✓		Тип на основе Прямоугольник
Графические объ	екты	
A Text_1		Текст
🗡 Данные		
s name_type	:	Уведомляющий string
Редактор свойств		
Q		
Свойство	Харак	Значение
✓ S Текст	R W 🤅	Text
Начальное значе	ние	<не определено>
Вычисляемое зна	чение 🗸	(f) name_type
Редактор свойств Со	бытия	

9. Перейдите на форму **Form\_1**, в структуре элемента выделите один из типов **Rectangle\_1**. В его свойстве name\_type введите <u>1</u>. То же сделайте и для оставшихся экземпляров.

Структура объекта	
Имя	Описание
✓	Тип на основе Форма
<ul> <li>Графические объекты</li> </ul>	
Rectangle_1_1	Rectangle_1
Rectangle_1_2	Rectangle_1
Rectangle_1_3	Rectangle_1
Редактор свойств	
Q.	
Свойство	Харак Значение
> S name_type	R W 🗲 1

10.Сохраните проект, запустите.





🔀 SandBox			_			×
Переменная и Переменная	А Вычи	слить на Ом	Резуль	тат н	а Ом	
	Вычислит	ь функцией	Результа	т фу	нкцией	
				0		
	Вычислит	ь сервером	:	184		
1		2			3	

Теперь типовые элементы имеют своё индивидуальное имя. Это свойство поможем в наименовании всех датчиков и задвижек, которые будут располагаться на мнемосхеме в демонстрационном проекте.

# Создание демонстрационного проекта в SePlatform.HMI

В демонстрационном проекте будут и датчики, и задвижки. Все экземпляры будут располагаться на форме MainForm.

- 1. Откройте форму MainForm двойным нажатием мыши.
- 2. В Структуре объекта выделите *MainForm* и в свойствах Ширина и Высота укажите 700.
- 3. В свойстве Размеры окна укажите Автоподбор, Положение окна По центру монитора.
- 4. В свойстве Заголовок окна введите Демо-проект.

5. В библиотеке компонентов во вкладке Проект щелкните ПКМ на форме MainForm → Установить форму главной.

6. Сохраните проект, запустите.



Перейдём к созданию типов датчика и задвижки.

#### Создание типа с датчиком

Сейчас наша задача вывести на главную форму экземпляры датчиков и их значения. Датчики будут располагаться внутри прямоугольника. Поэтому создадим новый тип на основе прямоугольника.

1. Создайте тип на основе Визуального компонента Прямоугольник, назовите его Sensor. Перейдите в него.

2. Перетяните из библиотеки компонентов элемент *Текст*, назовите его Text\_Pressure (в нём в дальнейшем будет отображаться значение давление).

3. В свойстве Текс этого элемента введите <u>Давление</u>. Отредактируйте шрифт и положение текста.





MainForm* 🗵	Sensor* 🙁				Структура объекта	
					Имя У П Sensor У Графические объекты А Text_Pressure	Описание Тип на основе Прямоугольник Текст
	Да	ВЛ	ни	e		

4. Перейдите к форме *MainForm* и разместите на ней 3 экземпляра Типа Sensor.

Теперь необходимо на каждый экземпляр датчика вывести значение давления, то есть нужно оформить подписку на сервер для того, чтобы он подписался на конкретные тэги датчика и вывел значение уже в RunTime. Но сначала давайте посмотрим, как должен выглядеть путь до датчика.

5. Перейдите в OpcExplorer и в Обозревателе кликните ЛКМ на сигнал Pressure внутри Sensor\_1.



Путь до конкретного датчика будем разбивать на 4 части:

root.Sensors.Sensor\_1.Pressure root.Sensors.Sensor\_2.Pressure root.Sensors.Sensor\_3.Pressure

- 1. Точка подключения номер хоста и номера порта. Укажем в «ApSource\_Main» (в «Global»)
- 2. Часть пути «root.Sensors». Укажем в «Ap\_source\_Sensors» (в том же «Global»)
- 3. Уникальная часть имени экземпляра. Укажем в «ApSource\_Sensor\_Type» (уже в самом типе) и добавим уведомляющее поле «\_Sensor\_Path» для индивидуальной настройки имени для каждого экземпляра «Sensor\_Type»
- 4. Элемент Ap\_Pressure + Pressure
  - 1) Точка подключения (хост и порт). Эта точка подключение уже существует, *ApSource\_Main* (*Global*).
  - 2) Следующая часть пути повторяется абсолютно у всех датчиков:



127



**root.Sensors**. Эту часть пути поместим в *Источник AP*, назовём его ApSource\_Sensors\_Global и тоже разместим его в глобальных объектах (*Global*). Его особенностью будет то, что он от предыдущего источника будет получать часть пути, в которой описаны хост и порт. И будет описывать свою часть пути: **root.Sensors**.

- 3) Следующая часть пути это название конкретного датчика: Sensor\_1, Sensor\_2, Sensor\_3. То есть как раз то, что отвечает за уникальность. Эту уникальную часть имени (SensorPath, будет храниться в уведомляющем поле) стоит создать внутри графического типа и разместить её в *ApSource\_SensorType* (в вычисляемом значении свойства Путь). Особенность этого источника в том, что он будет наследовать то, что есть у других источников (то есть то, что есть у глобального источника и будет получать часть путь root.Sensors).
- 4) После этого создадим Элемент AP Ap\_Pressure с оставшейся частью пути, которая повторяется везде. Этот элемент будет наследовать то, что было в предыдущих частях каждого конкретного датчика, но добавит оставшуюся часть пути – Pressure.

6. Откройте *SePlatform.HMI*, перейдите в глобальный объект *Global*. Перетяните сюда из Библиотеки компонентов *Источник АР*. Назовите его ApSource\_Sensors\_Global.

7. В Структуре объекта выделите *Источник ApSource\_Sensors\_Global* и в свойстве Родительский источник сошлитесь на <u>ApSource\_Main</u>. Установите ему Активность <u>true</u>. В свойстве Путь введите одинаковую часть пути: <u>root.Sensors</u>.

Структура об	ьекта			
Имя			Описание	
🗸 👫 Globa	I		<базовый тип	не задан>
∨ Данны	le			
	ApSource_Main		Источник АР	
	ApSource_Sensor_G	Global	Источник АР	
Редактор свой	іств			
Q				
Свойство		Харак	Значение	
🛄 Родит	ельский источник	R W →	ApSource_M	lain
> 😐 Порт		RW	4388	
> 🥶 Порт	истории	RW	4950	
> 医 Путь		RW	root.Sensors	
> 🖪 Актие	ность	RW	true	
Редактор св	ойств События			

8. Перейдите в *Tun Sensor*. Создайте здесь *Уведомляющее поле* типа string, назовите его SensorPath (здесь будет храниться индивидуальная часть пути Sensor 1, Sensor 2 и Sensor 3).

9. Перетяните из Библиотеки компонентов на Тип Sensor *Источник AP*, назовите его ApSource\_SensorType. В свойстве Родительский источник сошлитесь на <u>ApSource\_Sensors\_Global</u>, находящийся в глобальных объектах. В вычисляемое значение свойства Путь поместите <u>SensorPath</u>, Активность: <u>true</u>.





Структура объекта		
Имя		Описание
✓		Тип на основе Прямоугольник
<ul> <li>Графические объекты</li> </ul>		
A Text_Pressure		Текст
🗸 Данные		
SensorPath		Уведомляющий string
🐺 ApSource_Sensor_T	lype	Источник АР
Редактор свойств		
4		
Свойство	Xap	Зириение
🛄 Родительский источник	RW	→ 🐘 unit.Global.ApSource_Sensor_Global
✓ S Путь	RW	
Начальное значение		<не определено>
Вычисляемое значение	~	(f) SensorPath
> В Активность	RW	true

10.Перетяните из Библиотеки компонентов на тип Sensor **элемент Ар** float4, назовите его Ap\_Pressure. В свойстве Источник сошлитесь на <u>ApSource\_SensorType</u>, в свойстве Путь укажите оставшуюся часть – <u>Pressure</u>.

Структура объекта			
Имя		Описан	ие
🛩 🔲 Sensor		Тип на	основе Прямоугольник
Графические	объекты		
A Text_F	ressure	Текст	
✓ Данные			
Senso	rPath	Уведом	ляющий string
🐺 ApSo	urce_Sensor_Type	Источн	ик АР
🕌 Ap_Pr	essure	Элемен	τ AP float4
Редактор свойств			
٩			
Свойство		Харак	Значение
Отображаем	ое имя		Ap_Pressure
🔞 Кардинально	е число		1
🛄 Источник		R W →	ApSource_Sensor_Type
> 🕼 Путь		R W 🗸	Pressure
> 📧 Значение		R W 🗇	<не определено>
Редактор свойств	События		

11.В Структуре объекта выделите элемент *Text\_Pressure*, и в Вычисляемое значение его свойства текст поместите значение Ap\_Pressure в текстовом формате: <u>Ap\_Pressure.ValueAsString</u>.

Структура объекта	
Имя	Описание
✓	Тип на основе Прямоугольник
<ul> <li>Графические объекты</li> </ul>	
A Text_Pressure	Текст
✓ Данные	
SensorPath	Уведомляющий string
🐺 ApSource_Sensor_T	уре Источник АР
👙 Ap_Pressure	Элемент AP float4
Редактор свойств	
۹	
Свойство	Харак Значение
✓ ⑤ Текст	R W 😳 Давление
Начальное значение	<не определено>
Вычисляемое значение	<ul> <li>(f) Ap_Pressure.ValueAsStrin</li> </ul>

Первая часть создания типа датчика готова, можно переходить к редактированию экземпляров.



12.Перейдите в экранную форму **MainForm**, выделите в Структуре объекта **Графический объект Sensor\_1**. В свойстве SensorPath укажите индивидуальную часть пути: <u>Sensor\_1</u>.

Структура объекта	
Имя	Описание
✓	Тип на основе Форма
Графические объекты	
Sensor_1	Sensor
Sensor_2	Sensor
Sensor_3	Sensor
Редактор свойств	
٩	
Свойство	Харак Значение
> SensorPath	R W 9 Sensor_1
Редактор свойств События	

13.Повторите пункт 12 для Sensor\_2 и Sensor\_3.

14.Сохраните проект, запустите форму в RunTime.

F	Демо-проект			_		×
	]					7
	40	50			50	

Таким образом мы пробросили значение *Параметра Pressure* на экземпляры датчика в *Alha.HMI*. Осталось пробросить имя каждого датчика, которое хранится внутри атрибута Описание у объектов.

15.Перейдите в *Тип Sensor* и перетяните сюда из Библиотеки компонентов визуальный элемент *Текст*. Назовите его Text\_Name. В его свойстве Текс введите <u>Имя</u>. Настройте шрифт и выравнивание текста.

Для того, чтобы вывести имя, нужно каким-то образом вытащить описание у объекта, внутри которого находятся наши датчики. Но *SePlatform.HMI* не умеет подписываться на объекты. Он умеет

подписываться только на конкретные сигналы. Соответственно свойства он может вытащить только у сигналов. Мы можем сделать в сервере дополнительный тэг, который будет хранить в себе имя объекта, внутри которого он содержится. Для этого нам нужно будет немного модифицировать проект в *DeveloperStudio*.

16.Откройте *DevStudio*, перейдите в Пространство имён с типами *Types* и добавьте сюда из Панели элементов *Тип сокета*. Назовите его info.

17.Перейдите в *Сокет info* и перетяните сюда из Панели элементов *Параметр* типа string. Назовите его ObjectName.

18. Добавьте **Параметру ObjectName** атрибуты Описание и Начальное значение. В атрибуте Описание введите <u>Имя объекта</u>, в атрибуте Начальное значение -

<u>@(object:System.Attributes.Description)</u> (это обращение к объекту, внутри которого будет содержаться данный параметр и обращение по пути к атрибуту). Таким образом мы вытащили стандартный атрибут System.Attributes.Description.







Теперь необходимо разместить этот *Тип сокета* в типах датчика и задвижки со стороны Сервера ввода-вывода.

19.Перейдите в *Types* → *Sensor\_IOS* и из Панели элементов перетяните сюда *Тип сокета* info.

Поиск 🔎 -	💊 <u>Demo.Types.Sensors. Sensor IOS</u> Логический тип 🖷 🛛 Demo.omx	<ul> <li>info</li> </ul>
Панель элементов         Ф. Ф. Х.           Поиск         Ф. Ф. Х.           Ф. () Demo         Ф. Г.           Ф. () SandBox_Type         Ф. () SandBox_Type           Ф. () SandBox_Type         Ф. () Valves           Ф. () Valves         Ф. () State           Ф. () Valves         Ф. () Valves	→ ① SetImit → ② ③ SetImit State → ① Pressure → ① Turn: info	State Trun: State
<ul> <li>b {} Om.PLC</li> <li>b {} Om.Server</li> <li>b Вычисления</li> </ul>	○ SObjectName	Pressure 🤿 🖗

20.Перейдите в *Types* → *Valve*\_*IOS* и из Панели элементов перетяните сюда *Tun сокета info*.

21.Постройте решение, перейдите к развёртыванию и залейте конфигурацию на AstraServer.

Теперь необходимо подписаться на этот строковый элемент у датчика в SePlatform.HMI.

22.Откройте проект в *SePlatform.HMI*, перейдите в *Tun Sensor*. Из библиотеки компонентов перетяните сюда *Элемент AP* string, назовите его Ap\_ObjectName.

23.В структуре объекте выделите *Ар\_ОbjectName*. В свойстве Источник сошлитесь на

<u>ApSource\_SensorType</u>, а в свойстве Путь то, чего еще не хватает: <u>info.ObjectName</u> (полный тэг до сигнала можно посмотреть в *OpcExplorer*).

root Senror Senror 1 info ObjectName - O	n-Evolorer	Структура объекта	
Файл Вид Проект Справка	🔹 🚱 Astra_UA 🔹 🔹 🍇 🔹 🖳	Имя	Описание
Сбозреватель         Ф           Поиск         О           Ф         Ф           Авта ЦА на 10.0.2.14         В           В         Server           В         Server           В         NativeTypes	Инспектор 1 root.Sensors.Sensor_1.info.ObjectName № × © root.Sensors.Sensor_1.info.ObjectName в Astra UA на 10.0.2.14 Има объекта Датчик №1	<ul> <li>Sensor</li> <li>Графические объекты</li> <li>Данные</li> <li>SensorPath</li> <li>ApSource_Sensor_Type</li> <li>Ap_Pressure</li> <li>Socie with</li> </ul>	Тип на основе Прямоугольник Уведомляющий string Источник АР Элемент АР float4
<ul> <li>F proxies</li> <li>F root</li> <li>Sensors</li> </ul>	Значение Качество Время	Редактор свойств	Source of the second
<ul> <li></li></ul>		Свойство © Отображаемое имя © Кардинальное число Щ. Источник > © Путь > © Значение	Харак Значение Ap_ObjectName 1 R W → ЩApSource_Sensor_Typ R W ✓ info.ObjectName R W ☉ <не определено>

24.В структуре объекта выделите элемент *Text\_Name* и в Вычисляемое значение его свойства Текст поместите <u>Ap\_ObjectName</u>.





Имя	Описание
✓	Тип на основе Прямоугольни
Графические объекты	
A Text_Pressure	Текст
A Text_Name	Текст
✓ Данные	
Редактор свойств	
۹	
Свойство	Харак Значение
✓ S Текст	R ₩ ۞ Имя
Начальное значение	<не определено>
Вычисляемое значение	e 🗸 🗍 Ap_ObjectName

25.Сохраните проект, запустите главную форму в RunTime. Теперь все датчики подписаны и отображают значение давления.

F	Демо-проект		_		×
	Датчик №1	Датчик №2	Да	тчик N	23
	50	60		70	

Самостоятельное задание. Создайте *Тип Valve*, разместите на нём текстовое поле, в котором будет выводиться имя задвижки. Разместите 3 экземпляра типа на форме MainForm.

	Струк	тура объекта	
Имя		] Valve Графические объекты А Техt_Name Данные В ValvePath В ValvePath В ApSource_Valve_Type Д ApSource_Valve_Type Д Ap_ObjectName	Описание Тип на основе Прямоугольни Текст Уведомляющий string Источник АР Элемент AP string
💌 Демо-проект		- 0	×
Датчик №1 30	Датчик №2 40	Датчик №3 50	
Задвижка №1	Задвижка №2	Задвижка №	23





# Добавление цветовой индикации

У задвижки есть 4 состояние: Opened, Closed, Stoped, inProgress (сокет StateValve). Наша задача: нарисовать на *Tune Valve* задвижку, подписаться на эти состояния и в зависимости от состояния закрашивать задвижку.



1. В *SePlatform.HMI* перейдите в *Tun Valve*. Из Библиотеки компонентов (раздел Визуальные) перетяните элемент *Линия* и нарисуйте с её помощью задвижку. В структуре объекта назовите эту линию Valve\_Pic.



2. В структуре объекта выделите *Линию Valve\_Pic*, в свойстве Стиль заливки выберите <u>Сплошная</u> <u>заливка</u>. Задвижка окрасится в чёрный цвет.

3. Из библиотеки компонентов (раздел АР) перетяните **Элемент АР** bool, назовите его Ap\_Opened.

4. В Структуре объекта выделите *Ар\_Орепеd*, в свойстве Источник сошлитесь на

<u>ApSource\_Valve\_Type</u>, в свойстве путь укажите недостающую часть пути: <u>StateValve.Opened</u>.





Структура	объекта				
Имя			Описан	ие	
🗸 🗖 Val	lve		Тип на с	основе Прямоугольник	
🗸 Гра	афические	объекты			
∽ Да	нные				
	S Valvel	Path	Уведом.	ляющий string	
	🐺 ApSoi	urce_Valve_Type	Источник АР		
	着 Ap_O	bjectName	Элемен	τ AP string	
	🐥 Ap_O	pened	Элемен	т AP bool	
Редактор о	войств				
٩					
Свойство			Харак	Значение	
<b>S</b> От	ображаем	ое имя		Ap_Opened	
18 Ka	рдинально	е число		1	
🛄 Ис	точник		R W →	ApSource_Valve_Type	
: 🖲 Пу	ть		R ₩ ~	StateValve.Opened	
> 🗊 Зн	ачение		R W ♡	<не определено>	
Редактор	свойств	События			

5. Повторите пункты 3 и 4 для *Ap\_Closed*, *Ap\_Stoped* и *Ap\_inProgress*.

Теперь изменим цвет заливки задвижки.

6. В структуре объекта выделите элемент *Valve\_Pic*, перейдите в свойство Цвет заливки → Вычисляемое значение → Формула по условию → Редактировать. Откроется Редактор формул по условию. Введите значения в соответствии с изображением.

Условие		Значение
Ap_Opened		0xff00aa00
Ap_Closed		0xffe70000
Ap_Stoped		0xff00007f
Ap_inProgress		0xffffff00
Нет условия		Нет действия
По умолчанию		Нет действия
О Ошибок <u>1</u> 0 Предупреждений	і) 0 Сообще	ений

7. Сохраните проект, запустите главную форму в RunTime. Теперь задвижки меняют свой цвет в зависимости от состояния.







Самостоятельное задание. У каждого датчика есть набор состояний state: normal, warning, failure. В *SePlatform.HMI* у датчика есть контур прямоугольника (в свойствах он называется пером). Нужно, чтобы контур каждого прямоугольника окрашивался в цвет в зависимости от состояния: normal – зелёный, warning – жёлтый, failure – красный. То есть нужно будет подписаться на дополнительные булевские параметры и сделать окрашивание у датчиков.

🚦 Обозреватель	<b>~</b> ± ×		Структура объекта	
Поиск	<u>ب</u> م	Γ		
<ul> <li>G Astra_UA на 10.0.2.14</li> <li>E Server</li> </ul>			Имя	Описание
🔺 🕖 root			✓	Тип на основе Прямоугольник
🔺 🇭 Sensors			<ul> <li>Графические объекты</li> </ul>	
ExSensor_4			A Text_Pressure	Текст
ExSensor_5     Sensor_5     Sensor_5			A Text_Name	Текст
ExSensor_6			✓ Данные	
▲ ④ info			SensorPath	Уведомляющий string
S Obie	ectName		ApSource_Sensor_Type	Источник АР
Pressure	e		4 Ap_Pressure	Элемент AP float4
👂 🕕 Setlmit			Ap ObjectName	Элемент AP string
4 🏈 State			🐣 Ap Normal	Элемент AP bool
Failute	ire		🐥 Ap Warning	Элемент AP bool
⊵ 🕕 Wan	ning		Ap Failure	Элемент AP bool
	🛛 Демо-проект		- 🗆 X	
	Датчик №1	Датчик №2	Датчик №3	
	70	80	90	
	Задвижка №1	Задвижка №2	2 Задвижка №3	

# Открытие форм через обработчик

Следующая задача – это создание формы управления для каждого датчика и каждой задвижки. У задвижек есть команды управления: open, close, stop. Кнопки управления будут размещены на отдельной форме, через которую в дальнейшем будет осуществляться управление задвижкой. Для управления датчиками тоже создадим отдельную форму, где будет отображено текущее значение, состояние и имя датчика. Также здесь будет размещён флажок, позволяющий запускать и останавливать имитацию.

Для начала создадим новую экранную форму.

1. В библиотеке компонентов (вкладка Проект) щелкните ПКМ по Экранные формы → Создать. Откроется мастер создания элементов. В поле Тип элемента оставьте Тип, Базовый тип элемента – Форма (раздел Визуальные). Назовите эту форму Sensor\_Form.

2. Перейдите в форму Sensor\_Form, в Структуре объекта выделите форму, перейдите к её свойствам. Заголовок окна: <u>Управление датчиком</u>, Размеры окна: <u>Автоподбор</u>, Положение окна: <u>По центру монитора</u>.

Эта форма будет открываться в тот момент, когда произойдёт нажатие мыши по любому датчику. Давайте это настроим.

3. Перейдите в **Графический тип Sensor**, в Структуре объекта выделите прямоугольник **Sensor**, перейдите во вкладку События.





4. Щелкните ПКМ по событию MouseClick  $\rightarrow$  Добавить обработчик  $\rightarrow$  Открыть в новом окне  $\rightarrow$  Sensor\_Form.

Структура объекта				
Имя	(	Описание		
✓	1	Тип на основе Прямоугольник		
Графические объ	екты			
A Text_Pressu	ire 1	Гекст		
A Text_Name	1	Текст		
🗸 Данные				
События				
Q				
Имя	Харак	Обработчик		
MouseMove				
✓ MouseClick				
		Открыть в новом окне		
> Handler_1		C Sensor_Form		
Редактор свойств Соб	бытия			

5. Сохраните проект, запустите главную форму в RunTime. При нажатии на любой из датчиков открывается форму управления.



Самостоятельно задание. Создайте новую фурму управления задвижкой Valve\_Form, задайте все необходимые свойства и настройте открытие этой формы при нажатии на любую из задвижек на главной форме.







# Добавление анимации

Было бы хорошо дать понять пользователю, что элемент интерактивный, и с ним можно взаимодействовать. Обычно это значит, что нужно добавить какую-то анимацию, например, выделение контура элемента.

1. Перейдите в **Графический тип Sensor**. В структуре объекта выделите прямоугольник **Sensor**, перейдите во вкладку События.

2. Щелкните ПКМ по событию MouseEnter → Добавить обработчик → Выполнить код → Редактировать. Откроется окно Исходный код. Впишите сюда код в соответствии с изображением (свойство PenWidth отвечает за толщину пера).

🔀 Исхо	удный код	×
1	PenWidth = 4;	
150% V		
🕘 о Ош	ибок 🔔 0 Предупреждений 🕕 0 Сообщений	
Описа	ание	
<u>Я</u> зык: От	<b>•</b>	ОК Отмена

3. При срабатывании события MouseLeave создайте обработчик, который приводит толщину пера к состоянию, равному 1.

4. Сохраните проект, запустите главную форму в RunTime. При наведении курсора на любую из задвижек контур становится жирнее.







Самостоятельное задание. С делайте то же самое для датчиков: при наведении мыши на прямоугольник контур должен становиться жирнее.



#### Создание ссылок на основе примитива

Теперь необходимо в эти дочерние формы передать имена датчиков и задвижек. Эти имена у нас уже проброшены в типах Sensor и Valve, поэтому необходимо сослаться на элемент *Ap\_ObjectName*, который и отвечает за имя объекта. Имя в дочерних формах Sensor\_Form и Valve\_Form будет размещаться в текстовом поле.

1. Перейдите на экранную форму Sensor\_Form. Перетяните сюда из Библиотеки копмонентов элемент *Текст*, назовите его Text\_SensorForm. Пропишите внутри «Имя», настройте шрифт и положение текста.

2. Для того, чтобы создать ссылку в Структуре объекта нажмите ПКМ по **Sensor\_Form** → Создать. Откроется Мастер создания элементов. В поле Тип элемента выберите Ссылка, Базовый тип – Элемент AP string. В Структуре объекта назовите её \_Ap\_ObjectName\_SensorForm.

🙀 Мастер создания эл	ементов	×
Новый элемент Значения атрибутов	Создание нового элемента Тип элемента Базовый тип элемента Сылка Базовый тип элемента Святия Съзлава Съ	•
	Врэзер и Сочник АР     Запрос значений множества сигналов АР     Запрос значений злемента АР     Я Запрос значений злемента АР     Логочник АР     Запрос значений злемента АР     Логочник АР     Злемент AP string     Злемент AP timestamp	×
	ОК Отмена	1

3. В Структуре объекта выделите элемент *Text\_SensorForm*, в Вычисляемое значение его свойства Текст поместите <u>Ap ObjectName SensorForm</u>

Структура объекта				
Имя Описание				
✓	Тип на основе Форма			
Графические объекты				
A Text_SensorForm	Текст			
🗸 Данные				
🛔 _Ap_ObjectName_Se	ensorForm Ссылка на Элемент AP string			
Редактор свойств				
Q				
Q				
Свойство	Харак Значение			
Свойство ✓ ⑤ Текст	Харак Значение R W 🔆 Имя			
Свойство У Б Текст Начальное значение	Харак Значение <b>R W ©</b> Имя <не определено>			
Q. Свойство ▼ ⑤ Текст Начальное значение Вычисляемое значение	Харак Значение R W © Имя ке определено> (f) _Ap_ObjectName_SensorForm			





Теперь эту ссылку необходимо проинициализировать. Инициализация родительского и дочернего контекста, то есть момент соприкосновения родительского и дочернего контекста, происходит в обработчиках. Форма *SensorForm* открывается тогда, когда мы кликаем на датчик. То есть, когда срабатывает событие MouseClick.

4. Перейдите в тип **Sensor**. В Структуре объекта кликните на прямоугольник **Sensor** и откройте вкладку События. Раскройте список MouseClick, раскройте список Handler. В конце списка есть как раз та самая ссылка, здесь её и нужно проинициализировать. Нажмите ПКМ по значению свойства \_Ap\_ObjectName\_SensorForm → Сослаться → Ap\_ObjectName.

Структура объекта			
Имя	Описание		
✓	Тип на основе Прямоугольник		
<ul> <li>Графические объекты</li> </ul>	······,·····		
A Text_Pressure	Текст		
A Text_Name	Текст		
✓ Данные			
События			
Q			
Имя	Харак Обработчик		
MouseMove			
✓ MouseClick			
✓ Handler_1 0	ткрыть <sensor_form> в новом окне</sensor_form>		
	orForm R ⊆ → ♣Ap_ObjectName		
Редактор свойств События			

5. Сохраните проект, запустите главную форму в RunTime. При нажатии на любой из датчиков открывается форма управления с соответствующей надписью.

🔀 Демо-проект				
Датчик №1	Датчик №2	Датчик №3		
0	10	20		
Задвижка №1	Задвижка №2	Задвижка №3		
👿 Управление датчи	ком	– 🗆 X		
Датчик №1				

Самостоятельное задание. Пробросьте на форму управления задвижек имя соответствующей задвижки.







Это один из способов работы со ссылкой. Давайте попробуем сейчас вывести значение датчика в текстовом поле.

### Создание ссылок на основе графического типа

1. Перейдите в форму SensorForm. Добавьте сюда элемент *Текст* с надписью Значение датчика. Назовите ero Text\_Sensor\_Value. Настройте шрифт, положение текста.

2. Для того, чтобы создать ссылку на графический тип, нажмите в Структуре объекта ПКМ на Sensor\_Form → Создать. Тип элемента: Ссылка, Базовый тип элемента: Типы → Sensor. Назовите её \_Sensor.

Новый элемент	Создание нового элемента
Значения атрибутов	Тип элемента Ссылка
	Базовый тип элемента
	> 🛞 AP
	> 🛞 OPC
	> 🐣 SQL
	> 🐣 Security
	> 🛞 Базовые
	> 🛞 Визуальные
	> 👶 Встроенные
	> 🚓 Графики
	> 🚷 Окружение
	> 🚷 Таблицы
	🗸 Типы
	Rectangle_1
	Sensor
	□ Valve
	Экранные формы

Добавление этой ссылки означает, что мы уже можем взаимодействовать с теми элементами, которые есть внутри графического типа.

3. В Структуре объекта выделите *Text\_Sensor\_Value* и в Вычисляемом значении его свойства Текст введите скрипт в соответствии с изображением.







То есть выведем описание, значение в текстовом формате и единицы измерения элемента *Ар\_Pressure*.

Таким образом можно воспользоваться любым из параметров, который есть внутри типа, к которому ведёт ссылка.

Теперь осталось только проинициализировать эту ссылку.

4. Перейдите в тип **Sensor**. В Структуре объекта выделите прямоугольник **Sensor**, перейдите к его событиям. Раскройте список события MouseClick, раскройте список обработчика Handler, перейдите к пункту \_Sensor, нажмите ПКМ → Сослаться → here (here – специальное слово для того, чтобы ссылаться на весь тип).

Структура объекта	
Имя	Описание
🗙 🗖 Sensor	Тип на основе Прямоугольник
<ul> <li>Графические объекты</li> </ul>	
A Text_Pressure	Текст
A Text_Name	Текст
✓ Данные	
События	
م	
14	V
	Ларак Обработчик
MouseMove	
✓ MouseClick	
✓ Handler_1	Открыть <sensor_form> в новом окне</sensor_form>
🚑 _Ap_ObjectName_Sensor	Form 🛛 🛛 🗧 → 🚑 Ap_ObjectName
Sensor	R ⊆ → □ here
Редактор свойств События	

5. Сохраните проект, запустите главную форму в RunTime. При нажатии на любой из датчиков откроется форма управления датчиком с соответствующими именем и значением датчика.

🛛 Демо-проект				
Датчик №1	Датчик №2	Датчик №3		
10	0	-10		
Задвижка №1	Задвижка №2	Задвижка №3		
🔀 Управление датч	иком	– 🗆 X		
Датчик №1				
Давление 10 МПа				

Самостоятельное задание. У задвижек есть 4 булевских параметра: Opened, Closed, Stoped и inProgress. На форме Valve\_Form необходимо разместить **элемент Текст** и в это поле необходимо





поместить описание состояния задвижки: в зависимости от того, у какого из булевских параметров значение true, то описание и выводить. При выполнении использовать ссылку на графический тип Valve.



### Работа с параметром инициализации

У датчиков и задвижек есть сигналы, через которые можно отправлять различные управляющие воздействия. Например, в случае с датчиком это *сигнал SetImit*, которому можно установить либо true, либо false. Если отправите true, то имитация будет продолжаться, если false, то имитация остановится. У задвижек такими командами являются Open, Close, Stop.

Управляющие воздействия будут отправляться с форм управления. Но для этого нужно пробросить путь до источника. Для того, чтобы создать ссылку на источник, где уже будет путь до определённого объекта, нужно создать **Параметр инициализации**.

### 1. Перейдите в форму Sensor\_Form.

2. Для того, чтобы создать **Параметр инициализации**, щелкните ПКМ на **Sensor\_Form** в Структуре объекта → Создать. Откроется Мастер создания элементов. В поле Тип элемента выберите Параметр инициализации, в поле Базовый тип элемента – АР → Источник АР. Назовите его \_ApSource\_SensorForm.

🔀 Мастер создания элементов		
Новый элемент	Создание нового элемента	
Значения атрибутов	Тип элемента Параметр инициализации	•
	Базовый тип элемента	
	× & AD	
	🖬 Браузер источника АР	
	🔄 Запрос алармов	
	👼 Запрос значений множества сигналов АР	
	🔊 Запрос значений элемента АР	
	💭 Источник АР	
	🚊 Очередь активных алармов	
	ОК Отмена	

3. Перетяните из Библиотеки компонентов элемент AP bool, назовите его Ap\_SetImit.

4. Выделите Ap\_SetImit в Структуре объекта, в свойстве Источник сошлитесь на

<u>\_ApSource\_SensorForm</u>, а в свойстве Путь укажите оставшуюся часть пути: <u>SetImit</u>.

5. Перетяните из Библиотеки компонентов элемент **Флажок** (раздел Визуальные), назовите его CheckBox\_SetImit. В свойстве Текст введите <u>Управление имитацией</u>. Настройте шрифт.

6. В Структуре объекта выделите *Ap\_SetImit*, в Вычисляемом значении его свойства Значение поместите <u>CheckBox\_SetImit.State</u> (то есть меняется State и сразу отправляется в сервер).



Структура объекта	
Имя	Описание
✓	Тип на основе Форма
<ul> <li>Графические объек</li> </ul>	ты
Данные	
🚣 _Ap_ObjectN	lame_SensorForm Ссылка на Элемент AP string
🐥 Ap_SetImit	Элемент AP bool
<ul> <li>Параметры инициа.</li> </ul>	лизации
🛄 _ApSource_S	ensorForm Ссылка на Источник AP
Редактор свойств	
۹	
Свойство	Харак Значение
🕠 Источник	R W → M_ApSource_SensorForm
> 🜀 Путь	R W 🗸 SetImit
✓	R ₩ ⊙ <не определено>
Вычисляемое значе	ение 🗸 (f) Checkbox_SetImit.State
Редактор свойств Собы	тия

Теперь нужно проинициализировать эту ссылку. Инициализация также происходит в момент соприкосновения родительского и дочернего контекста, а этой точкой соприкосновения является момент открытия формы.

7. Перейдите к типу **Sensor**. В Структуре объекта выделите прямоугольник **Sensor**, перейдите во вкладку События. Раскройте список события MouseClick, раскройте список обработчика Handler, перейдите к пункту \_ApSource\_SensorForm, нажмите ПКМ → Сослаться → ApSource\_Sensor\_Type.

Структура объекта	
Имя	Описание
🗸 🔲 Sensor	Тип на основе Прямоугольник
<ul> <li>Графические объекты</li> </ul>	
A Text_Pressure	Текст
A Text_Name	Текст
Данные	
События	
۵,	
Имя	Харак Обработчик
MouseMove	
<ul> <li>MouseClick</li> </ul>	
• Handler_1 Откр	ыть <sensor_form> в новом окне</sensor_form>
🚑 _Ap_ObjectName_SensorForm	R ⊆ → ♣Ap_ObjectName
Sensor	R ⊆ → 🗖 here
🛄 _ApSource_SensorForm	R ⊆ → 🛄 ApSource_Sensor_Type
Редактор свойств События	

8. Сохраните проект, запустите главную форму в RunTime. При нажатии на любой из датчиков открывается форма управления, где можно управлять датчиком.





X	Демо-проект			
	Датчик №1	Датчик №2	Датчик №3	
	10	20	10	
				] 
	Задвижка №1	Задвижка №2	Задвижка №3	
	🙀 Управление датч	иком	- 0	×
	Датчик №1			
	µ ∑ Ynr	авление то	ашией	

Самостоятельное задание. На форме *Valve\_Form* создать кнопки для управления задвижкой. Подписаться на 3 команды управления задвижкой. При нажатии на кнопку отправлять значение true.



### Наследование

В демонстрационном проекте помимо простых датчиков и задвижек имеются ещё расширенные датчики и задвижки. Расширенные датчики предоставляют информацию об уставках, а расширенные задвижки предоставляют информацию о проценте открытия. Их мы будем создавать, как и в *DeveloperStudio*, на основе типов **Sensor** и **Valve**.

*SePlatform.HMI* позволяет разрабатывать графические типы не только на основе графических примитивов, но и на основе других элементов, которые уже существуют в проекте. То есть можно создать графический тип на основе другого типа, и это будет считаться наследованием.

1. Для создания расширенного типа задвижки в Библиотеке компонентов (вкладка Проект) кликните ПКМ по Типы → Создать. Откроется окно Мастер создания элементов. В поле Тип элемента выберите Тип, в поле Базовый тип элемента – Типы → Valve. Назовите его ExValve.




ементов Х
Создание нового элемента Тип элемента Тип Базовый тип элемента
> ♣ AP Типы ☐ Rectangle_1 ☐ Sensor ☐ Valve > Загазные формы
Не задан

Добавим к расширенному типу *Индикатор прогресса*, который будет отображаться в проценте открытия задвижки. Но перед этим разместим экземпляры этого типа на главную форму

2. Перейдите в форму MainForm. Разместите здесь 3 экземпляра типа ExValve.

3. Выделите в Структуре объекта *ExValve\_1* и в свойстве ValvePath укажите оставшуюся часть пути до этой задвижки: <u>ExValve\_4</u>. Проделайте то же самое и с *ExValve\_2* и *ExValve\_3*.

4. Сохраните проект, запустите главную форму в RunTime.

Так как расширенные задвижки созданы на основе простых, на них уже проброшены имя и обработчик открытия формы управления



Самостоятельное задание. Создайте расширенный тип датчика и также разместите 3 экземпляра на главной форме.







5. Перейдите к типу **ExValve**. Из Библиотеки компонентов (раздел Визуальные) перетяните элемент *Индикатор прогресса*. Назовите его ProgressBar\_OpenPrc.

6. Из Библиотеки компонентов перетяните на тип **ExValve** *Элемент AP* uint4, назовите его Ap\_OpenPrc.

У простой задвижки уже есть элемент *АрSource*, то есть уже есть источник с путём до объекта. Значит у расширенной задвижки добавлять Источник не нужно, он как бы есть, но в структуре наследника он не отображается.

7. В Структуре объекта выделите *Ap\_OpenPrc*, в свойстве Источник сошлитесь на <u>ApSource\_Valve\_Type</u>, в свойстве Путь укажите недостающую часть пути: <u>OpenPrc</u>.

8. В Структуре объекта выделите *ProgressBar\_OpenPrc*, в Вычисляемом значении его свойства Значение введите <u>Ap\_OpenPrc</u>.

9. Сохраните проект, запустите его в RunTime. *Индикатор прогресса* отображает процент открытия задвижки.







Самостоятельное задание. На форме Sensor необходимо разместить *Кнопку*, при нажатии на которую должна открываться новая форма. В этой форме нужно отобразить текущее состояние уставок: описание, значение в текстовом формате и единицы измерения. Уставки хранятся внутри *сокета Presets* (HiHiLimit. HiLimit, LoLimit, LoLoLimit). При выполнении задания следует воспользоваться наследованием и передачей ссылки на источник через *Параметр инициализации* для того, чтобы подписаться на дополнительные тэги.







Сохраните проект, передайте его на машину с ОС Linux, запустите проект там (шрифты на ОС Windows и Linux могут отличаться).

#### Установка дополнительных библиотек и готовых решений в SePlatform.HMI

Перед тем, как устанавливать новые компоненты, закройте *SePlatform.HMI* на OC Windows и OC Linux (чтобы дополнительные библиотеки подтянулись в проект).

На OC Windows установите внешние библиотеки и готовые решения:

- *SePlatform.hmi.charts* (для получения возможности представлять данные, принятые от источника, в виде графиков. Внешняя библиотека.);
- SePlatform.hmi.tables (для получения возможности помещать данные, полученные от источника по TCP, в таблицу и формировать собственные таблицы данных. Внешняя библиотека);
- SePlatform.hmi.alarms (компонент, выполненный в виде проекта SePlatform.HMI. Чтобы использовать

SePlatform.HMI.Alarms в проекте автоматизации, нужно подключить его как внешний модуль. Для работы с SePlatform.HMI.Alarms понадобится также библиотека вспомогательных компонентов – внешний модуль CommonLib);

- SePlatform.hmi.trends (компонент, предназначенный для просмотра графиков изменений параметров технологического процесса. Выполнен в виде проекта SePlatform.HMI);
- SePlatform.hmi.commonlib (это расширение среды разработки и исполнения SePlatform.HMI. Расширение представляет собой библиотеку компонентов, которые можно использовать в своих проектах автоматизации: диалоговые окна, контекстное меню, календарь, файловый менеджер, компоненты безопасности).

Чтобы использовать SePlatform.HMI.Alarms и SePlatform.HMI.Trends в проекте автоматизации, нужно подключить их как внешние модули. Для этого:

1. Откройте 2 проводника: папка с проектом в SePlatform.HMI и

C:\Program Files\SePlatform\SePlatform.HMI.Extensions.

2. В папке с проектом создадим папку с именем externals.

3. Скопируйте всё из папки C/Program Files\SePlatform\SePlatform.HMI.Extensions и вставьте в созданную папку externals.

4. Откройте проект в *SePlatform.HMI*, нажмите на вкладку Проект → Обновить ссылки на внешние юниты. После этого в Библиотеке компонентов появится новая вкладка Внешние модули с разделами Alarms, CommonLib и Trends.

🔀 C:/l	.earning/H	MI/Der	noHMI.hm	ii - Дизайнер Alpha.HMI					
Файл	Правка	Вид	Проект	Помощь					
	🖻 🖪		Соз	дать	Ctrl+T		Библ	лиотека компонентов	₽×
Библиот	тека компон	ентов	Cop	тировка			Филь	ьтр	
Фильтр			3an	уск компиляции	F5		>	Общие элементы	
> *	Π		Запуск компиляции текущего типа Показать активную вкладку в рантайме Показать главную форму в рантайме Проверка целостности		Ctrl+F!	> > > >	Внешние модули Alarms CommonLib Trends		
			Ссь	ілки на юниты новить ссылки на внешние юниты		L	/		

Перейдём к установке внешних библиотек на OC Linux. Здесь нужно установить только *SePlatform.hmi.charts* и *SePlatform.hmi.tables*.

- 1. Откройте PuTTY Configuration, проверьте, что Вы находитесь в папке с дистрибутивами.
- 2. Для установки SePlatform.hmi.charts введите команду sudo dpkg -i







#### SePlatform.hmi.charts\*\*\*\*\*\*.deb.

3. Для установки *SePlatform.hmi.tables* введите команду sudo dpkg –i SePlatform.hmi.tables\*\*\*\*\*\*.deb.

Так как у нас в трендах и алармах будут использоваться исторические данные, давайте внесём небольшие изменения в проект *DeveloperStudio*.

#### 1. Откройте *DevStudio*, перейдите в *AstraServer*.

2. Выделите *TcpServer*. В свойстве Номер TCP порта в разделе Настройки доступа к истории введите <u>5388</u> (для того, чтобы *SePlatform.HMI* высчитывал исторические данные).



3. Постройте решение, перейдите к Мастеру развёртывания и примените конфигурацию к AstraServer.

4. Откройте SePlatform.HMI, перейдите в глобальный объект Global.

5. Выделите в Структуре объекта элемент *ApSource\_Main* и в свойстве Порт истории укажите <u>5388</u>. Так же укажите стандартные настройки в соответствии с изображением.



6. Сохраните изменения.





#### Добавление графиков в проект SePlatform.HMI

Модернизируем форму управления датчиком Sensor\_Form.

1. Перейдите в форму Sensor\_Form. Задайте ей ширину <u>1000</u>, высота <u>650</u>.

2. Из Библиотеки компонентов (вкладка Внешние модули) перетяните тип Trends на форму. В

Структуре объекта назовите его Trends. Разместите все имеющиеся на форме элементы в поле зрения.



3. В Структуре объекта выделите элемент *Trends*, для подключения трендов к источнику в свойстве int\_AP сошлитесь на главный источник <u>ApSource\_Main</u>, размещённый в глобальных объектах. Задайте остальные свойства в соответствии с изображением.

Структура объекта	
Имя	Описание
✓	Тип на основе Форма
<ul> <li>Графические объекты</li> </ul>	
A Text SensorForm	Текст
A Text Sensor Value	Текст
Sensor	Ссылка на Sepsor
Checkbox SetImit	Флажок
	Trands
L irenus	irenus
<ul> <li>Данные</li> <li>Параметры инициализации</li> </ul>	
<ul> <li>Данные</li> <li>Параметры инициализации</li> </ul>	
<ul> <li>Данные</li> <li>Параметры инициализации</li> <li>Редактор свойств</li> <li>Предз</li> </ul>	
	Vapar 2
<ul> <li>Данные</li> <li>Параметры инициализации</li> </ul> Редактор свойств Q. Предз Сосйство: (В. Предзагрузка истории в операти	Vapar 2usuana BHOM p R W ≸ true
<ul> <li>Данные</li> <li>Параметры инициализации</li> <li>Редактор свойств</li> <li>Преда</li> <li>Сасйство</li> <li>В Предзагрузка истории в операти Вычисляемое значение</li> </ul>	Харан Значение вном р R W # true <не определено>
<ul> <li>Данные</li> <li>Параметры инициализации</li> <li>Редактор свойств</li> <li>Предз</li> <li>Сесйство</li> <li>В Предзагрузка истории в операти Вычисляемое значение</li> <li>ФЗ Арнимый интервал оперативно</li> </ul>	Харан <mark>Значение</mark> вном р <b>R W ∮ true</b> <не определено> го режима, <b>R W ∮</b> 10
<ul> <li>Данные</li> <li>Параметры инициализации</li> <li>Редактор свойств</li> <li>Предз</li> <li>Слойскае</li> <li>Вычисляемое значение</li> <li>Данисляемое значение</li> <li>Хранимый интервал оперативно Вычисляемое значение</li> </ul>	Харан Энаналия вном р R W ∮ true <ве определено> го режима, R W ∮ 10 <не определено>
<ul> <li>Данные</li> <li>Параметры инициализации</li> </ul> Редактор свойств Предза Соойство <ul> <li>В Предзагрузка истории в операти вычисляемое значение</li> <li>З Хранимый интервал оперативно Вычисляемое значение</li> <li>В Ницимость панели инструментог</li> </ul>	Харак Эначения вном р R W ∮ true <не определено> го режима, R W ∮ 10 <не определено> в R W ∮ true
<ul> <li>Данные</li> <li>Параметры инициализации</li> <li>Редактор свойств</li> <li>Предз</li> <li>Свойство</li> <li>В Предзагрузка истории в операти Вычисляемое значение</li> <li>З Хранимый интервал оперативно Вычисляемое значение</li> <li>В ндимость панели инструментов Вычисляемое значение</li> </ul>	Харас         Эначение           вном р         R W ∮ true               го режима,         R W ∮ 10                Колределено>           s         R W ∮ true
<ul> <li>Данные</li> <li>Параметры инициализации</li> <li>Редактор свойств</li> <li>Преда</li> <li>Свойство</li> <li>В Предзагрузка истории в операти вычисляемое значение</li> <li>Хранимый интервал оперативно Вычисляемое значение</li> <li>В Видимость панели инструментог вычисляемое значение</li> <li>В Ридимость панели инструментог вычисляемое значение</li> <li>В Ридимость панели инструментог вычисляемое значение</li> </ul>	Харак         Значание           вном р         R W ∮ true <he определено="">           го режима,         R W ∮ 10 <he определено="">           в         R W ∮ true <he определено="">           R W ∮ true</he></he></he>
<ul> <li>Данные</li> <li>Параметры инициализации</li> <li>Редактор свойств</li> <li>Предза</li> <li>Предзагрузка истории в операти Вычисляемое значение</li> <li>В Ардимость панели инструментов Вычисляемое значение</li> <li>В Видимость панели инструментов Вычисляемое значение</li> <li>В Режим интерполяции Вычисляемое значение</li> </ul>	Харан         Эначание           вном р         R W \$ true               го режима,         R W \$ 10                   ке определено>            в         R W \$ true

Необходимо, чтобы при открытии формы управления какого-либо датчика, сразу открывался соответствующий график. Для осуществления данной задачи нужно воспользоваться методом ForcedAddItem (добавляет сигнал в список отображаемых сигналов и график сигнала на трендовое поле), которая на вход принимает 3 строковых аргумента: путь до тега, единицы измерения и описание. Этот метод будем использовать при срабатывании события Opened у всей экранной формы.

4. В структуре объекта выделите форму *Sensor\_Form*, перейдите во вкладку События, кликните ПКМ по событию Opened → Добавить обработчик → Выполнить код. Введите скрипт в соответствии с изображением.





🙀 Исходный	код
1 Tr	ends. <i>ForcedAddItem</i> (_Sensor.Ap_Pressure.Tag, _Sensor.Ap_Pressure.Units,
_S	ensor.Ap_Pressure.Description);
150% ~	
🕘 0 Ошибок	1 0 Предупреждений <ol> <li>О Сообщений</li> </ol>
Описание	Положение
<u>Я</u> зык: От	• ОК Отмена

Теперь при открытии формы сразу должен открываться нужный график.

5. Сохраните проект, запустите его в RunTime. Откройте форму управления любого из датчиков. Теперь вместе с этой формой появляется соответствующий график.

🕎 Управление датчиком					- 🗆 ×
Датчик №1	Давл	ение 50 МПа	🗹 Управлени	е имитацией	
🗲 😂 । 🔅 📶 र 层 र 🖏 😓 ।	<b>- 1</b> = 1	■本図 お	🌾 📕 🔍 🔍	<b>Q Q </b> (a (a) (l)	
> NativeTypes	111.0000				
> GojectTypes					
> proxies	100.0000	11111	111111	1111111	1 1 1 1 1 1 1
> SandBox					
> Service	80.0000		ппипи	пппппп	пппппп
			плапла	плвлвпв	
		<u>n 11 11 11 11 11 11</u>	11 11 11 11 11 11 11		
	60.0000	* ** ** ** ** **	** ** ** ** ** **	** ** ** ** ** ** **	<del>                                      </del>
		#######################################			
	Ε 40.0000	шиннин	шаша	пппппп	
	40.0000	11 11 11 11 11 11		ר זר דר דר ער דר ער ד	
		11 11 11 11 11 11	t 11 17 17 17 17 17 1	17 17 17 17 17 17 17 17 1	1 17 11 11 11 17 11 1 7 1
	20.0000	++ ++ ++ ++ ++		{	* ** ** ** ** ** ** **
		ПЛПТЛ			1
	0.0000				
	0.0000	111111			
	-21.0000				
		06:48:00	06:50:00	06:52:00 06:54:0	0 06:56:00
		2022-07-14	2022-07-14	2022-07-14 2022-07	-14 2022-07-14 Время
	$\mathbb{N}$				
4					

#### Отображение событий в проекте SePlatform.HMI

Теперь добавим ещё одну форму для отображения событий. Она будет универсальной и для датчиков, и для задвижек. Событий тоже будем фильтровать: при открытии формы любого из элементов, отображаются отфильтрованные события по соответствующему объекту.

1. Создайте новую экранную форму, назовите её Alarms\_Form.

2. Перейдите в форму Alarms\_Form, в свойстве Заголовок окна введите <u>События</u>, в свойстве Размеры окна: <u>Автоподбор</u>, Положение окна: <u>По центру монитора</u>. Установите ширину формы <u>1000</u>, высоту – <u>600</u>.

3. Из Библиотеки компонентов (вкладка Внешние модули) перетяните тип Alarms. В Структуре объекта назовите его Alarms.





Библиотека компонентов 🗗 🗙	MainForm 🔀 Sensor_Form 🔀 Alarms* 🗵		Структура объекта	
Фильтр		m		
> Общие элементы			Имя	Описание
<ul> <li>Внешние модули</li> </ul>	ч Некато интеррала: 🔯 Соонт-бря (2020 г.) 🗸 (00,00.00 💭 Кринц интеррала: 🖾 сонт-бря (2020 г.) 🔻 (00,00.00 💭	•	Y Alarms	Тип на основе Форм
✓ Alarms			У Графинестие объекты	initia ocnose +opin
Глобальные объекты				Alarmar
∀ Типы			L Alams	Aldinis
Alarms				
> Экранные формы				
> CommonLib				
> Trends				
Проект <demohmi></demohmi>				

4. Для того, чтобы *Alarms* получали данные от источника, в свойстве Ссылка на источник AP сошлитесь на главный источник <u>ApSource Main</u>, который находится в глобальном объекте <u>Global</u>.

Структура объекта				
Имя		Опи	сани	e
🗸 🛅 Alarms		Типн	на о	снове Форма
У Графические	объекты			
Alarms	;	Alarn	ns	
Редактор свойств				
Q				
Свойство		Харак	Зна	чение
🛄 Ссылка на ис	точник АР	R⊆→		unit.Global.ApSource_Main
Редактор свойств	События			

Теперь нужно здесь осуществить процесс фильтрации. Для того, чтобы это сделать, будем использовать метод SetAdvancedFilter (фильтрует отображаемые сообщения о событиях по указанному столбцу таблицы сообщений.) Нужно фильтровать по столбцу Источник (идентификатор данного столбца – source).

5. Перейдите на форму Sensor\_Form. Создайте здесь *Кнопку* с именем Button\_Alarms (при нажатии на эту кнопку будет открываться форма Alarms\_Form). В свойстве Текст введите <u>События</u>. Настройте размер шрифта.

6. В Структуре объекта выделите Кнопку *Button\_Alarms*, перейдите во вкладку События, щёлкните ПКМ по ButtonPressed → Добавить обработчик → Открыть в новом окне → Alarms\_Form.

Структура объекта	
Имя	Описание
M E Course Forms	T
Sensor_Form	тип на основе форма
• трафические объекты	_
A Text_SensorForm	Текст
A Text_Sensor_Value	Текст
Sensor	Ссылка на Sensor
Checkbox_SetImit	Флажок
Trends	Trends
Ettel Button_Alarms	Кнопка
> Данные	
> Параметры инициализац	ии
События	
Q	
Имя Харак	Обработчик
✓ ButtonPressed	•
	Открыть в новом окне
Handler_1	Carms_Form
Редактор свойств События	





7. Сохраните проект, запустите в RunTime. Откройте форму управления одного из датчиков, нажмите на кнопку События. Убедитесь, что события (пока ещё со всего проекта) отображаются.



Теперь настроим саму фильтрацию.

8. Перейдите в форму Alarms\_Form, создайте здесь Уведомляющее поле → уведомляющий string. В Структуре объекта назовите его SourceFiltre (здесь от родительского окна к дочернему будет передаваться путь).

9. В Структуре объекта выделите форму *Alarms\_Form*, перейдите во вкладку События, щёлкните ПКМ по событию Opened → Добавить обработчик → Выполнить код → Редактировать. Введите код в соответствии с изображением.

🔀 Исхо	удный код	×
1	Alarms.SetAdvancedFilter("source",	SourceFilter);
150% V		
🕒 0 Ош	ибок 🔬 0 Предупреждений 🕕 0 Сообщений	
Описа	Положение	
<u>Я</u> зык: От	T	ОК Отмена

Теперь нужно в это Уведомляющее поле при открытии окна передать значение.

10.Перейдите в форму Sensor\_Form, в Структуре объекта выделите Кнопку Button\_Alarms, перейдите во вкладку События, раскройте список события ButtonPressed, раскройте список обработчика Handler, найдите свойство SourceFiltre, щёлкните по нему ПКМ → Задать формулу. Введите формулу в соответствии с изображением (то есть берём тег с именем формы, удаляем из него путь, который дописывали и меняем его на пустоту).





Структура объекта				
Имя  ✓ Графические объекты  A Text_SensorForm  A Text_Sensor_Value  Sensor  Checkbox_SetImit  Trends  Image Button_Alarms  Дараметры инициализации	Описание Тип на основе Форма Текст Текст Ссылка на Sensor Флажок Trends Кнопка		<mark>№</mark> Редак 1 150% ~	ор свойств × String. <i>RepLace</i> (_Ap_ObjectName_SensorForm.Tag, _Ap_ObjectName_SensorForm.Path, "") ОК Отмена
События				
Имя	Харак Обработчик			
✓ ButtonPressed		Сослат	ъся	
✓ Handler_1	Открыть <alarms_form> в н</alarms_form>	Указать	значение	
SourceFilter     Pедактор свойств     События	К W 7 <не определено>	(f) Задать	формулу	
		🔗 Выреза	ать Ctrl+	X

11.Сохраните проект, запустите его в RunTime. Откройте форму управления любого из датчиков, нажмите на кнопку События. Теперь на форме отображаются события по конкретному датчику.

🔀 События	
🗲 号   🐯 🗖 🎚	<b>- </b> □
Время генерации	Сообщение
14.07.2022 07:45:35.145	Демо-проект. Датчики. Датчик №1. Объект перешёл в аварийное состояние
14.07.2022 07:45:32.125	Демо-проект. Датчики. Датчик №1. Объект перешёл в предаварийное состояние
14.07.2022 07:45:29.109	Демо-проект. Датчики. Датчик №1. Объект перешёл в нормальное состояние
14.07.2022 07:45:26.097	Демо-проект. Датчики. Датчик №1. Объект перешёл в предаварийное состояние
14.07.2022 07:45:24.086	Демо-проект. Датчики. Датчик №1. Объект перешёл в аварийное состояние
14.07.2022 07:45:21.021	Демо-проект. Датчики. Датчик №1. Объект перешёл в предаварийное состояние
14.07.2022 07:45:18.004	Демо-проект. Датчики. Датчик №1. Объект перешёл в нормальное состояние
🔲 🛒 🛒 🜒 27 сообщ	ений из 171

Самостоятельное задание. Добавьте кнопку на форме управления задвижкой, которая будет открывать форму с отфильтрованными событиями (воспользуйтесь методом String.Replace).



🗲 号   🐯 🗖 🖡	▋ 🖫 🖶    ▾ ╤↓   🕕 ଏ⊚ ଏ▶। ଏ×
Время генерации	Сообщение
14.07.2022 07:52:21.645	Демо-проект. Задвижки. Задвижка №2. Задвижка остановилась
14.07.2022 07:52:21.443	Демо-проект. Задвижки. Задвижка №2. Подана команда на остановку задвижки
14.07.2022 07:52:19.381	Демо-проект. Задвижки. Задвижка №2. Подана команда на закрытие задвижки
14.07.2022 07:52:17.166	Демо-проект. Задвижки. Задвижка №2. Задвижка пришла в движение
14.07.2022 07:52:16.915	Демо-проект. Задвижки. Задвижка №2. Подана команда на открытие задвижки
14.07.2022 06:06:53.511	Демо-проект. Задвижки. Задвижка №2. Объект перешёл в нормальное состояние
14.07.2022 06:06:53.511	Демо-проект. Задвижки. Задвижка №2. Задвижка закрылась

Сохраните проект, передайте его на машину с ОС Linux, проверьте работу.

#### Использование безопасности. Установка SePlatformSecurity. Конфигурирование OpenLDAP

В проектах автоматизации на различных производствах и не только очень важно обеспечить безопасность контролируемого технологического процесса, обеспечить аутентификацию пользователей для разграничения прав и предотвратить несанкционированный доступ. За обеспечение безопасности в



проекте отвечает компонент *SePlatform.Security*, который состоит службы из **SePlatform.Security.Agent**, отвечающей за аутентификацию и распределение прав пользователей, и **OpenLDAP** являющийся защищённой базой данных пользователей и их прав. Также есть приложение *SePlatform.Security Конфигуратор*, с помощью которого осуществляется конфигурирование OpenLDAP.



Прежде чем устанавливать дистрибутивы давайте закроем *SePlatform.HMI* на обеих машинах предварительно сохранив проект. Мы будем загружать в *SePlatform.HMI* дополнительные библиотеки, и для того, чтобы они были правильно загружены и установлены, необходимо закрыть программу.

Так как разработка проекта кроссплатформенная, в этом разделе будем делать акцент преимущественно на OC Linux (всю подробную информация по установке *SePlatform.Security* можно найти в документации).

1. Откройте PuTTY, убедитесь в том, что Вы находитесь в папке с установочными файлами, введите команду для установки *SePlatform.Security*: sudo dpkg —i SePlatform.security\*\*\*\*\*\*.deb.

2. Для того, чтобы настроить OpenLDAPвведите команду sudo apt-get install slapd Idap-utils. После вопроса «Хотите продолжить?» введите Y (yes). Откроется окно настройки пакета.

OpenLDAP — это хранилище, в хранилище могут находиться несколько баз данных безопасности, здесь Вы указываете пароль для хранилища, то есть пароль для того, чтобы подключиться. Обычно по умолчанию используется пароль secret.

- 3. В поле Пароль администратора введите secret, нажмите ОК.
- 4. Повторно введите пароль secret.

 Чтобы указать свои данные в конфигурации OpenLDAP, переконфигурируйте его командой sudo dpkg-reconfigure slapd. Ответьте на следующие вопросы в соответствии с изображением (в полях для пароля вводите secret).

- > Не выполнять настройку сервера OpenLDAP? «Нет»
- > Доменное имя DNS «maxcrc.com»
- > Название организации <название> (можно указать любое)
- > Пароль администратора <пароль>
- > Повторный пароль <пароль>
- > Используемые серверы баз данных «HDB»
- Удалять базу данных при вычитке slapd? «да»
- > Переместит старую базу данных? «да», если есть файлы





6. Перейдите в Midnight Commander путём ввода команды <mark>sudo mc</mark>. Перейдите в папку /opt/SePlatform/SePlatform.Security, выделите файл **SePlatform.security.ldif**.

7. Для того, чтобы перейти в правую часть коммандера нажмите клавишу Tab.

8. В правой части коммандера перейдите в папку /etc/ldap/schema. Вернитесь в левую часть коммандера нажатием клавиши Tab.

9. Скопируйте из /opt/SePlatform/SePlatform.Security в /etc/ldap/schema файл SePlatform.security.ldif нажатием на клавишу F5.

Pmc [root@astra]:/etc/ldap/schema			- 0 )
Левая панель Файл Команда Настройки	Правая панель		
<pre>- /opt/Automiq/Alpha.Security</pre>	.[^]>	/etc/ldap/schema	.[^]>
.и Имя	Размер Время правки	и Имя	Размер Время правки
1	-ВВЕРХ- май 26 07:41	/	-ВВЕРХ- май 25 12:37
/Utils	4096 май 25 12:31	README	3512 anp 20 2020
*alpha.security.agent	147455K des 18 14:24	a.ldif	111 май 25 12:40
alpha.security.agent.json	25409 фев 18 14:24	a2.ldif	113 май 25 12:41
alpha.security.agent.xml	14061 май 25 13:05	alpha.security.ldif	4769 des 18 14:24
alpha.security.fa.xml	6935 фев 18 14:24	collective.ldif	2036 anp 20 2020
alpha.security.ic.xml	1802 den 18 14:24	collective.schema	2180 anp 20 2020
alpha.security.ldif	4769 фев 18 14:24	corba.ldif	1845 anp 20 2020
alpha.security.schema.export.sh	94 фев 18 14:24	corba.schema	2084 anp 20 2020
*alpha.security.useractivity	51636К фев 18 14:24	core.ldif	21196 anp 20 2020
*alpha.security.useractivity.add.anotheruser.sh	1458 фев 18 14:24	core.schema	21083 anp 20 2020
*alpha.security.useractivity.sh	771 den 18 14:24	cosine.ldif	12006 anp 20 2020
*libtcp transport.so	2153240 фев 18 14:24	cosine.schema	14030 anp 20 2020
openldap-conf-and-data-backup.sh	146 des 18 14:24	duaconf.ldif	4842 amp 20 2020
openldap-conf-and-data-restore.sh	346 des 18 14:24	duaconf.schema	6249 anp 20 2020
openldap-enable-syncrepl-consumer.ldif	466 фев 18 14:24	dyngroup.ldif	3330 amp 20 2020
openldap-enable-syncrepl-consumer.sh	85 фев 18 14:24	dyngroup.schema	3289 anp 20 2020
openldap-enable-syncrepl~tiprovider-server-1.ldi	1524 фев 18 14:24	inetorgperson.ldif	3481 anp 20 2020
openIdap-enable-syncrepl~ultiprovider-server-1.s)	99 des 18 14:24	inetorgperson.schema	3915 anp 20 2020
openldap-enable-syncrepl~tiprovider-server-2.ldi;	1524 фев 18 14:24	java.ldif	2979 апр 20 2020
openIdap-enable-syncrepl~ultiprovider-server-2.s)	99 des 18 14:24	java.schema	3295 апр 20 2020
openldap-enable-syncrepl-provider.ldif	446 des 18 14:24	misc.ldif	2082 and 20 2020
openldap-enable-syncrepl-provider.sh	85 dep 18 14:24	misc.schema	2387 апр 20 2020
		nis.ldif	6809 and 20 2020
		nis.schema	7640 and 20 2020
		openldap.ldif	3308 and 20 2020
		openIdap.schema	1514 amp 20 2020
		pmi.ldif	6904 апр 20 2020
		pmi schema	21051 amp 20 2020
		pholicy ldif	4570 am 20 2020
		ppointegriait	
alpha security ldif		alpha security ldif	
alpha.security.idii	64711/120 (40%)	aipha.securicy.iuii	647134/120 (405)

10.Вернитесь к командной строке, нажав на клавишу F10. Следующие команды необходимо вводить, находясь в режиме суперпользователя. Для перехода в этот режим введите команду sudo su (теперь для ввода команд не нужно в начале использовать слово sudo).

11.Перейдите в директорию /etc/ldap/schema введя команду cd /etc/ldap/schema.

12.Находясь в директории /etc/ldap/schema, выполните команду ls \*.ldif | xargs -t -n 1 ldapadd -Q -Y EXTERNAL -H ldapi:/// -f.

13.Находясь в директории /etc/ldap/schema, выполните команду <mark>ldapadd -Q -Y EXTERNAL -H</mark> ldapi:// -f SePlatform.security.ldif.

14.Перейдите в Midnight Commander, введя команду <mark>mc</mark>. Перейдите в /etc/ldap/schema. Нажмите комбинацию клавиш Shift+F4 для создания файла. Пропишите в нём строчки, описанные ниже (для вставки текста используйте ПКМ или комбинацию клавиш Shift+Insert):

dn:olcDatabase={1}hdb,cn=config

changetype: modify

replace: olcAccess

dn:olcDatabase={1}hdb,cn=config changetype: modify replace: olcAccess olcAccess: to \* by users write by \* read

olcAccess: to \* by users write by \* read

15.Нажмите клавишу F2 для сохранения. Здесь же назовите этот файл *a.ldif*. Выйдите из файла нажатием клавиши F10. Выйдите в командную строку (F10).

16.Проверьте, что Вы находитесь в директории /etc/ldap/schema. Находясь в этой директории введите команду sudo ldapmodify -Y EXTERNAL -H ldapi:// -W -f a.ldif. Введите тот самый пароль, который был указан при установке: secret (ввод будет зашифрован, поэтому никаких символов при вводе Вы не увидите).

17.Перейдите в Midnight Commander, введя команду <mark>mc</mark>. Перейдите в /etc/ldap/schema. Нажмите комбинацию клавиш Shift+F4 для создания файла. Пропишите в нём строчки, описанные ниже (для вставки текста используйте ПКМ или комбинацию клавиш Shift+Insert):





dn:olcDatabase={1}hdb,cn=config changetype: modify replace: olcAccess

olcAccess: {0}to \* by users write by \* read

18.Нажмите клавишу F2 для сохранения. Здесь же назовите этот файл *a2.ldif*. Выйдите из файла нажатием клавиши F10. Выйдите в командную строку (F10).

19.Проверьте, что Вы находитесь в директории /etc/ldap/schema. Находясь в этой директории введите команду sudo ldapadd -Y EXTERNAL -H ldapi:// -W -f a2.ldif. Введите тот самый пароль, который был указан при установке: secret (ввод будет зашифрован, поэтому никаких символов при вводе Вы не увидите).

OpenLDAP на OC Linux переконфигурирован.

Так же установим на машину с ОС Linux пакет для *SePlatform.HMI*, чтобы он мог работать с библиотекой безопасности.

20.Откройте PuTTY, введите в командную строку команду exit для выхода из режима суперпользователя.

21.Убедитесь, что Вы находитесь в папке с установочными файлами и введите команду установки: sudo dpkg –i SePlatform.hmi.security\*\*\*\*\*\*\*.deb

Теперь необходимо установить конфигуратор безопасности. Чтобы это сделать на машине с ОС Windows, установите дистрибутив: *SePlatform.Security*. Этот компонент устанавливает сразу несколько компонентов: *SePlatform.Security* – это SePlatform.Security агент, который нужен на каждой локальной машине для того, чтобы *SePlatform.HMI* умел работать с компонентами безопасности. OpenLDAP уже установлен на

линуксе, компонент Security для HMI – компонент, который нужен, чтобы *SePlatform.HMI* умел взаимодействовать с безопасностью. Он подложит специальную библиотеку в папку с установленным *SePlatform.HMI. Конфигуратор Security* – это тот самый конфигуратор, который нужен для того, чтобы сконфигурировать базу, то есть добавить пользователей и приложение.

#### Работа с SecurityConfigurator

1. Запустите установленный Конфигуратор из меню Пуск  $\rightarrow$  SePlatform  $\rightarrow$ 

SecurityConfigurator. В строке Сервер введите IP-адрес машины с OC Linux, уберите галочку с TLS. Придумайте имя

пользователя и пароль для него. Нажмите Подключиться.

Подключение к серверу							
Укажите имя сервера для подключения, введите имя пользователя и пароль							
Сервер: 10.0.2.14 ~							
Безопасность: SSL TLS							
Корневая папка:	ou=AlphaSecurity,dc=maxcrc,dc=com ~						
Пользователь:	root						
Пароль:	••••						
EN	Проверить Подключиться Отмена						

2. «Создать ли новую конфигурацию?» → Да.

3. В строке Администратор LDAP меняем «Manager» на «Admin». Пароль администратора LDAP – secret. Нажмите Далее.

4. Вводите имя пользователя, которого хотите создавать и пароль для этого пользователя. После этого нажмите Готово.





LDAP Администратор	Администратор	
	В создаваемую конфигу пользователь с правами конфигурации безопасн его пароль.	рацию будет добавлен новый и на редактирование юсти. Укажите имя этого пользователя и
	Администратор безопасности:	root
	Пароль администратора:	
	Подтверждение пароля:	•••₹
	подверядение переия.	

Начнём создавать пользователей. Важное замечание: пользователи должны состоять в группе, так как права распространяются на группу, а не на конкретного пользователя.

5. Нажмите на кнопку Приложения.



6. Нажмите на кнопку Добавить и введите название HMI.

7. Для того, чтобы создать право, нажмите на кнопку Логическое право. Назовите его OpenPopup (право на открытие формы).

8. Чтобы дать ему более понятное описание, сначала сохраните изменения и перейдите внутрь этого права. В свойстве Описание введите: Право на открытие формы управления. Сохраните изменения и снова перейдите в приложение.

9. Нажмите на кнопку Группы → Добавить. Назовите группы Dispatchers.

10.Нажмите на кнопку Править и здесь – Добавить права. Выберите право OpenPopup внутри HMI. Явно укажите, что это право разрешено для группы Диспетчеров.

🏝 Dispatchers					
Идентификатор	Dispatchers	Тип	Право	Значение	Эффективное значение
Отображаемое имя	Dispatchers	) (O I	HMI		
Роли		₿	Право на открытие формы управления	<b>Ø</b> 🗢	<b>S</b>

#### 11.Сохраните изменения.

Самостоятельное задание. Создайте группу Операторов, запретите им право на открытие формы управления.

🌯 Operators					
Идентификатор	Operators	Тип	Право	Значение	Эффективное значение
Отображаемое имя	Operators	۰ ۱	IMI		
Роли		B	Право на открытие формы управления	Ø 🗢	•

Теперь можем переходить к созданию пользователей.

12. Нажмите на кнопку Пользователи. Введите логин, пароль, состоящий из 7 символов и ФИО пользователя.

13.Нажмите на кнопку Добавить в группы → Dispatchers. Снимите галочку с Требовать смены пароля при следующем входе в систему. Сохраните изменения.





Главная								
Назад Вперед Обновить Навигация	Приложения Рабочи места Ра	е Групп зделы	а Пользователи	Анарияния и Соловий Полизователь Поли	Задать пароль зователь	Добавит в группи	Ана Собав назначить Добав роли прав Права	ить Удалит грава
🤱 Диспетчеров Дисп	етчер Диспетчерс	вич						
Логин	Dispatchers	Тип	Прав	0	Значение		Эффективное значен	ние
Пароль		$\odot$	Alpha.Security					
Подтверждение					🗹 Цифры		Цифры	
Фамилия	Диспетчеров	0	Сложность парол	я	Буквы нижнего	регистра		
Имя	Диспетчер				Специальные с	имволы		
Отчество	Диспетчерович	0	Минимальная дл	ина пароля		7	7	
Отображаемое имя	Диспетчеров Диспетчер Диспетчерович		Количество паро	лей в истории		4	4	
Должность			nvn				•	
Подразделение			право на открыт	ке формы упр			0	
Адрес почты								
Телефон								
Дополнительные сведения								
Группы	A Dispatchers							
Роли								

Самостоятельное задание. Создайте нового пользователя, добавьте его в группу операторов.

#### Настройка SePlatform.secutiry.agent.XML

Настроим **SePlatform.security.agent** на машине с OC Windows.

- 1. Перейдите в папу C:\Program Files\SePlatform\SePlatform.Security, откройте SePlatform.secutiry.agent.XML с помощью NotePad++.
- В строке с тэгом EntryPointNetAgent введите адрес той машины, на которой Вы находитесь IPадрес Wndowsмашины.

7	>	
8	<pre><entrypointnetagent address="&lt;/pre&gt;&lt;/th&gt;&lt;th&gt;10.0.2.17" port="1010"></entrypointnetagent></pre>	
9		

3. В строке с тэгом LDAPServer введите адрес машины, на которой сконфигурирован LDAPServer – IP-адрес Linux машины.

			±	T	±	1		
15	-	>						
16 E	þ	<ldaphosts></ldaphosts>						
17		<ldapserver ac<="" th=""><th>ldress="</th><th>10.0.2.14"</th><th>Port</th><th>:="389"/&gt;</th><th></th><th></th></ldapserver>	ldress="	10.0.2.14"	Port	:="389"/>		
18		<LDAPServ</th <th>ver Addr</th> <th>ess="199.99</th> <th>.99.</th> <th>111" Port</th> <th>="389"/&gt;</th> <th>&gt;</th>	ver Addr	ess="199.99	.99.	111" Port	="389"/>	>
19								

Если OpenLDAP сконфигурирован на OC Windows то в строке с тэгом LdapUser в атрибуте value указать cn=manager, если же на OC Linux, то cn=admin.

- В строке с тэгом LdapPassword необходимо ввести хэшированный пароль. Захешировать его можно с помощью специальной программы SePlatform.security.crypter.exe через командную строку:
  - 1) Откройте через меню Пуск командную строку.
  - Для перехода в папку с программой введите cd C:\Program

Files\SePlatform\SePlatform.Security\Utils нажмите Enter.

- Для запуска программы введите SePlatform.security.crypter.exe.
- 4) Введите пароль secret и скопируйте получившуюся строку (для сохранности

вставьте в отдельный блокнот). Вставьте этот пароль в строку с тэгом LdapPassword.



3)

<!-- Пароль LDAP --> <LdapPassword value="qPQg3EuTdhKym8Cp9AQzTPHeaCWG6X/gGXbRwKcHKhVeEjW4cVHbXQYBHf3XH8SUc

 Отключите выполнение контроля целостности: в строке с тэгом Options в атрибуте ICMode введите значение 0.







6. Сохраните SePlatform.security.agent.XML.

7. Откройте Диспетчер задач → Службы, перезапустите службу SePlatform.Security.Agent.

Теперь настроим SePlatform.security.agent на машине с OC Linux.

 Откройте PuTTY, перейдите в Midnight Commander при помощи команды sudo mc. Перейдите в директорию /opt/SePlatform/SePlatform.Security, откройте файл SePlatform.security.agent.hml нажатием клавиши F4.

ns LoggerLevel="2" ICMode="0" kbDriverString="0x1D+0x38+0x53;0x1D+0x2A+0x01;" UseRightsCacheStorage="0" 💋

9. В строке с тэгом EntryPointNetAgent введите адрес той машины, на которой Вы находитесь – IPадрес Linux машины.

---><EntryPointNetAgent Address="10.0.2.14" Port="1010"/>

10. В строке с тэгом LDAPServer введите адрес машины, на которой сконфигурирован LDAPServer – IP-адрес Linux машины.



Если OpenLDAP сконфигурирован на OC Windows то в строке с тэгом LdapUser в атрибуте value указать cn=manager, если же на OC Linux, то cn=admin.

11. В строку с тэгом LdapPassword необходимо ввести хэшированный пароль, который был сгенерирован в пункте 4 (пароль secret).

<!-- Пароль LDAP --> </LapPassod value="#ppg3surdhKym8cp9AgzmPHeacW66X/gGXbRwKcHKhVeEjW4cVHbXQYBHf3XH8SUdyDo7XmSSy4ileIKyD6KHq9EF45huBFl3+25bqUhVuu81UK/L

12. Отключите выполнение контроля целостности: в строке с тэгом **Options** в атрибуте **ICMode** введите значение 0.

<Options LoggerLevel="2" ICMode="0" UseRightsCacheStorage="0" FAMode="1"/>

- 13. Сохраните **SePlatform.security.agent.hml** нажатием клавиши F2, перейдите в командную строку нажатием клавиши F10.
- 14. Перезапустите службу командой sudo systemctl restart SePlatform.Security.
- 15. Проверьте её статус: sudo systemctl status SePlatform.Security.





#### Добавление в проект компонентов безопасности

Откройте *SePlatform.HMI* на машине с OC Windows, откройте проект. Обратите внимание, в библиотеке компонентов добавился раздел Security.

Биб	блиот	тека компонентов	Ð	×
Фил	пьтр			
~		Общие	элементы	
>	&	AP		
>	&	OPC		
>	&	SQL		
>	&	Security		
>	&	Базовые		
>	&	Визуальные		
>	&	Встроенные		
>	&	Графики		
>	*	Окружение		
>	&	Таблицы		

На главной форме создадим поле для аутентификации. Разместим на главной форме текстовое поле, куда будет выводиться логин текущего пользователя, добавим 2 поля ввода: логин и пароль. Также добавим кнопку, нажимая на которую будет производиться вход.

1. На форме MainForm разместите элемент *Текст*, назовите его Text\_CurrentUser. В свойстве Текст укажите <u>Текущий пользователь</u>. Настройте шрифт и положение текста.

2. Добавьте сюда же 2 *Поля ввода*: TextEdit\_UserName и TextEdit\_Password. В Свойстве Текст заполнителя укажите <u>Логин</u> и <u>Пароль</u> соответственно. Настройте шрифт и положение текста. У поля ввода *TextEdit\_Password* в свойстве Скрывать ввод укажите значение <u>true</u> для того, чтобы никто не увидел пароль пользователя.

3. Из Библиотеки компонентов перетяните на форму MainForm элемент *Кнопка*, назовите её Button\_Login. В свойстве Текс укажите <u>Войти</u>. Настройте шрифт и положение текста.



Для того, чтобы взаимодействовать с безопасностью, необходимо использовать элементы безопасности. Элементы безопасности мы поместим в глобальные объекты, т.к. они будут использоваться всюду.

4. Перейдите в глобальный объект Global, из Библиотеки компонентов (раздел Security) перетяните сюда элемент *Контекст безопасности*. В Структуре объекта назовите его SecurityContext\_Global.





Добавление *Контекста безопасности* в глобальные объекты означает то, что в проекте будет взаимодействие с агентом безопасности.

5. Перейдите в MainForm. Для того, чтобы вывести текущего пользователя, выделите в Структуре объекта *Text\_CurrentUser*, перейдите в Вычисляемое значение его свойства Текст, введите сюда скрипт в соответствии с изображением:

Структура объекта					
Имя	Описание				
<ul> <li>MainForm</li> <li>Графические объекты</li> </ul>	Тип на основе Форма				
А Text_CurrentUser Текст					
Редактор свойств					
Q					
Свойство	Харак Значение				
✓ S Текст	R W 🗇 Текущий пользователь				
Начальное значение	<не определено>				
Вычисляемое значение	✓ (f) unit.Global.SecurityContext_Global.Current	entUser			

6. Сохраните проект, запустите в RunTime.

guest
Логин
Пароль
Вход

Теперь начнём обеспечивать вход конкретному пользователю.

7. Выделите *Кнопку Button\_Login*, перейдите во вкладку События, щелкните ПКМ по событию ButtonPressed → Выполнить код → Редактировать. Введите скрипт в соответствии с изображением.

🔀 Исхо	одныі	ый код			×
1	u	nit.Global	<pre>SecurityContext_Global.Login(TextEdit_UserName,</pre>	TextEdit_	Password);
200% ~					
0 o Ou	ибок	с 🕼 0 Предупреждений	(j) 0 Сообщений		
Описа	ание	•	Положение		
Язык: От	ı	•			ОК Отмена

8. Сохраните проект, запустите в RunTime. Попробуйте зайти под Оператором и под Диспетчером.

Dispatcher	Operator
Dispatcher	Operator
•••••	•••••
Вход	Вход





Когда пользователю будет разрешено право OpenPopup, то он сможет открыть форму управления, если запрещено, то нужно вызвать форму с текстом «Доступ запрещён».

9. Создайте новую экранную форму, назовите её AccessDeied\_Form. Зайдите внутрь настройте свойства Размеры окна и Положение окна.

10.Перетяните на форму элемент *Текст* с надписью <u>Доступ запрещён!!!</u>. Настройте шрифт и положение текста.

11.Перетяните на *Текст* «Доступ запрещён!!!» элемент *Мигание* (раздел Визуальные). В свойстве Период укажите <u>200</u>, Активность <u>true</u>.

	Структура объекта				
Доступ	Имя Имя Пассезя Denied_Form Графические объекты А Text_1 Данные Вlinker_1		Описание Тип на основе Форма Текст Мигание		
	Редактор свойств				
запрешении					
	Свойство	Харак	Значение		
	> 🚳 Период	RW	200		
	> (В) Активность	RW	true		
	Редактор свойств События				

Для того, чтобы явно указать, что в проекте будет взаимодействовать с безопасностью (конкретнее с булевским правом), необходимо добавим **Булевский элемент безопасности**.

12.Перейдите в глобальный объект Global и перетяните сюда из Библиотеки компонентов (раздел Security) *Булевский элемент безопасности*, назовите его OpenPopup\_Token. Укажите свойства в соответствии с изображением (значения свойств Приложение и Право берутся из SecurityConfigurator).

Структура объекта				
Имя		Описание		
✓ {♣} Global		<базовый тип не задан>		
🛩 Данные				
🐺 ApSource_Main		Источник АР		
ApSource_Sensors_	Global	Источник АР		
ApSource_Valve_Gl	obal	Источник АР		
🔓 SecurityContext_GI	SecurityContext_Global			
OpenPopup_Token				
Редактор свойств				
Q				
Свойство	Харак	Значение		
Отображаемое имя		OpenPopup_Token		
🚯 Карлинальное число		1		
🔓 Контекст безопасности	R≦→	SecurityContext_Global		
(5) Приложение	Rsv	HMI		
S Право	RSV	OpenPopup		
Редактор свойств События				

Сейчас форма управления открывается сразу без каких-либо проверок, как только происходит событие MouseClick.

Теперь нужно, чтобы при срабатывании события MouseClick сначала происходила проверка, а потом, если она пройдена, открывалось дополнительное окно. Если она не пройдена, то должно открываться окно с сообщением «Доступ запрещён!!!».

Для того, чтобы из обработчика вызвать другие обработчики, которые будут зависеть от каких-то условий, необходимо создать дополнительные обработчики в форме. Для этого используется встроенный компонент, который называется *Команда*.





13.Перейдите к типу Valve. Из Библиотеки компонентов (раздел Встроенные) перетяните сюда элемент *Команда*. Назовите его Action\_AccessDenied (Эта команда будет вызвана, если проверка не будет пройдена).

14.Выделите в Структуре объекта Команду *Action\_AccessDenied*, перейдите во вкладку События, щелкните ПКМ по событию Invoked → Добавить обработчик → Открыть в диалоговом окне → AccessDenied\_Form.

15.Из Библиотеки компонентов (раздел Встроенные) перетяните сюда элемент **Команда**. Назовите ero Action\_OpenPopup (эта команда будет вызвана в случае успешной проверки). Здесь необходимо настроить обработчик открытия нового окна. Чтобы снова все ссылки не прокидывать выделите в Структуре объекта прямоугольник **Valve**, перейдите во вкладку События, раскройте список события MouseClick, скопируйте имеющийся обработчик и вставьте его в событие Invoked команды **Action\_OpenPopup**.

16.Выделите в Структуре объекта прямоугольник *Valve*, перейдите во вкладку События, удалите имеющийся обработчик в событии MouseClick. Добавьте новый обработчик → Выполнить код. Введите код в соответствии с изображением.

Структура объекта			8 ×	🖾 Исходный код
Имя	Описание		A 1	<pre>1 if (unit.Global.OpenPopup_Token.Value)</pre>
<ul> <li>Valve</li> <li>Графические объекты</li> <li>Данные</li> </ul>	Тип на основе Г	Ірямоугольник		<pre>2 { 3 Action_OpenPopup.Invoke();</pre>
События Q. Имя MouseMove V MouseClick Handler_4	Характеристики	Обработчик Выполнить код	<i>₿</i> × Редактировать	<pre>4 } 5 else 6 { 7 Action_AccessDenied.Invoke(); 8 } 200%.</pre>
Редактор свойств События				<ul> <li>О Ошибок ▲ ОПредупреждений () О Сообщений</li> <li>Описание</li> <li>Языкс От ▼</li> </ul>

17.Сохраните проект, запустите его в RunTime. Залогиньтесь под Диспетчером, при нажатии на задвижку, открывается форма управления. Залогиньтесь под Оператором. При нажатии на задвижку открывается форма AccessDenied\_Form.

**Самостоятельное задание.** Необходимо осуществить проверку доступа для Датчиков. Диспетчер может открывать форму управления датчиками, Оператор – нет.

# 10. Резервирование

Зачастую на разных уровнях автоматизации резервируют различные системы. Резервирование по сути повышает надёжность проекта. Проект должен быть безопасным, надёжным и предсказуемым в том числе и с точки зрения мониторинга и управления технологическим процессом.

Для осуществления резервирования с точки зрения контроллера (то есть когда на каком-то объекте стоит контроллер, и, чтобы в случае выхода из строя этого контроллера продолжалась автоматизация управления технологическим процессом) обычно на предприятиях ставят резервные контроллеры, которые также работают с этим же оборудованием и могут общаться между собой, чтобы понимать, у кого на данный момент будет управление. То есть они могут быть в режиме полного дублирования функционала, либо в режиме резервирования, когда они вместе работают с одним и тем же набором оборудования, но при этом распределяют роли между собой.

#### Резервирование источников

В этом случае необходимо добавить ещё один компонент, который будет резервировать тот самый вычислитель, с которым по умолчанию работает сервер.





1. Откройте DeveloperStudio. Через Обозреватель решений перейдите в пространство Domain.

2. Из Панели элементов перетяните в *Domain* элемент *Компьютер*, назовите его PLC\_Reserve.

3. Перейдите внутрь *PLC\_Reserve*, выделите *EthernetAdapter*, в его свойстве Адрес введите IP (для

примера укажем IP-адрес несуществующей машины). В свойстве Сеть укажите <u>Ethernet</u>.

Можно добавить для примера ещё один *EthernetAdapter* с другим адресом.

Demo.Domain, PLC Reserve	🔹 🗒 EthernetAdapter Адаптер Eth	ernet
	Общие	
📳 EthernetAdapter 📑	Адрес	1.2.3.4
1.2.3.4	Сеть	Ethernet
	Има	EthernetAdapte

4. Для того, чтобы сервер понимал, что это новое устройство не просто так существует, а нужно для резервирования внутри *PLC\_Reserve* разместите *Резервный исполняющий компонент* (из Панели элементов). В свойстве Исполняющий компонент укажите, что он будет резервировать <u>Imitator</u>, находящийся в PLC.

🗔 Обозреватель решений	- 4 ×	PLC_Reserve* や × Demo ユ		🔗 Свойства			
Поиск	P -	Demo.Domain. PLC Reserve :		RedundantRuntime Резервный исполняющий компонен-			
Панель элементов	- 4 ×			Общие			
Поиск	<i>ρ</i> -	00					
🔅 Исполняющий компонент		🚱 RedundantRuntime 💷 🧼 🥑	1.2.3.4	Имя	RedundantRuntime		
() Резервный исполняющий ког	мпонент				🍰 Укажите путь к элементу		
() Дублирующий исполняющий	і компонен				Поиск		
🖳 Адаптер Ethernet			4.3.2.1		A I PLC		
Папка обмена					Imitator		
Последовательный порт							
Gepвер истории Microsoft SQ	L Server						
间 Сервер истории PostgreSQL							
Система ПОРТАЛ							
🔄 Мастер ПК							

5. Постройте решение, перейдите к Мастеру развёртывания и примените конфигурацию к линуксовому серверу.

Вот и всё, резервирование контроллеров настроено. То есть достаточно указать, где находится резервное устройство и какую программу оно выполняет. В случае, если произойдёт разрыв с источником, сервер должен знать, что есть второй точно такой же. А переключение между контроллерами программируется внутри самих контроллеров.

Посмотреть информацию о том, с каким контроллером идёт обмен, можно посмотреть через *OpcExplorer* в ветке сервисных сигналов Service.

Также у станции могут быть резервные каналы связи.

6. Для добавления резервного канала связи перейдите в *Компьютер PLC* при помощи Обозревателя решений.

7. Внутрь *Компьютера PLC* из Панели элементов перетяните ещё один *EthernetAdapter*, в свойстве Адрес укажите IP-адрес несуществующей машина (в качестве примера), в свойстве Сеть укажите <u>Ethernet</u>.





🗔 Обозреватель решений	- 4 ×	PLC* + ×		😤 Свойства	
Поиск	P - [	Demo.Domain. PLC Komn	ьютер 🛋 🛛 Demo.omx	EthernetAdapter1 Адаптер	Ethernet
🕜 Панель элементов	- a ×			🔿 Общие	
Поиск	P -	👒 Imitator 🖃 😽	🖤 EthernetAdapter 📑	Адрес	2.2.2.2
🗔 Исполняющий компонент		ModbusTcpSlave	10.0.2.16	Сеть	Ethernet
Резервный исполняющий ко	омпонент	lecTm104Slave		Имя	EthernetAdapter1
() Дублирующий исполняющи	й компонен		2.2.2.2	1)	
Папка обмена			5 S		
Последовательный порт					
🕖 Сервер истории Microsoft SC	QL Server				
🕖 Сервер истории PostgreSQL					
📖 Система ПОРТАЛ					
Мастер ПК					

Теперь при построении конфигурации у источника будет добавлен ещё один канал, ну и в случае разрыва связи по одному из каналов, связь может быть восстановлена по другому каналу.



8. Постройте решение, перейдите к Мастеру развёртывания и примените конфигурацию к линуксовому серверу.

#### Резервирование серверов

В SePlatformServer поддерживается режим горячего резервирования. Это значит, что у Вас в RunTime поднято 2 сервера с одинаковой конфигурацией, то есть они уже исполняются, они уже запущены, и не нужно тратить время на инициализацию каких-то резервных махинаций. И в случае, если необходимо переключиться, серверу достаточно просто переключить состояние. В этом суть горячего резервирования. То есть исполняющий компонент уже работает. Не нужно будет тратить время в случае разрыва соединения, выхода из строя или отключения питания. Резервный сервер сразу готов подхватить работу.

Для организации резервирования серверов необходимо *SePlatformServer* развернуть на разных машинах. В этом и есть суть резервирования. Вы резервируете полноценные среды исполнения. Соответственно резервная пара не может быть развёрнута внутри одной машины. И в качестве резервной машины мы будем использовать машину с OC Windows.

Перед осуществлением резервирования серверов удалите связи с несуществующими *EthernetAdapter*.







1. Перейдите в **Узел learning00** (виндовая машина). Удалите имеющийся здесь сервер.

2. Перетяните сюда из Панели элементов **Резервный SePlatform.Server**. В свойстве Резервируемый сервер необходимо указать, какой сервер он будет резервировать (AstraServer). В разделе свойств Основной канал внутри Адаптер основного укажите <u>EthernetAdapter Linux машины</u>, Адаптер резервного – <u>EthernetAdapter Wndowsмашины</u>.

#### [скриншот из DevStudio]

Обратите внимание, у этого резервного сервера теперь одна и та же конфигурация с *AstraServer*, тот же самый набор модулей

Чтобы резервирование работало, был заранее отключён брэндмауэр и все **Узлы SePlatform.Domain** названы в соответствии с именами машин.

Резервирование Systeme Platform-серверов настроено. Теперь необходимо настроить развёртывание, чтобы

SePlatform.Domain мог развернуть конфигурацию резервного сервера.

3. Перейдите в **SePlatform.domain.agent.XML** (C:\Program Files\SePlatform\SePlatform.Domain). В строке с тэгом **Component** необходимо ввести имя исполняющего компонента, который находится внутри узла Wndowsмашина: RedundantServer.

4. Сохраните SePlatform.domain.agent.XML, перезапустите службы SePlatform.Net. Agent и SePlatform.Domain.Agent через Диспетчер задач.





# 5. Вернитесь в *DeveloperStudio*. Постройте решение, перейдите к Мастеру развёртывания и примените конфигурацию на обеих машинах.

Для просмотра результата можно воспользоваться *OpcExplorer*. Подключитесь к виндовому и линуксовому OPC UA server, добавьте 4 окна Инспектор (вкладка Проект → Добавить инспектор). Разместите в Инспекторах информацию в соответствии с изображением (в левой части OpcExplorer данные по линуксовому источник, справа – по виндовому).

🕰 Инспектор 3 - OpcExplorer									
Файл Вид Проект Справка									
🔾 🔾 🔹 🛝 10.0.2.14	🕼 Astra_UA 🔹	• 🚳 🍇 • 🖣	$\mathbf{P} = \mathbf{A} \mid 0$	) 🕕 🖲	0				
<b>**</b> Обозреватель	Инспектор 1 та ×				Ли	нспектор 4 ½ ×			
Поиск	Тип Сигнал	Значение	Источник		Тип	Сигнал	3	начение Источник	
<ul> <li>В Astra_UA на 10.0.2.14</li> <li>В AutomigOpcUaServer на learning00</li> </ul>	1 root.Sensors.Sensor 1.Press	sure 40	<u>Astra UA</u>		٩	root.Sensors.Sensor 1.Pressu	ure 4	40 <u>AutomiqOpcUaServer</u>	
	B root.Sensors.Sensor 1.Setin	nit True	Astra UA		0	root.Sensors.Sensor 1.SetImi	it 1	rue AutomiqOpcUaServer	
	i root.Sensors.Sensor_1.State	Failure False	Astra_UA		0	root.Sensors.Sensor_1.State.	Failure I	alse <u>AutomiqOpcUaServer</u>	
	10 root.Sensors.Sensor 1.State	Normal True	Astra_UA		0	root.Sensors.Sensor_1.State.	Normal 1	rue <u>AutomiqOpcUaServer</u>	
	1 root.Sensors.Sensor 1.State	Warning False	Astra_UA		0	root.Sensors.Sensor_1.State.	Warning I	alse <u>AutomiqOpcUaServer</u>	
	/ Инспектор 5 та ×				SV	Інспектор 3 та ×			
	Тип Сигнал	Значение	Описание	Источник	Тип	Сигнал	Значение	Описание	Источник
	Service.State.Server	True Состоян	ие сервера	Astra UA	0	Service.State.Server	False	Состояние сервера	AutomigOpcUaServer
	O Service.State.Server.Set	False		Astra UA	0	Service.State.Server.Set	False		AutomiqOpcUaServer
	Service.Redundancy.Switch	False Сигнал ј	резервного переход	a <u>Astra UA</u>	0	Service.Redundancy.Switch	False	Сигнал резервного перехода	AutomigOpcUaServer

В режиме резервирования резервные сервера могут быть в двух состояниях: в работе или в резерве. Сервер, который находится в работе, опрашивает поле, отправляет команды, генерирует события и отправляет историю. Сервер, который находится в резерве, поле не опрашивает, команды не отправляет, не генерирует события и историю не сохраняет. Его задача – это от основного сервера через канал репликации данных (через который они взаимодействуют между собой) получать актуальные данные. Соответственно, у вас есть рабочий сервер, который производит всю работу и есть резервный, который на подхвате. И в случае резервного перехода у резервного сервера уже будут актуальные данные, потому что он их получал с основного сервера. И после того, как произошел резервный переход, резервный сервер уже будет иметь актуальные данные, ему не нужно будет тратить время ни на загрузку, ни на общий опрос. Он просто будет продолжать работу. И буквально доли секунды нужно для того, чтобы всё это сработало.

За состояние сервера для того, чтобы определить, какой именно находится в работе, а какой в резерве, отвечает сигнал Service.State.Server. А за осуществление резервного перехода отвечает сигнал Service.Redundancy.Switch. Этот сигнал можно отправлять с любого сервера, не важно, в работе он или в резерве.

Для проверки попробуйте отследить значение давление датчика при отправлении сигнала резервного перехода или при осуществлении перезапуска рабочего сервера.





# 11. Работа с SePlatform.AccessPoint

SePlatform.AccessPoint выполняет роль конечной точки доступа к оперативным данным и событиям множества источников данных. Это точка доступа, которая консолидирует данные с разных машин. Вы к нему подключаетесь, и он сразу предоставляет данные с того сервера, который находится в работе. То есть через AccessPoint вы видите актуальные данные и можете отправлять управляющие воздействия. Он сам будет определять, с какими серверами нужно работать, будет сам передавать и собирать данные с рабочей машины. Вы к нему подключаетесь и видите уже актуальную информацию. Для того, чтобы использовать AccessPoint, нам нужно его сначала сконфигурировать.

1. Откройте *DeveloperStudio*, при помощи Обозревателя решений перейдите в пространство **Domain**. Из Панели элементов перетяните сюда **Рабочее место** (WorkStation). Элемент **Рабочее место** характеризует рабочие АРМы.

Вы можете один раз настроить DeveloperStudio и разворачивать конфигурацию сколько угодно раз.

2. Перейдите внутрь *WorkStation*.

Обратите внимание, здесь есть клиентский *EthernetAdapter*, он отличается от *EthernetAdapter*, который был в *узлах* и в *компьютерах* тем, что у него нет IP адреса. Это клиентская машина, это она цепляется к источникам, кто к ней будет цепляться, ей неважно. Учитывая то, что рабочее место может быть сконфигурировано один раз, а развёрнуто раз 10, IP-адрес не указывается. Здесь указывается только принадлежность к сети. *Access.Point* и все сервера, с которых он собирает данные, должны находиться в одной сети.

Выберите в свойстве сеть клиентского EthernetAdapter – Ethernet.

Workstation* + ×	👻 🔗 Свойс	гва
Demo.Domain. Workstation	Ethern	etAdapter Клиентский ада
	🔿 Общи	e
EthernetAdapter	Сеть	Ethernet
Клиентский адаптер Ethernet	Имя	EthernetAdapter

3. Внутрь *WorkStation* перетяните из Панели элементов *SePlatformAccessPoint*.





🗔 Обозреватель решений	- 4 ×	Workstation* += X
Поиск	P -	📝 <u>Demo.Domain. Workstation</u> Рабочее место 🖷 Demo.omx
🖓 Панель элементов	<b>→</b> ‡ ×	· · · · · · · · · · · · · · · · · · ·
Поиск Клиентский адаптер Ethernet Папка обмена Последовательный порт Аlpha.AccessPoint	0	AccessPoint ДаServer ДаServer НubModule НdaServer

*AccessPoint* не может быть развернут на центральном узле, поэтому мы будем его разворачивать на машине с OC Linux (в нашем случае она дочерняя).

4. Перейдите внутрь *AccessPoint*. Здесь уже добавлены модули DA, AE, HDA. Удалите модули DA и HDA (на OC Linux они не работают). Из Панели элементов (раздел OPC) перетяните *OPC UA Cepbep*. В его свойстве Имя точки подключения введите <u>AccessPoint UA</u>. В свойстве Порт OPC TCP протокола введите <u>62543</u> (чтобы не было конфликтов с OPC UA, расположенным в AstraServer).



5. Внутрь AccessPoint перетяните из Панели элементов **Модуль истории** (раздел Служебные модули) для того, чтобы AccessPoint мог работать с историей, но базы добавлять не нужно. О базах, с которыми нужно будет связываться, AccessPoint узнает после подключения.



6. Для связи AccessPoint с SePlatform.HMI нужно добавить коммуникационный модуль TCPсервер из Панели элементов, в его свойствах укажите номер TCP-порта: <u>4387</u> для оперативных данных, и номер TCP порта в настройках доступа к истории: <u>4951</u>.







С коммуникационной точки зрения модули настроены. Теперь необходимо настроить то, какие данные будет транслировать *AccessPoint*.

7. Внутрь AccessPoint из Панели элементов перетяните Приложение. Назовите его Application\_AP.



8. Перейдите внутрь **Application\_AP**. Здесь нужно указать, какая ветка будет транслироваться. Перетяните сюда из Обозревателя решений объект **root**, который находится в **AstraServer** (как ссылку). Удалите в названии нижнее подчёркивание для того, чтобы в *SePlatform.HMI* не менять пути у источников и элементов AP.



9. Добавьте для объекта **root** внутри **Application\_AP** атрибут Om/Server → Attributes → Раскрывать ссылку динамически. Этот атрибут используется для того, чтобы динамически транслировать дерево сигналов, которое есть у источника.





Application_AP* + X	🝷 🚰 Свойства	
Demo.Domain.Workstation.AccessPoint, Application AP Приложени	е 🗐 🔹 🍓_гоот Объектная ссылка	
Caroot =	(*) Общие Объект	astra.AstraServer.AstraApplication.root
Ссылка на astra.AstraServer.AstraApplication.root	Исполняющий компонент Только чтение	Нет
Sensors	Аспектная	Her
	<ul> <li>Фтрибуты</li> </ul>	1001
	👍_root Объектная ссылка	
	Атрибут	Значение
	🏽 🎎 Выберите типы атрибутов	×
	Фильтр по имени           4 {} Om.Server           4 {} Attributes           Image: Packpulsata cclum           Image: Packpulsata cclum <td>р – ытия по ссылке у динамически</td>	р – ытия по ссылке у динамически

AccessPoint настроен, осталось настроить развёртывание.

10.Откройте **SePlatform.domain.agent.XML** (C:\Program Files\SePlatform\SePlatform.Domain). После закрытия тэга

Server вставьте строки, описанные ниже:

<Domain>

<WorkstationRoles>

<WorkstationRole Name="Workstation">

<KnownWorkstations>

<KnownWorkstation

SePlatformNetPath="ChildNode" Description="Точка доступа на астралинуксовой машине"/>

</KnownWorkstations>

</WorkstationRole>

</WorkstationRoles>

<ConfigurationCache Path="C:/temp"/>

</Domain>

11. Сохраните SePlatform.domain.agent.XML.

12.Через Диспетчер задач перезапустите службы SePlatform.Net.Agent и

SePlatform.Domain.Agent, посмотрите информацию в журнале EventLogViewer.

13.Откройте PuTTY. Для установки *SePlatform.AccessPoint* убедитесь, что Вы находитесь в папке с установочными файлами и введите команду sudo dpkg –i SePlatform.accesspoint \*\*\*\*\*\*\*.deb.

14.Перейдите в Midnight Commander при помощи команды <mark>sudo mc</mark>, откройте SePlatform.domain.agent.XML нажатием клавиши F4 (/opt/SePlatform/SePlatform.Domain).





15.Для того, чтобы привязаться к службе, после строки с описанием службы на ОС Linux введите строку, описанную ниже:

<SePlatform.Server Name="Server\_Workstation" ServiceName="SePlatform-accesspoint.service" />

16.После закрытия тэга **Server** вставьте строки, описанные ниже:

<Workstation Dynamic="false">

<WorkstationRoles>

<WorkstationRole Name="Workstation">

<Components StoragePath="/usr/local/DomainStorage/cache/server">

<Component InstalledName="Server\_Workstation"

Name="AccessPoint"/>

</Components>

</WorkstationRole>

</WorkstationRoles>

</Workstation>

>
>
> <workstation dynamic="false"></workstation>
> <workstationroles></workstationroles>
> <workstationrole name="Workstation"></workstationrole>
><> <components storagepath="/usr/local/DomainStorage/cache/server"></components>
<pre>&gt;&lt;&gt; <component installedname="Server Workstation" name="AccessPoint"></component></pre>
>
>
>

17.Сохраните SePlatform.domain.agent.XML нажатием клавиши F2, перейдите к командной строке нажатием клавиши F10.

18. Перезапустите службы SePlatform.Net.Agent и SePlatform.Domain.Agent, посмотрите их статус:

sudo systemctl restart SePlatform.Net.service sudo systemctl status SePlatform.Net.service sudo systemctl restart SePlatform.Domain.service sudo systemctl status SePlatform.Domain.service

19. Откройте *DevStudio*, постройте решение, перейдите к Мастеру развертывания, примените конфигурацию на все имеющиеся машины.





Mac	гер развертывания						
<b>v</b> 1	Исполняющий компонент	Построенная версия		Активная версия		Стабильная версия	
) <u><u>*</u></u>	Domain						
	learning00						
$\checkmark$	RedundantServer	1.0.0	1	1.0.0.d28 or 26.05.2022 11:32:3	$\checkmark$		
	astra						
$\checkmark$	AstraServer	1.0.0	*	1.0.0.d26 ot 14.07.2022 6:06:52	1	1.0.0.d13 от	\$
0	Workstation						
-	AccessPoint	1.0.0	1	1.0.0.d28 or 26.05.2022 11:32:3	1		
۵	Точка доступа на астрал	инуксовой маши	не				
	AccessPoint	1.0.0	1	1.0.0.d28 or 26.05.2022 11:32:3	1		

20. Посмотреть результат при переключении работы с одного сервера на другой можно в *OpcExplorer*. Подключитесь к OPC UA серверу AccessPoint\_UA и добавьте ещё одно окно с инспектором для *AccessPoint*. Перетяните сюда те же сигналы, что и у виндового и линуксового серверов.

Спецификация	🕑 UA (tcp)
Сервер	AccessPoint_UA
Порт	62543

Также необходимо настроить SePlatformHMI так, чтобы он работал с AccessPoint.

21.Откройте проект в *SePlatform.HMI*. Перейдите в глобальный объект **Global**. Выделите в Структуре объекта главный источник *ApSource\_Main*. Добавьте ему свойства в соответствии с изображением.

Структура объекта				
Имя	Описание <базовый тип не задан>			
<ul> <li>✓ {4} Global</li> <li>✓ Данные</li> </ul>				
🌹 ApSource_Main	Источник АР			
🖏 ApSource_Sensor_Global	Источник АР			
🖏 ApSource_Valve_Global	Источник АР			
🔓 SecurityContext_Global	Контекст безопасности			
Редактор свойств				
۵				
Свойство	Харак Значение			
> 🜀 Хост	R W 10.0.2.14			
> 🚾 Порт	R W 4387			
> 🥶 Порт истории	R W 4951			
Редактор свойств События				

22.Сохраните проект, запустите в RunTime. Всё как работало, так и работает, только теперь через *AccesPoint*.

## 12. Резервирование истории

1. Установите SePlatform. Historian на машине с OC Windows.

Сервер всё пишет сразу в 2 базы. Соответственно, если что-то случилось с одной машиной, то на другой всегда есть актуальная история. Историю можно писать на ту локальную машину, где находится





сам сервер, так и на удалённые машины. В нашем случае история пишется на машину с OC Linux, и мы также можем настроить SePlatformHistorian на машину с OC Windows.

2. Откройте *DevStudio*, перейдите с помощью Обозревателя решений в **Узел** Wndows машины (learning00).

3. Перетяните сюда из Панели элементов **Дублирующий SePlatform.Historian**. В Свойстве Дублируемый сервер выберите базу, которую он будет дублировать.

Для того, чтобы это всё заработало, необходимо указать то же имя этой базе данных внутри SePlatform.Historian.Server.HML.

4. Откройте на машине с OC Windows **SePlatform.Historian.Server.HML** (внутри папки C:\Program Files\SePlatform\SePlatform.Historian). Для отключения DCOM в строку с тэгом **SePlatform.Historian.Server** 

необходимо дописать **EnableDCOM="0"**, тем самым отключив поддержку DCOM.

24 FXAlpha.Historian.Server StatPort="3388" DefaultPrimaryDir="c:\Alpha.Historian\Databases" MaxPoolMemorySize="512" EnableDCOM="0">

5. Спуститесь ниже к тэгу **<Base>**. Перед закрытием данного тэга необходимо раскомментировать строчку для определения базы данных. Удалите комментарий и в атрибуте Alias введите название для данной базы: History\_Database.

102	-	>							
103		<base< th=""><th>Alias="</th><th>History</th><th>Database"</th><th>PreferredCommonCacheLimit="128"</th><th>ActiveStorageDepth="3"</th><th>StorageDepth="365"</th><th>/&gt;</th></base<>	Alias="	History	Database"	PreferredCommonCacheLimit="128"	ActiveStorageDepth="3"	StorageDepth="365"	/>
104	-								

6. Сохраните SePlatform.Historian.Server.HML, перезапустите службу SePlatform.Historian.Server через Диспетчер задач.

7. Откройте *DevStudio*, постройте решение, перейдите к Мастеру развёртывания и примените конфигурацию на все имеющиеся машины.

Теперь сервер пишет историю сразу в 2 базы.

### Заключение

В данном учебном материале описан базовый функционал и порядок действий для создания первого простого проекта. Это лишь малая часть возможностей *SePlatform*. Более подробную информацию о функционале компонентов можно найти в документации. Стоит также учитывать, что компоненты *SePlatform* могут использоваться не только в проектах АСУ ТП. Для углубления знаний о компонентах *SePlatform* рекомендуются курсы:

- Systeme Platform. Расширенный курс разработки (в разработке)
- Systeme Platform. Курс администратора (по запросу)



