


Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет энергетический
Кафедра автоматизации производственных процессов и электротехники
Направление подготовки 15.03.04 – Автоматизация технологических процессов
и производств
Направленность (профиль) образовательной программы Автоматизация
технологических процессов и производств в энергетике

ДОПУСТИТЬ К ЗАЩИТЕ
И.о. зав. кафедрой
 О.В. Скрипко
«26» июня 2024 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: «Имитационное моделирование систем управления на базе
технологии OPC».

Исполнитель

студент группы 041-об

 25.06
(подпись, дата)

Т. В Мухамедияров

Руководитель


профессор, д-р техн. наук

 25.06.2024
(подпись, дата)

О. В. Скрипко

Консультант: по безопасности
и экологичности

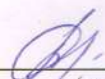
доцент, канд. техн. наук

 19.06.2024
(подпись, дата)

А. Б. Булгаков

Нормоконтроль

профессор, д-р техн. наук

 25.06.2024
(подпись, дата)

О. В. Скрипко


Благовещенск 2024

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет энергетический
Кафедра автоматизации производственных процессов и электротехники

УТВЕРЖДАЮ

и.о. зав. кафедрой


подпись

О.В. Скрипко
И.О. Фамилия

« 26 » мая 2024 г.

ЗАДАНИЕ

К выпускной квалификационной работе студента Мухамедиярова Тимура Валерьевича

1. Тема выпускной квалификационной работы: Имитационное моделирование систем управления на базе технологии OPC

(утверждена приказом от 22.05.24 № 1334-уч)

2. Срок сдачи студентом законченной работы (проекта) 26.04.2024

3. Исходные данные к выпускной квалификационной работе: 1) Приказ об утверждении темы бакалаврской работы; 2) Материалы, собранные в ходе практики.

4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов):

- 1) Описание технологии OPC;
- 2) Описание стандарта OPC UA;
- 3) Разработка имитационной модели;
- 4) Безопасность и экологичность.

5. Перечень материалов приложения:

Лист 1: Методика выполнения лабораторной работы №1

Лист 2: Методика выполнения лабораторной работы №2

Лист 3: Блок-схема алгоритм программы автоматического управления

Часть 1

Лист 4: Блок-схема алгоритм программы автоматического управления

Часть 2

Лист 5: Принципиальная схема стенда управления задвижкой

Лист 6: Методика выполнения лабораторной работы №3

6. Консультанты по выпускной квалификационной работе:

Безопасность и экологичность – Булгаков А.Б., доцент, канд. техн. наук

7. Дата выдачи задания 05.02.2024

Руководитель выпускной квалификационной работы:

О. В. Скрипко, профессор, д-р техн. наук

(фамилия, имя, отчество, должность, ученая степень, ученое звание)

Задание принял к исполнению (дата): 05.02.2024

Мест

(подпись студента)

РЕФЕРАТ

Бакалаврская работа содержит: 76 страниц, 63 рисунка, 20 источников

OPC UA, OPC DA, OPC, OLE, МИКРОКОНТРОЛЛЕР, ПРОТОКОЛ ОБМЕНА ДАННЫМИ, ИМИТАЦИОННАЯ МОДЕЛЬ

В ходе выполнения бакалаврской работы был проведён сбор информации о стандартах OPC, в частности о стандарте OPC UA и его преимуществах по сравнению со стандартом OPC DA. Были разработаны три лабораторных работы, с помощью которых студенты смогут ознакомиться с особенностью OPC UA серверов, а также с созданием конфигурационных файлов сервера в программе Modbus Universal Master OPC server и созданием проектов в клиентском приложении в MasterSCADA 3.7. В ходе подготовке материала для написания лабораторной работы была разработана блок-схема алгоритм программы автоматического управления и принципиальная схема стенда управления задвижкой (в ходе выполнения одной из лабораторных работ студенты должны будут подключиться к реальному объекту по протоколу Modbus. Были разработаны три файла-конфигурации OPC UA, содержащие 14 различных тегов, а также три проекта MasterSCADA 3.7, один из которых содержит полноценный проект операторской панели и программу автоматического управления задвижкой. Создана виртуальная локальная сеть.

СОДЕРЖАНИЕ

Введение	8
1. Описание технологии OPC	9
1.1 Обзор стандартов OPC	9
1. Описание стандарта OPC UA	11
2.1 Область применения стандарта OPC UA	11
2.2 Серверы OPC UA и клиенты OPC UA	15
2.3 Службы (методы и интерфейсы) OPC UA	19
3. Разработка имитационной модели	25
3.1 Выбор потенциальных клиентов и OPC UA серверов для разработки имитационных моделей	25
3.2 Установка связи с OPC UA сервером по протоколу	
MODBUS	27
3.3 Установка связи со стендом. создание визуализации в masterSCADA и написание программы автоматизированного управления	34
3.4 Установка связи с удалённым OPC UA сервером посредством сети интернет	66
4 Безопасность и Экологичность	69
4.1 Безопасность	69
4.2 Экологичность	70
4.3 Чрезвычайные ситуации	71
Заключение	74
Библиографический список	75

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей бакалаврской работе использованы ссылки на следующие стандарты и нормативные документы:

ГОСТ 2.104-68 ЕСКД Основные надписи

ГОСТ 2.105-95 ЕСКД Общие требования к текстовым документам

ГОСТ 2.106-96 ЕСКД Текстовые документы

ГОСТ 2.111-68 ЕСКД Нормоконтроль

ГОСТ 2.316-68 ЕСКД Правила нанесения на чертежах надписей, технических требований и таблиц

ГОСТ 2.605-68 ЕСКД Плакаты учебно-технические. Общие технические требования

ГОСТ 2.701-84 ЕСКД Схемы. Виды и типы. Общие требования к выполнению

ГОСТ 2.721-74 ЕСКД Обозначения условно-графические в схемах.

Обозначения общего применения

ГОСТ 3.1103-83 ЕСКД Основные надписи

ГОСТ 3.1130-93 ЕСКД Основные требования к формам и бланкам документов

ГОСТ 3.1105-84 ЕСКД Правила оформления документов общего назначения

Стандарт ОРС UA

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

OPC – Open Platform Communications, открытая межплатформенная коммуникация.

OPC DA – OPC Data Access (доступ к данным OPC).

OPC UA – OPC Unified Architecture (унифицированная архитектура OPC).

OPC HDA – OPC History Data Access (доступ к истории изменения данных).

OPC DX – OPC Data eXchange (изменение данных по протоколу Ethernet).

OPC AE – OPC Alarms & Events (OPC тревоги и события).

OPC UA Binary DataEncoding – OPC UA декодировка двоичных данных.

SCADA - supervisory control and data acquisition (диспетчерское управление и сбор данных).

MES – manufacturing execution system (система управления производственными процессами).

COM – Component Object Model (объектная модель компонентов).

DCOM – Distributed COM (Распределённый COM).

XML – eXtensible Markup Language (расширяемый язык разметки).

SOAP – Simple Object Access Protocol (протокол доступа к объектам).

HTTP – HyperText Transfer Protocol (протокол передачи гипертекста).

HTTPS – HTTP Secure (безопасность HTTP)

ВВЕДЕНИЕ

После того, как в 2008-ом году компания OPC FOUNDATION опубликовала стандарт OPC UA, навыки работы с OPC DA серверами стали устаревать. Это не произошло мгновенно, так как компания озаботилась о возможности плавного перехода, ведь стандарт OPC UA является не столько новой разработкой, сколько совмещением всех стандартов обмена данными между клиентами и серверами, описанных в предыдущих спецификациях данной компании с несколькими важными нововведениями, главными из которых является возможность запуска на сервера на аппаратуре, отличной от компьютера с операционной системой windows и независимость от стандарта Microsoft DCOM. Благодаря этому, оставалась возможность импортировать данные из OPC DA серверов и клиентов в OPC UA формат и даже настроить обмен данными между клиентами и серверами разных стандартов.

Однако, с каждой модернизацией программно-аппаратурного обеспечения, возрастает востребованность в специалистах, знакомых со спецификой OPC UA серверов, возрастает, в то время как навыки обращения с OPC DA архитектурой теряют актуальность.

Стандарт OPC UA более универсален, он разрабатывался с целью заменить все стандарты OPC разработанные в прошлом, не только OPC Data Access, но и OPC Data eXchange и OPC Batch, OPC UA сервера менее зависимы от операционной системы устройств и лишены уязвимостей, существовавших в прошлых спецификациях. Грамотный производитель просто не видит причин использовать устаревший стандарт.

Поэтому, необходимо предоставить студентам навыки работы и знания о работе с OPC UA серверами.

1 ОПИСАНИЕ ТЕХНОЛОГИИ OPC

1.1 Описание технологии OPC

OPC (Open Platform Communications, ранее OLE for Process Control) – набор программных технологий для обмена данными, разработанный фирмой OPC Foundation. Целью внедрения данных стандартов является унификация процедуры обмена данными между элементами SCADA системы или MES систем, в частности, как протокол передачи данных от программируемого логического контроллера на элементы Человеко-Машинного Интерфейса.

Использование унифицированных протоколов взаимодействия между устройствами позволяет снизить затраты на создание и эксплуатацию приложений промышленной автоматизации.

Под общим названием OPC выпущена целая группа стандартов, каждый из которых описывает набор функций определенного назначения. В данный момент широко используются следующие стандарты:

- OPC DA (Data Access) – устаревший, но всё ещё наиболее распространённый стандарт, описывающий функции обмена данными в реальном времени с программируемыми логическим контролерами, распределёнными системами управления и другими устройствами. В данный момент медленно замещается стандартом OPC UA, обладающим рядом заметных преимуществ.
- OPC HDA (Historical Data Access) – стандарт, предоставляющий операции взаимодействия с архивированными данными.
- OPC AE (Alarms & Events) – стандарт, описывающий функции уведомления о тех или иных событиях, в частности об аварийных ситуациях, действиях оператора и передающий информационные сообщения.

- OPC Batch – разработанная на основе стандарта S88.01 спецификация, предоставляющая функции шагового и рецептурного управления технологическим процессом
- OPC DX (Data eXchange) – стандарт, используемый для создания шлюзов обмена между устройствами и программами по сети Ethernet, а также регулирующий некоторые другие функции при данном виде связи.
- OPC XML-DA (XML-Data Access) – стандарт, регулирующий обмен XML/текст сообщения по протоколам SOAP и HTTP.
- OPC Security – стандарт, определяющий методику предоставления доступа клиентов к данным системы управления посредством OPC-сервера и накладываемые при этом ограничения.
- OPC UA (Unified Architecture) – новейший из стандартов OPC и единственный из всего множества стандартов, не требующий наличия операционной системы Windows. В то время, как каждая из предыдущих спецификаций OPC определяла уникальную модель адресного пространства и свой собственный набор служб, OPC UA объединяет все существующие доселе модели в единое адресное пространство с единым набором методов взаимодействия с данными.

2 ОПИСАНИЕ СТАНДАРТА OPC UA

2.1 Область применения стандарта OPC UA

OPC UA – платформонезависимый стандарт обмена данными между различными видами систем и устройств по различным типам сетей с использованием различных протоколов связи. Данный стандарт обладает полным функционалом интегрированных служб, реализованных в предыдущих спецификациях OPC, а также отличается повышенной защищённостью, поскольку основывается на разработках OPC Foundation, а не на открытом стандарте DCOM, а следовательно, не имеет его уязвимостей. OPC UA позволяет серверам предоставлять клиентам определения типов для объектов, к которым осуществляется доступ из адресного пространства. Это позволяет использовать информационные модели для описания содержимого адресного пространства. OPC UA позволяет отображать данные в различных форматах, включая двоичные структуры и документы XML. Формат данных может быть определен OPC, организациями по разработке стандартов или поставщиками. Через адресное пространство клиенты могут запрашивать у сервера метаданные с описанием формата данных. Клиенты, не имеющие заранее запрограммированных знаний о форматах данных, могут определять форматы во время выполнения и правильно использовать данные. Тип данных, используемый для обмена, может быть определён пользователем. В свою очередь, сервера определяют модели объектов, с которыми будут взаимодействовать клиенты. Данный протокол позволяет предоставлять доступ как к текущим данным, так и к архивным, сигналам тревог и событиям.

В данной спецификации поддерживаются неиерархические взаимодействия между узлами, в зависимости от запроса и прав клиента, сервер может предоставить данные в различных конфигурациях. Столь высокая вариативность позволяет использовать данную спецификацию в широком спектре областей. [1] Как показано на рисунке 1, протокол OPC UA может применяться

как в человеко-машинном интерфейсе, так и в SCADA системе, интерфейсах ПЛК, а также описывать адресное пространство и методы для высокоуровневых операций.

Это означает, что данный стандарт применяется для производственного программного обеспечения в таких областях применения, как периферийные устройства, системы управления, системы управления производственными процессами и системы планирования ресурсов предприятия, т.е. всех систем для обмена информацией и использования команд и управления производственными процессами.

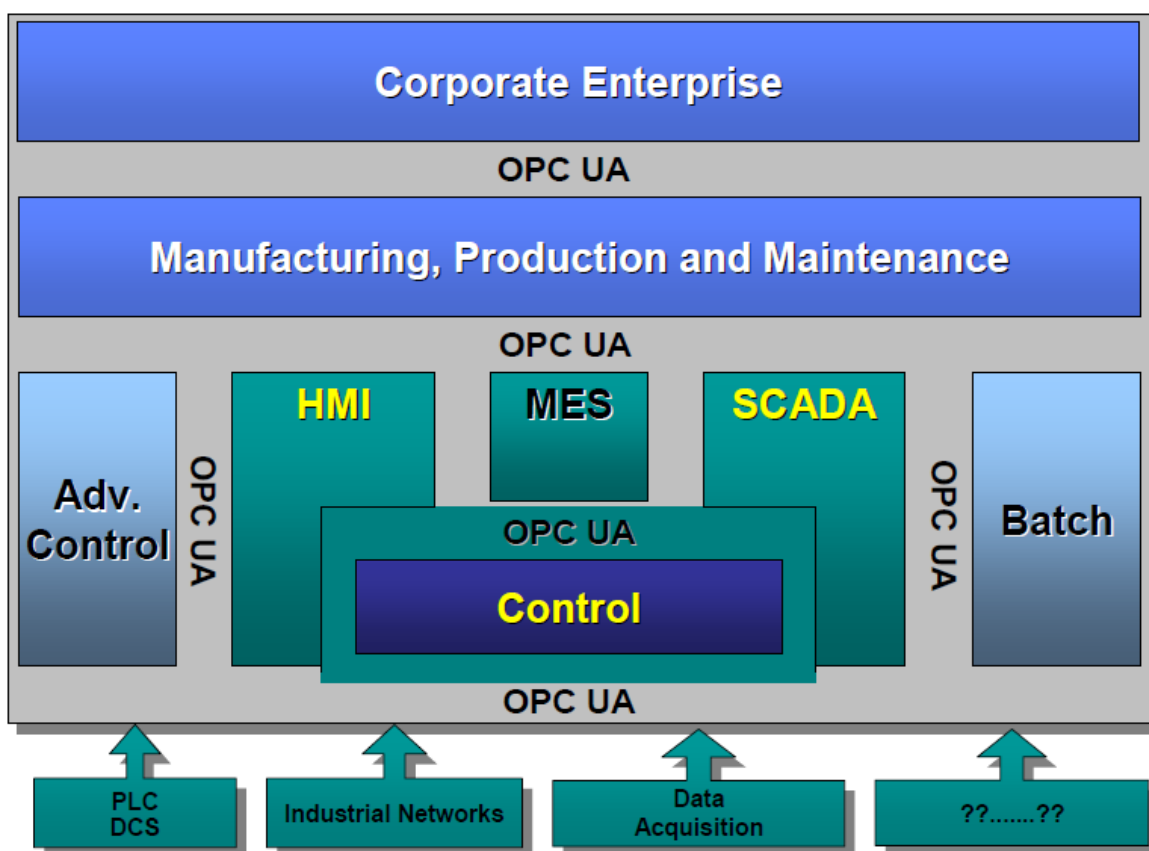


Рисунок 1- Области применения стандарта OPC UA

Следствием столь широкой распространённости – данный стандарт регулирует обмена данных на всех уровнях, от взаимодействия с отдельным ПЛК в производственных сетях до межсерверного взаимодействия в корпоративных сетях – стала некоторая избыточность функциональности, а потому и настраиваемость. OPC UA предоставляет серверу все функциональные

возможности, которые могут понадобиться, а сервер, в свою очередь, выбирает профиль, подмножество возможностей, которые он будет исполнять. Клиенты способны взаимодействовать с сервером посредством данных профилей. [2]

Так как OPC UA – многоуровневая спецификация, для отделения основной структуры от низкоуровневых вычислений было решено определить два типа данных: XML/текст и OPC UA Binary DataEncoding.

Последний является форматом данных, используемый для обмена информацией по алгоритмам, описанным в стандарте OPC UA. При разработке стандарта основное внимание уделялось возможности быстро закодировать и декодировать полученную информацию, однако также учитывался объём информации, передаваемой по проводу.

OPC UA Binary DataEncoding основывается на нескольких информационных примитивах с четко определенными правилами кодирования, которые могут быть последовательно записаны в binary stream или считаны из него. [3] кодирование структуры осуществляется путем последовательной записи закодированной формы каждой ячейки. Если в данной ячейке располагается другая структура, то значения ее полей записываются последовательно перед записью следующего поля в содержащую структуру. В binary stream должны быть записаны все структуры, даже те, которые содержат нулевые значения.

OPC UA Binary DataEncoding не содержит никакой информации о типе или имени поля, поскольку предполагается, что все приложения OPC UA будут заранее знать об используемых сервисах и структурах. Исключением является ExtensionObject, который предоставляет идентификатор и размер для структурированного типа данных, которую он представляет. Это позволяет декодеру пропускать неизвестные ему типы данных.

В качестве протоколов транспортного уровня используются: SOAP/HTTP, HTTPS и OPC UA TCP.

Поскольку и клиенты, и сервера поддерживают различные методы кодирования данных и протоколы их передачи, пользователь может настраивать процедуру обмена, в зависимости от того, что ему сейчас важнее: производительность или совместимость с веб-службами. Как правило типы данных разработанные OPC Foundation обладают наилучшей производительностью, однако требует отдельных приложений для взаимодействия с веб-службами.

Одним из преимуществ стандарта OPC UA является его совместимость с предыдущими стандартами OPC. Данные, предоставляемые COM-серверами OPC DA, OPC HDA и OPC A&E могут быть без проблем использованы OPC UA. Пользователи могут как перенести данные из COM серверов OPC на OPC UA, так и использовать внешние оболочки для преобразования из OPC UA в OPC COM и обратно.

Другим весомым преимуществом спецификации OPC UA является защищённость. Хотя сама спецификация и не определяет обстоятельства, при которых необходимы определённые механизмы защиты – она определяет механизмы и параметры защиты. Защищённость OPC UA связана прежде всего с аутентификацией клиентов и серверов, а также с отказом от использования стандарта DCOM. [4]

При установлении сессии, пользователь обязан пройти одноразовую аутентификацию, чтобы получить доступ к данным, хранящимся на сервере. Однако сам по себе стандарт OPC UA не описывает методы прохождения аутентификации, особенности аутентификации могут широко варьироваться в зависимости от реализованных приложений.

Защита на транспортном уровне связана прежде всего с использованием закрытого стандарта шифрования, из-за этого невозможно использовать уязвимости, имеющиеся в открытом стандарте DCOM. Это важно, так как старые стандарты, такие как OPC DA возможно было использовать только в частных сетях. Связь же по протоколу OPC UA может, с относительной

надёжностью осуществляться даже через интернет. В таком случае единственной уязвимостью будет OPC UA Binary DataEncoding.

В случае, если злоумышленник воспользовался данной уязвимостью, возможно посмотреть список операций в журнале аудита. Поддержка Журнала Аудита всегда была неотъемлемой особенностью стандартов OPC UA.

2.2 Серверы OPC UA и клиенты OPC UA

Адресное пространство OPC UA представляет из себя множество объектов, узлов, связанных ссылками. Клиенты могут получить доступ к соответствующим узлам посредством интерфейсов и методов OPC UA.

Каждый узел обладает атрибутами OPC.

Базовые атрибуты, присущие всем узлам сервера, определяют именование, классификацию и идентификацию. Отдельные классы узлов имеют собственные атрибуты, в которых хранится информация, специфичная для каждого узла.

Узлы на сервере OPC UA располагаются согласно иерархической структуре, верхние уровни которой одинаковы для любого сервера OPC UA. [5] Однако, как уже было сказано выше, данная спецификация поддерживает неиерархические взаимодействия между узлами, а потому, каждый из узлов может иметь в себе ссылки на другие узлы. Таким образом, помимо иерархического адресного пространства, данные на сервере могут быть представлены в виде полной ячеистой сети или любой другой конфигурации.

На рисунке 2 изображены основные элементы сервера OPC UA и их взаимосвязь.

Под Реальными Объектами на данной схеме подразумеваются физические или программные объекты, которые доступны серверному приложению OPC UA или которые оно обслуживает внутри. Примеры включают физические устройства и диагностические счетчики.

Элементы мониторинга – это объекты на сервере, созданные клиентом и отслеживающие узлы адресного пространства. Как только произойдёт из-

менение данных, находящихся в узле – или любой другое событие, на которое клиент осуществил подписку – клиент будет оповещён об изменении данных на сервере.

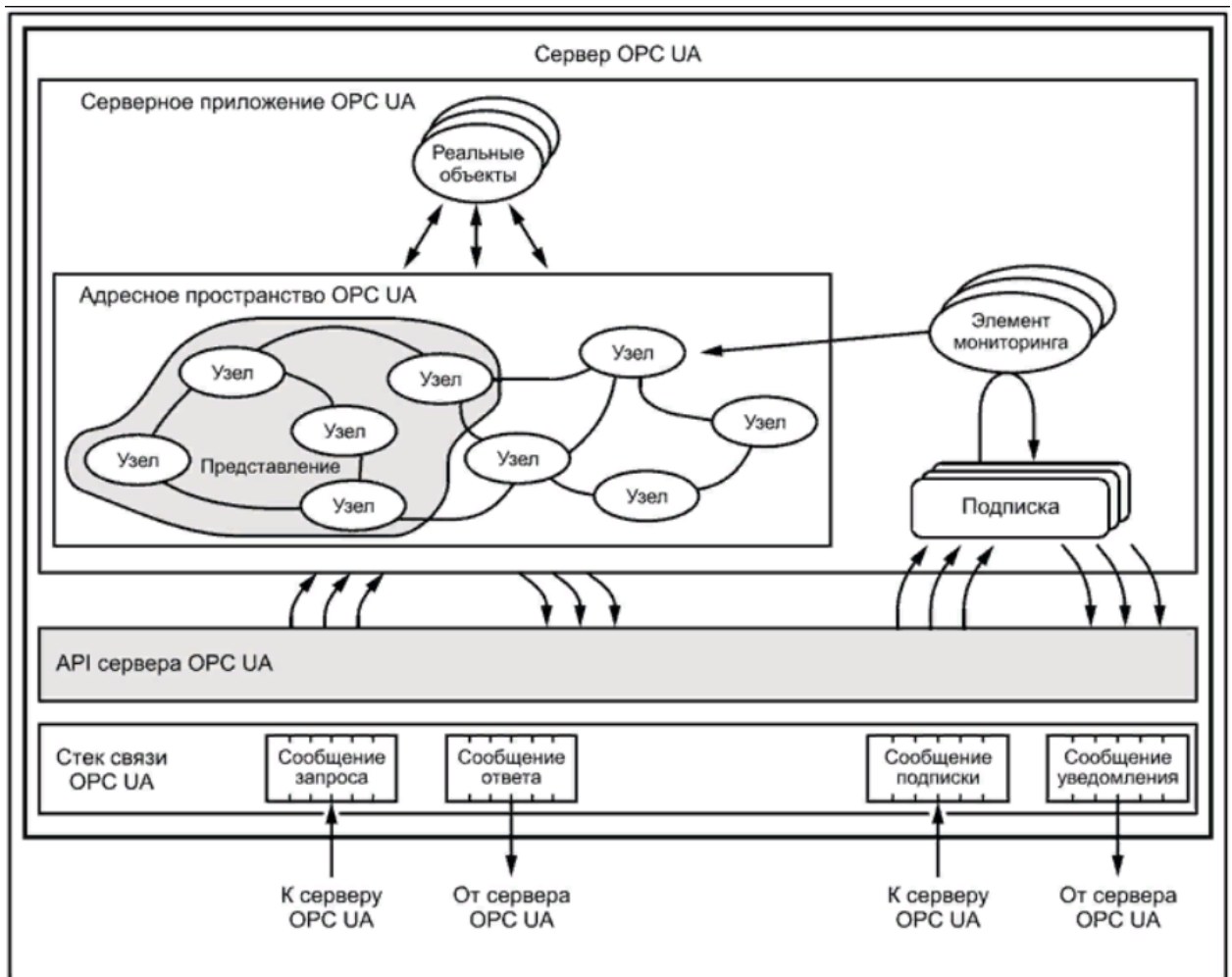


Рисунок 2 - Архитектура сервера OPC UA

Подписка представляет собой конечную точку на сервере, публикующую уведомления для клиента.

Службы запроса/ответа – это службы, вызываемые клиентом через интерфейс службы OPC UA для выполнения определенной задачи на одном или нескольких узлах в адресном пространстве и для возврата ответа.

Спецификация OPC UA допускает в том числе и межсерверное взаимодействие. В таком случае один из серверов выступает клиентом другого сервера. [6]

Поддержка межсерверных взаимодействий позволяет создавать сети с одноранговым взаимодействием серверов. Таким образом могут быть реализованы сети с избыточными серверами или сети, включающие в себя удалённые серверы.

Таким образом, возможно поддерживать горячую архивацию данных и гарантировать, что даже в случае происшествий, заканчивающихся выводом из строя одного из серверов, второй – сохранит все данные и сможет поддерживать производственный процесс.

Альтернативным применением будет объединение серверов в многоуровневую архитектуру. Такой подход позволит:

- 1) Агрегировать данные с серверов нижнего уровня.
- 2) Предоставлять клиентам высокоуровневую структуру данных.
- 3) Предоставлять одному клиенту доступ к нескольким серверам.

На рисунке 3 представлено объединение серверов OPC UA для вертикального доступа к данным на предприятии.

Взаимодействие между клиентом и сервером возможно только при активной сессии (сессия – логическое соединение между клиентом и сервером). После прохождения обязательной аутентификации, клиент не обязан проходить повторные проверки, сессия не будет прервана. Также, сессия не будет прервана при сбоях связи, поскольку сами по себе сессии независимы от базовых протоколов связи. Единственными условиями прерывания сессии являются запросы клиента, запросы сервера, а также бездействие клиента в течении определённого времени, обговоренного при создании сессии.

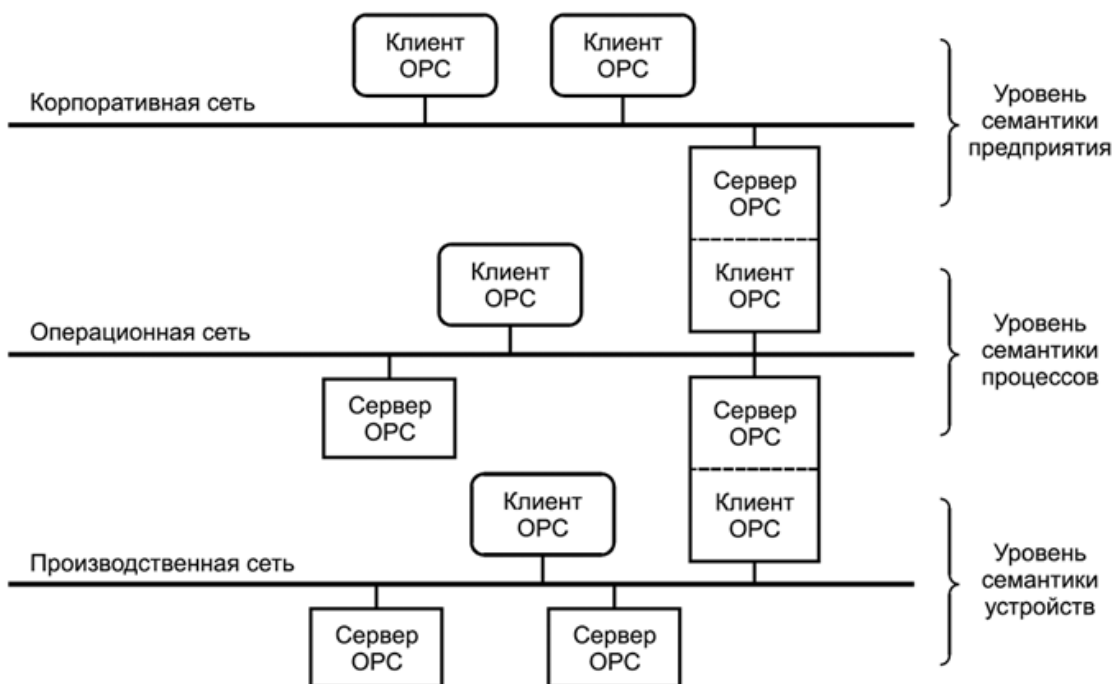


Рисунок 3 – Многоуровневая архитектура серверов

В зависимости от имеющихся ресурсов, сервер может ограничивать число единомоментно осуществляемых сессий.

Клиентская архитектура OPC UA моделирует конечную точку клиента во взаимодействиях клиента и сервера. На рисунке 4 показаны основные элементы типичного клиента OPC UA и их взаимосвязь.

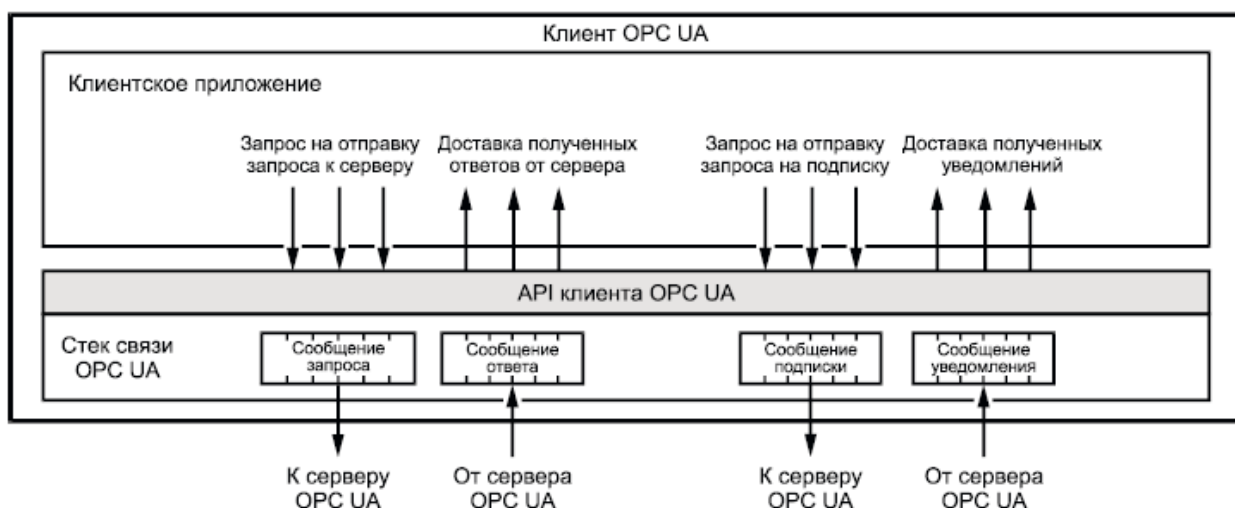


Рисунок 4 – Структура клиента OPC UA

Клиентское приложение – это код, который реализует функцию клиента. Приложение использует API клиента OPC UA для отправки и получения запросов служб OPC UA и ответов на сервер OPC UA.

API клиента OPC UA - внутренний интерфейс, который изолирует код клиентского приложения от стека связи OPC UA. Стек связи OPC UA преобразует вызовы API клиента OPC UA в сообщения и отправляет их через базовую сущность связи на сервер по запросу клиентского приложения. Стек связи OPC UA получает ответ и сообщения уведомления от базовой сущности связи и доставляет их клиентскому приложению через API клиента OPC UA.

2.3 Службы (методы и интерфейсы) OPC UA

Для доступа к хранящимся на сервере данным клиент обязан использовать методы и интерфейсы OPC UA. Совокупность процедур, направленных на получение данных с сервера – представляет собой интегрированную службу OPC UA. [7]

Данные службы подразделяются на множество наборов служб, каждая из которых предназначена для получения строго определённого подмножества данных с сервера и доступа к определённым узлам.

2.3.1 Службы обнаружения:

Службы обнаружения – набор служб, используемые для обнаружения доступных серверов OPC UA. Набор также обеспечивает клиентам способ чтения конфигурации защищенности для подключения к серверу. Службы обнаружения реализуются отдельными серверами и выделенными серверами обнаружения. Известные выделенные серверы обнаружения предоставляют клиентам возможность обнаружить все зарегистрированные серверы OPC UA.

2.3.2 Набор служб защищённого канала.

Набор служб защищенного канала определяет службы для открытия канала связи, который обеспечивает конфиденциальность и целостность всех сообщений обмена с сервером.

Защищенный канал – это долговременное логическое соединение между одним клиентом и одним сервером. Защищенный канал поддерживает набор ключей, которые известны только клиенту и серверу и используются для аутентификации и шифрования сообщений, отправляемых по сети. Службы защищенного канала позволяют клиенту и серверу безопасно договариваться об используемых ключах.

Приложения UA используют стек связи для обмена сообщениями. Службы защищенного канала используются для установления защищенного канала между двумя стеками связи, что обеспечивает безопасный обмен сообщениями. Приложения UA используют набор служб сессии для установления сессии приложения UA.

2.3.3 Набор служб сессии.

Набор служб сессии – службы для установления соединения на прикладном уровне в контексте сессии от имени конкретного пользователя.

2.3.4 Набор службы управления узлами

Набор служб управления узлами позволяет клиентам добавлять, изменять и удалять узлы в адресном пространстве. Службы управления узлами предоставляют интерфейс для конфигурирования серверов.

2.3.5 Набор служб представлений

Представления являются общедоступными, создаваемыми сервером подмножествами адресного пространства. Представлением по умолчанию является адресное пространство целиком, и службы представлений могут работать со всем адресным пространством.

Набор служб представлений позволяет клиентам обнаруживать узлы в представлении путем просмотра. Просмотр позволяет клиентам перемещаться вверх и вниз по иерархии или следовать ссылкам между узлами в представлении. Таким образом, просмотр позволяет клиентам обнаруживать структуру представления.

2.3.6 Набор служб запросов.

Набор служб запросов позволяет пользователям получать доступ к адресному пространству без просмотра и без знания логической структуры для внутреннего хранения данных.

Запросы позволяют клиентам выбирать подмножество узлов в представлении на основе критериев фильтра, предоставляемых клиентом. Узлы, выбранные в представлении по результатам запроса, называются набором результатов.

Обработка запросов, требующих доступа к данным во время исполнения, таким как данные устройства, включает ресурсоемкие операции или значительные задержки и может вызвать трудности. В этих случаях сервер может отклонить запрос.

2.3.7 Набор служб атрибутов

Набор служб атрибутов используется для чтения и записи значений атрибутов. Атрибуты являются простейшими характеристиками узлов OPC UA. Атрибуты не могут быть определены клиентами или серверами и являются единственными элементами в адресном пространстве со значениями данных. Атрибут значения используется для определения значения переменных.

2.3.8 Набор служб методов

Методы представляют вызовы функций объектов [8]. Методы вызываются и возвращаются после завершения независимо от результата. Время выполнения методов может варьироваться в зависимости от выполняемой ими функции.

Набор служб методов определяет средства для вызова методов. Метод всегда является компонентом объекта. Обнаружение осуществляется через службы просмотра и запроса. Клиенты обнаруживают методы, поддерживаемые сервером, путем просмотра имеющихся объектов, которые идентифицируют поддерживаемые методы.

Поскольку методы могут контролировать некоторые аспекты производственной работы, вызов метода может зависеть от условий окружающей среды или других условий. Это может быть особенно актуально при попытке повторно вызвать метод сразу после завершения выполнения. Условия вызова метода могут еще не вернуться в состояние, позволяющее повторный запуск метода. Методы могут поддерживать одновременные вызовы или иметь один вызов, выполняемый в данный момент времени.

2.3.9 Набор служб элементов мониторинга

Набор служб элементов мониторинга используется клиентом для создания и обслуживания элементов мониторинга. Элементы мониторинга отслеживают переменные, атрибуты и уведомители событий. Элементы мониторинга генерируют уведомления при обнаружении определенных условий, а также отслеживают переменные на предмет изменения значения или статуса, атрибуты на предмет изменения значения и уведомители событий на предмет созданных отчетов о сигналах предупреждения и событиях.

Каждый элемент мониторинга идентифицирует элемент для мониторинга и подписку для периодической публикации уведомлений для клиента. Каждый элемент мониторинга определяет скорость мониторинга (измерения) элемента, а для переменных и уведомителей событий - критерии фильтрации для генерации уведомления

Частота измерения элемента мониторинга может быть выше, чем частота публикации подписки. По этой причине элемент мониторинга может быть настроен на очередь всех уведомлений или на очередь только последних уведомлений для передачи по подписке. В последнем случае размер очереди равен единице.

Службы элементов мониторинга определяют режим мониторинга. Режим мониторинга может быть настроен так, чтобы отключить измерения и отчетность, включить только измерения или включить измерения и отчетность. Каждый образец оценивается для определения того, должно ли гене-

рироваться уведомление. При необходимости генерации уведомление ставится в очередь. Если включена отчетность, то очередь становится доступной подписке для передачи.

Элементы мониторинга могут быть настроены для запуска отчетов других элементов мониторинга. В этом случае режим мониторинга элементов отчета устанавливается только на измерение, и когда инициирующий элемент генерирует уведомление, любые уведомления в очереди для элементов отчета становятся доступными подписке для передачи.

2.3.10 Набор служб подписки

Набор служб подписки используется клиентом для создания и обслуживания подписок. Подписки – это сущности, которые периодически публикуют сообщения уведомления для назначенного им элемента мониторинга. Сообщение уведомления содержит общий заголовок, за которым следует серия уведомлений. Формат уведомлений зависит от типа элемента мониторинга (т.е. переменных, атрибутов и уведомлений о событиях)[9]

После создания существование подписки не зависит от сессии клиента с сервером. Это позволяет одному клиенту создать подписку, а второму, возможно резервному клиенту, получать от нее сообщения уведомления.

Подписки включают функции, которые поддерживают обнаружение и восстановление потерянных сообщений. Каждое сообщение уведомления содержит порядковый номер, который позволяет клиентам обнаруживать пропущенные сообщения. Если в течение интервала поддержания активности нет уведомлений для отправки, то сервер отправляет сообщение подтверждения активности, которое содержит порядковый номер следующего отправленного сообщения уведомления. Если клиенту не удастся получить сообщение по истечении интервала поддержания активности или если он обнаружил пропуск сообщения, то он может запросить у сервера переслать одно или несколько сообщений.

3 РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ

3.1 Выбор потенциальных клиентов и OPC UA серверов для разработки имитационных моделей

При выборе программ, используемых для разработки имитационных моделей и последующего обучения студентов работе со SCADA системами и OPC UA серверами учитывались следующие критерии:

Возможность интеграции в учебный процесс – программы должно быть возможно запустить на персональных компьютерах, находящихся в учебных аудиториях шестого корпуса, и не будут требовать наличия знания, которые будут получены на дальнейших курсах.

Простота освоения – программы должны быть достаточно просты в освоении, чтобы студенты могли без особых сложностей научиться с ней работать за время курса.

Актуальность и востребованность – выбранные программы должны использоваться на реальных производственных объектах в Российской Федерации.

Доступность – в идеале выбранные программы должны обладать бесплатными демоверсиями для обучения.

Универсальность – желательно, чтобы программы были совместима с широким списком контроллеров, а не была оптимизированной под один конкретный.

Основными отечественным производителями SCADA систем являются МПС софт, в данный момент выпустившие уже четыре версии программного пакета MasterSCADA и AdAstrA, разработчики TRACEMOD. Продукция обеих компаний чрезвычайно широко используется на производственных объектах по всей России в странах СНГ, а потому решающим фактором при выборе между ними была возможность интеграции в учебный процесс. Предполагается, что обучение взаимодействию с контроллерами и выполне-

ние лабораторных работ на стендах будет происходить на третьем курсе обучения, освоение же Trace Mod – проходит лишь в первом семестре четвертого курса.

Новейшим и наиболее востребованным из продуктов линейки MasterSCADA является программный пакет MasterSCADA 4D, имеющий ряд преимуществ по сравнению с предшественниками, в частности возможность программирования визуально-функциональных блоков на языке C++. К сожалению, использование данной версии MasterSCADA в учебном процессе не является возможным из-за её системных требований: она требует наличия на компьютере операционной системы Windows 10 или выше, в то время как на всех компьютерах в шестом корпусе стоят операционные системы Windows 7 и Windows XP. Потому, к рассмотрению были представлены программные пакеты MasterSCADA 3X и MasterSCADA 2. Было решено остановиться на третьей версии данной программы, так как она является наиболее новой, а следовательно, и востребованной среди тех, чьё использование возможно в нашем образовательном учреждении.

MasterSCADA 3 была признана Продуктом года по выбору русской редакции авторитетного международного журнала Control Engineering.[10]

В качестве OPC сервера рассматривались два бесплатных приложения: Modbus Universal MasterOPC Server и Owen OPC Server. Оба имеют бесплатные демоверсии и оба приложения были разработаны российскими компаниями – МПС Софт и Овен соответственно, однако, выгодным отличием Modbus Universal MasterOPC Server от Owen OPC Server было то, что он создавался специально для работы с MasterSCADA и при этом обладает более гибкой настройкой параметров. Приложения Owen же оптимизированы исключительно для работы с контроллерами Owen, и работа с неподходящими программами может быть затруднена.

3.2 Установка связи с OPC UA сервером по протоколу MODBUS

При выполнении первой лабораторной работы студенты должны будут научиться работать с Modbus Universal MasterOPC Server, создать и запустить OPC UA сервер, способный установить связь с удалённым компьютером по протоколу Modbus, а также подключить к созданному серверу MasterSCADA 3.

Для организации обмена данными между удалённым компьютером и OPC сервером необходимо, в первую очередь, создать OPC сервер. Для этого требуется запустить программу Modbus Universal MasterOPC Server.

При открытии приложения будут показаны три окна, номером один на рисунке 5 обозначено окно конфигурации, номером два – дерево объектов, номером три – окно параметров.

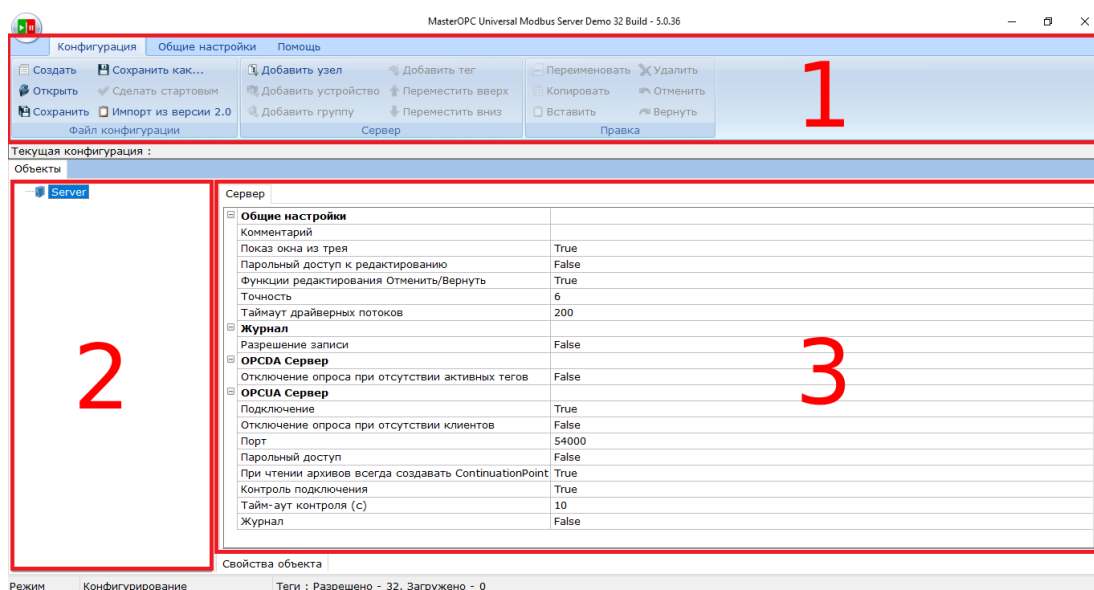


Рисунок 5 – Modbus Universal MasterOPC Server

Нажав правой кнопкой мышки по строке «сервер» в окне объектов и выбрав соответствующую строку во всплывающей подсказке, возможно добавить коммуникационный узел. Сделаем это. После добавления узла необходимо провести его настройку. В строке «Тип Узла» требуется выбрать ячейку «COM», а в настройках COM прописать номер порта, к которому подключён USB-кабель, передающий данные по протоколу MODBUS.

Окно настроек узла с соответствующими параметрами изображены на рисунке 6.

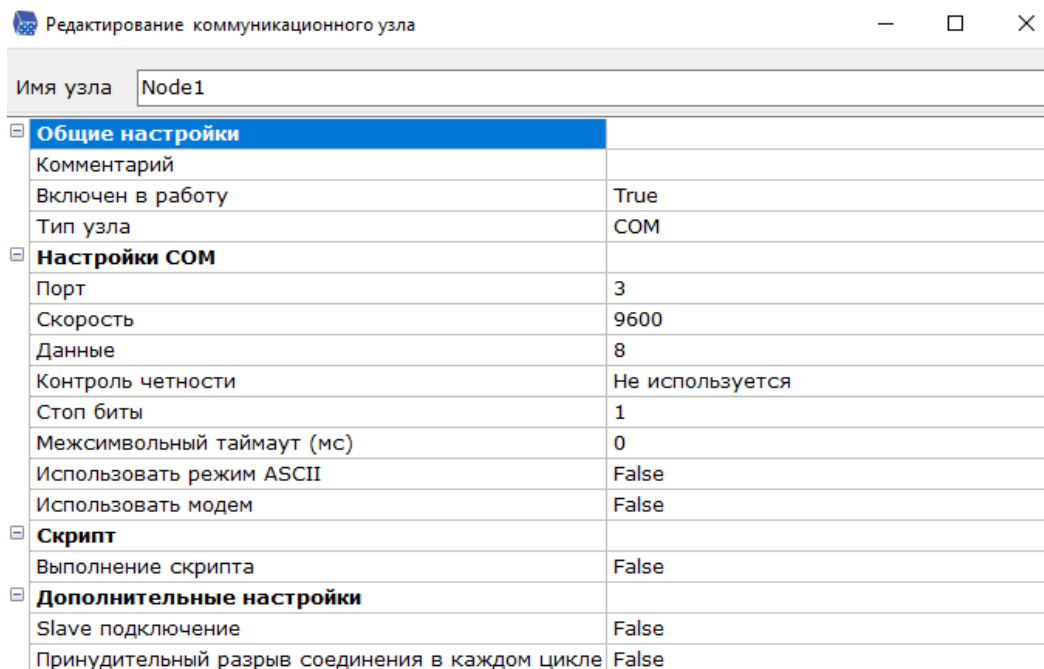


Рисунок 6 – Настройки коммуникационного узла

Для того, чтобы подключить сервер к клиенту необходимо добавить к коммуникационному узлу устройство. Во время настройки, во всплывающем окне, потребуется прописать в строке адрес адреса компьютера, с которым мы устанавливаем обмен данными по протоколу MODBUS.

Настройки устройства изображены на рисунке 7.

Согласно стандарту OPC UA, в адресном пространстве OPC UA сервера находятся узлы. Для того, чтобы приложение было способно проводить опрос переменных – необходимо создать соответствующие теги. Для первой лабораторной работы создадим два тега.

Настройку тега необходимо проводить сразу при создании, поскольку, после того как они будут занесены в сервер, некоторые параметры редактировать будет невозможно.

Запрос, посылаемый клиентом Modbus серверу, состоит из трёх полей: адрес ведомого устройства, код функции и данные. Адрес ведомого устрой-

ства – это адрес, который мы вписали в строку «адрес» при добавлении устройства в коммуникационный узел.

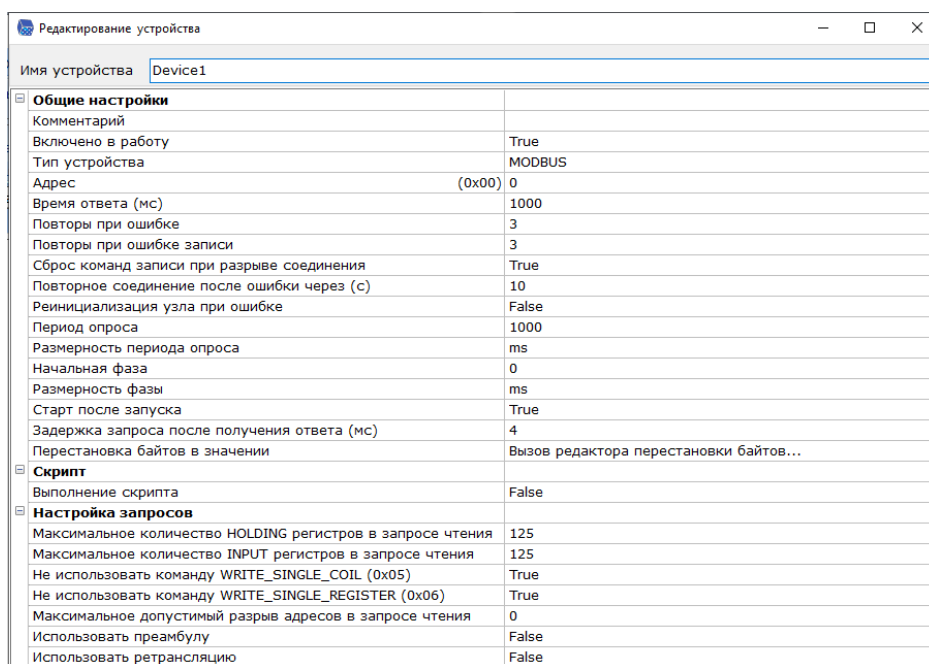


Рисунок 7 – Настройки параметров подключённого по протоколу MODBUS устройства

В окне «редактирование тега», имеется возможность указать код функции, выбрав один из пяти вариантов в графе «Регион» и адрес в сети Modbus в поле адрес. Адрес указывается в десятичной системе счисления, так что если в данной строке пропишите 11, то сервер будет получать данные об изменении переменной с адресом 0x000B.

Первой из созданных переменных будет тег А с адресом 0x0000, расположенный в регионе «HOLDING REGISTERS».

Перед сохранением необходимо удостовериться, что тип данных, передаваемых по протоколу Modbus и тип данных в устройстве – одинаковые. В нашем случае, в программе задана переменная с типом двухбайтовое слово, так что в строке «тип данных в устройстве» необходимо выбрать «int 16».

Для подключения архивации данных в строке HDA доступ необходимо прописать параметр «True».

Строки «Запись по изменению значения тега» и «Автоматическая запись» отвечают за параметры архивации. Если в параметрах первой из вышеназванных строк будет прописано True, то архивация будет происходить по факту изменения значения тега, если же True будет прописано в строке «Автоматическая запись» – архивация данных будет происходить лишь по истечению определённого промежутка времени.

Для данной переменной запретим автоматическую запись и разрешим запись по изменению значения тега. Настройки данной переменной отображены на рисунке 8.

Создадим переменную В с адресом 0x0001, расположенную в регионе «HOLDING REGISTERS».

Для подключения архивации данных пропишите в строке HDA доступ параметр «True».

Имя тега	
A	
Общие настройки	
Комментарий	
Включен в работу	True
Регион	HOLDING_REGISTERS
Адрес (0x0000)	0
Тип данных в устройстве	int16
Тип данных в сервере	int32
Тип доступа	ReadWrite
Использовать перестановку байтов устройства	True
Последний тег в групповом запросе	False
Пересчет (A*X + B)	False
Скрипт	
Разрешение выполнения скрипта после чтения	False
Разрешение выполнения скрипта перед записью	False
Дополнительно	
Извлечение бита из данных	False
Наличие отдельного регистра записи	False
Чтение сразу после записи	False
Сброс команды записи	True
Принудительная запись командой 6	False
HDA	
HDA доступ	True
Количество записей в архиве (100 - 4080)	1000
Автоматическая запись	False
Запись по изменению значения тега	True

Рисунок 8 – Параметры переменной А

Разрешим для данной переменной автоматическую запись и запретим запись по изменению значения тега. Параметры тега В изображены на рисунке 9.

Сохраним изменения и нажмём на круг в левом верхнем углу приложения. Нажмём «Старт». Если всё сделано правильно, то мы увидим список из всех переменных, видимых серверу и изменение их значений.

Редактирование тега	
Имя тега	B
Общие настройки	
Комментарий	
Включен в работу	True
Регион	HOLDING_REGISTERS
Адрес (0x0001)	1
Тип данных в устройстве	int16
Тип данных в сервере	int32
Тип доступа	ReadWrite
Использовать перестановку байтов устройства	True
Последний тег в групповом запросе	False
Пересчет (A*X + B)	False
Скрипт	
Разрешение выполнения скрипта после чтения	False
Разрешение выполнения скрипта перед записью	False
Дополнительно	
Извлечение бита из данных	False
Наличие отдельного регистра записи	False
Чтение сразу после записи	False
Сброс команды записи	True
Принудительная запись командой 6	False
HDA	
HDA доступ	True
Количество записей в архиве (100 - 4080)	1000
Автоматическая запись	True
Запись по изменению значения тега	True

Рисунок 9 – Параметры переменно В

Запустим приложение MasterScada 3.

Создадим новый проект. При открытии нового проекта мы увидим четыре окна. Окно проекта изображено на рисунке 10.

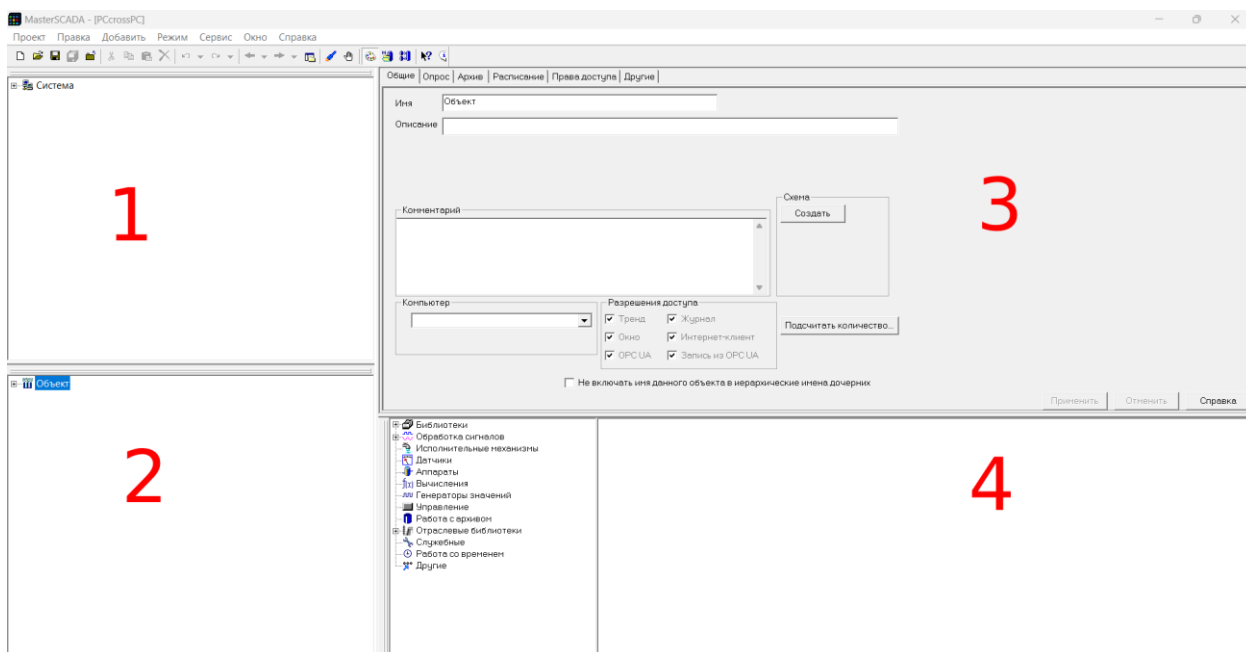


Рисунок 10 – Стартовая страница MasterSCADA

Первое окно является деревом системы, второе – деревом объектов, третьей – окном параметров, четвёртым – библиотека функциональных блоков. [11]

Добавим в дерево системы компьютер, а к компьютеру – протокол «OPC UA Server».

Итоговое дерево системы отображено на рисунке 11.

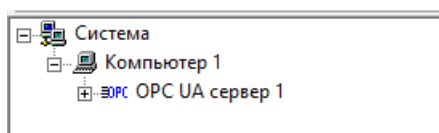


Рисунок 11 – Дерево системы

В окне параметров нажмём на заголовок «настройки», а внутри нажмём на кнопку «настройки». В открывшемся окне пропишем адрес сервера. 127.0.0.1 – это local host, ip адрес данного компьютера. 55000 – это номер

порта. Его можно посмотреть или изменить в приложении Modbus Universal MasterOPC Server в параметрах сервера.

Введём данные параметры в окне настроек OPC UA клиента. Окно настроек OPC UA клиента изображено на рисунке 12.

The screenshot shows the 'Настройки OPC UA клиента' (OPC UA client settings) window with the 'Дополнительно' (Additional) tab selected. The window is divided into several sections:

- Информация о сервере** (Server information): A text field for 'Сервер' (Server) contains 'opc.tcp://127.0.0.1:55000'. To the right is a 'Найти' (Find) button. Below the field are two buttons: '+ Добавить резервный' (Add backup) and '- Удалить резервный' (Remove backup).
- Настройки безопасности** (Security settings): Two dropdown menus. 'Политика безопасности' (Security policy) is set to 'None'. 'Режим безопасности сообщений' (Security mode) is also set to 'None'.
- Настройки аутентификации** (Authentication settings): The 'Анонимно' (Anonymous) radio button is selected. Below it are input fields for 'Имя пользователя' (Username) containing 'MasterSCADAUser' and 'Пароль' (Password), which is currently empty.
- Настройки сессии** (Session settings): An input field for 'Имя сессии' (Session name) contains 'Session 1'.

At the bottom right of the window are three buttons: 'Ок' (OK), 'Отмена' (Cancel), and 'Справка' (Help).

Рисунок 12 – Параметры настройки OPC UA клиента

Перейдём на вкладку «дополнительно», выберем колонку «чтение архива» и поставим галочку в ячейке «получать архивные данные». В колонке режим работы укажем пункт «чтение и подписка».

Последовательно нажмём кнопки «ОК» и «Применить».

Если всё сделано правильно, то в списке параметров высветятся переменные, содержащиеся в сервере. Обновлённое дерево системы изображено на рисунке 13.

В дереве объектов добавим к Объекту Объект, с названием: «Sin».

Выделим объект «Sin», перейдём на вкладку тренды и нажмём кнопку «добавить». Переместим OPC переменную из дерева системы на график тренда.

Переведём SCADA систему в режим исполнения, нажав на иконку с изображением ракеты.

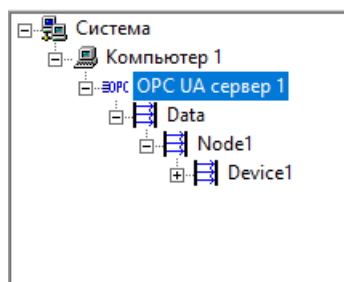


Рисунок 13 – Обновлённое дерево системы

3.3 Установка связи со стендом. создание визуализации в master-SCADA и написание программы автоматизированного управления

При выполнении второй лабораторно работы, студенты должны будут установить обмен данными между ПКП1И стенда, имитирующего управление задвижкой и персональным компьютером. [12] Для этого, как и в первой работе, они должны будут запустить сервер в программе Modbus Universal Master OPC Server, передать данные о тегах в MasterSCADA и написать программу автоматического управления задвижкой, позволяющую, в ответ на ввод целочисленного значения в соответствующем окне визуализации, открыть задвижку на заданную величину.

Запустите программу Modbus Universal MasterOPC Server и создайте новый проект. Дерево системы изображено на рисунке 14.

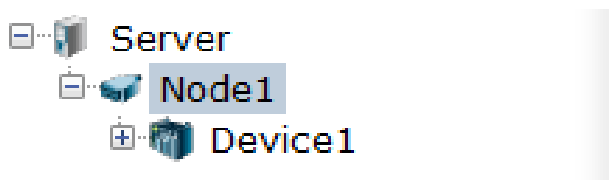


Рисунок 14 – Дерево системы

Настройка коммуникационного узла показана на рисунке 15. При присоединении устройства к коммуникационному узлу в графе адрес пропишите

число 17 (адрес станда), а в графе «не использовать команду WRITE_SINGLE_COIL (0x05)» – false. Настройки параметров устройства показаны на рисунке 16.

Для упрощения работы с тегами, в программе Modbus Universal MasterOPC Server присутствует возможность создания групп тегов. Группы не требуют никакой настройки параметров и единственная их роль в дереве объектов – вынести некое множество тегов в отдельный объект. Создадим подгруппу: «Оперативные» и добавим её тег «% положения задвижки» с параметрами:

Адрес – 0x0007

Регион – INPUT REGISTERS

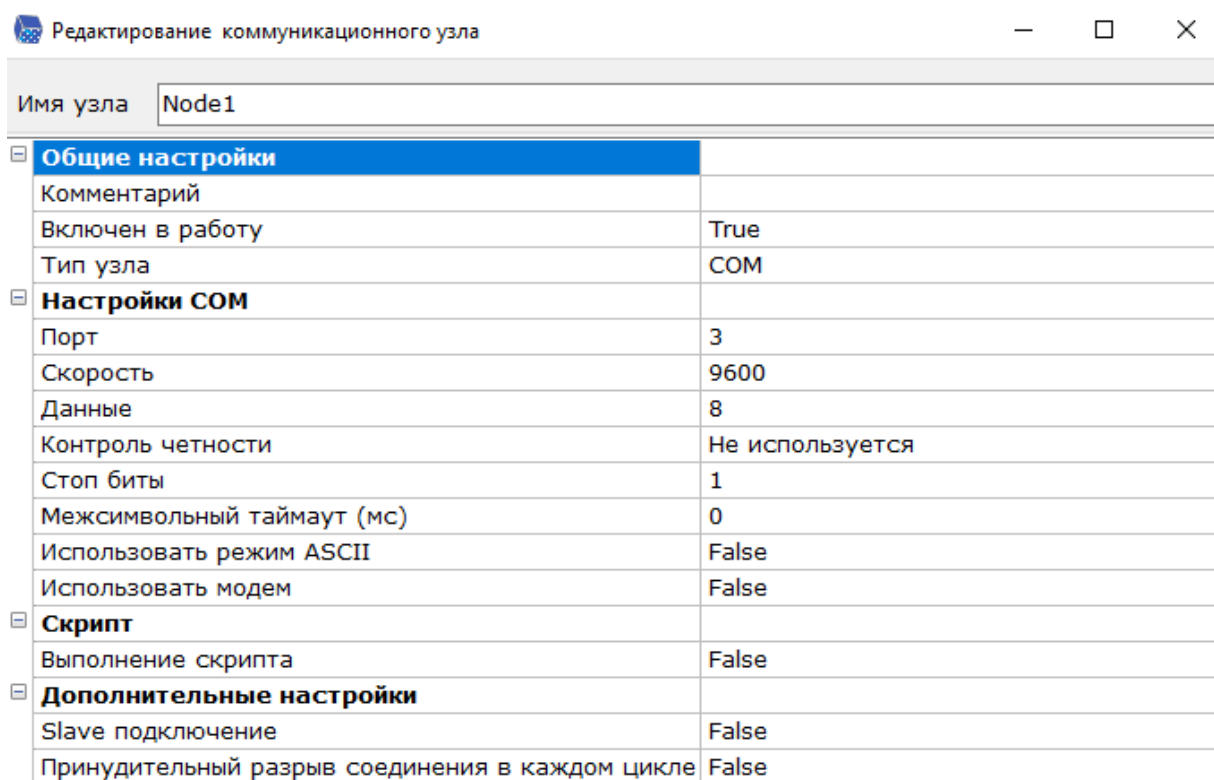


Рисунок 15 – Настройка параметров узла

Имя устройства		Device1
Общие настройки		
Комментарий		
Включено в работу		True
Тип устройства		MODBUS
Адрес	(0x11)	17
Время ответа (мс)		1000
Повторы при ошибке		3
Повторы при ошибке записи		3
Сброс команд записи при разрыве соединения		True
Повторное соединение после ошибки через (с)		10
Реинициализация узла при ошибке		False
Период опроса		1000
Размерность периода опроса		ms
Начальная фаза		0
Размерность фазы		ms
Старт после запуска		True
Задержка запроса после получения ответа (мс)		4
Перестановка байтов в значении		Вызов редактора перестановки байтов...
Скрипт		
Выполнение скрипта		False
Настройка запросов		
Максимальное количество HOLDING регистров в запросе чтения		125
Максимальное количество INPUT регистров в запросе чтения		125
Не использовать команду WRITE_SINGLE_COIL (0x05)		False
Не использовать команду WRITE_SINGLE_REGISTER (0x06)		True
Максимальное допустимый разрыв адресов в запросе чтения		0
Использовать преамбулу		False
Использовать ретрансляцию		False

Рисунок 16 – Настройка параметров устройства

Остальные параметры оставим неизменными. Данный тег будет хранить данные о текущем положении задвижки, величина указывается с точностью до десятой доли процента. На рисунке 17 показано окно настроек данной переменной.

Однако, помимо знания текущего положения задвижки, для построения рабочей SCADA системы необходимо знать и текущее состояние прибора, а именно: открыта задвижка полностью или закрыта, закрывается задвижка или же наоборот, а также иметь представление о текущих ошибках, в том числе об аварии, проскальзывании и перегрузе. Все эти параметры содержатся в переменной «состояние прибора». Добавим в подгруппу «Оперативные» тег «состояние прибора» с параметрами:

Регион – INPUT REGISTERS

Адрес – 0x000A

Параметры правильно настроенного тега «состояние прибора» показаны на рисунке 18

Тег <<INPUT_REGISTERS>> : %Положения задвижки	
Общие настройки	
Комментарий	
Включен в работу	True
Адрес (0x0007)	7
Тип данных в устройстве	int16
Тип данных в сервере	int32
Тип доступа	ReadOnly
Использовать перестановку байтов устройства	True
Последний тег в групповом запросе	False
Пересчет (A*X + B)	False
Скрипт	
Разрешение выполнения скрипта после чтения	False
Дополнительно	
Извлечение бита из данных	False
HDA	
HDA доступ	True
Количество записей в архиве (100 - 4080)	1000
Автоматическая запись	True
Запись по изменению значения тега	False

Рисунок 17 – Параметры настройки тега Положение Задвижки

Тег <<INPUT_REGISTERS>> : Состояние прибора	
Общие настройки	
Комментарий	
Включен в работу	True
Адрес (0x000A)	10
Тип данных в устройстве	int16
Тип данных в сервере	int32
Тип доступа	ReadOnly
Использовать перестановку байтов устройства	True
Последний тег в групповом запросе	False
Пересчет (A*X + B)	False
Скрипт	
Разрешение выполнения скрипта после чтения	False
Дополнительно	
Извлечение бита из данных	False
HDA	
HDA доступ	True
Количество записей в архиве (100 - 4080)	1000
Автоматическая запись	True
Запись по изменению значения тега	False

Рисунок 18 – Параметры настройки тега «Состояние прибора»

Система автоматического управления должна быть способна не только собирать данные о состоянии прибора и обрабатывать данные, но и отправлять команды. Для упрощения работы с данными командами создадим подгруппу «Управление» и внесём в неё переменную «Открыть», расположенную в регионе Coils по адресу 0x0000. Для чтения тегов из региона Coils используется команда F0x05, а потому необходимо убедиться, что при настройке устройства прописали «False» в строке «не использовать команду Write_single_coil (0x05).» Если этого не сделать, то будет использоваться команда «Force Multiple Coils (0x0F)», записывающая значение в несколько регистров флагов. [13], [14], [15], [16], [17]. Правильная настройка данного тега отображена на рисунке 19. Для того, чтобы отдать команду на закрытие задвижки у стенда предусмотрена команда «Закрыть». Добавим тег «Закрыть», расположенную в регионе COILS, по адресу – 0x0001. Окно настроек данного тега отображено на рисунке 20.

Тег <<COILS>> : Открыть

Общие настройки	
Комментарий	
Включен в работу	True
Адрес (0x0000)	0
Тип данных в устройстве	bool
Тип данных в сервере	bool
Тип доступа	WriteOnly
Скрипт	
Разрешение выполнения скрипта перед записью	False
Дополнительно	
Наличие отдельного регистра записи	False
Чтение сразу после записи	False
Сброс команды записи	True
HDA	
HDA доступ	True
Количество записей в архиве (100 - 4080)	1000
Автоматическая запись	True
Запись по изменению значения тега	False

Рисунок 19 – Параметры настройки тега «Открыть»

Тег <<COILS>> : Закрывать

Общие настройки	
Комментарий	
Включен в работу	True
Адрес (0x0001)	1
Тип данных в устройстве	bool
Тип данных в сервере	bool
Тип доступа	WriteOnly
Скрипт	
Разрешение выполнения скрипта перед записью	False
Дополнительно	
Наличие отдельного регистра записи	False
Чтение сразу после записи	False
Сброс команды записи	True
HDA	
HDA доступ	True
Количество записей в архиве (100 - 4080)	1000
Автоматическая запись	True
Запись по изменению значения тега	False

Рисунок 20 – Окно настроек тега «Закрывать»

Для остановки движения задвижки необходимо отправить стенду команду «Стоп», расположенную в регионе COILS по адресу 0x0002. Окно настроек данного тега отображено на рисунке 21.

Тег <<COILS>> : Стоп

Общие настройки	
Комментарий	
Включен в работу	True
Адрес (0x0002)	2
Тип данных в устройстве	bool
Тип данных в сервере	bool
Тип доступа	WriteOnly
Скрипт	
Разрешение выполнения скрипта перед записью	False
Дополнительно	
Наличие отдельного регистра записи	False
Чтение сразу после записи	False
Сброс команды записи	True
HDA	
HDA доступ	True
Количество записей в архиве (100 - 4080)	1000
Автоматическая запись	True
Запись по изменению значения тега	False

Рисунок 21– Окно настроек тега «Стоп»

Последней из команд, созданных в данной группе, будет команда сброс флагов. Запись любого значения в данный тег обнуляет значение тега «Состояние прибора» и выводит стенд из состояния ошибки. В системе автоматического управления она использоваться не будет, однако, если по тем или иным причинам, во время эксплуатации прибора, возникнет внештатная ситуация – у пользователя появится возможность вернуть прибор в рабочий режим. Настройки данного тега отображены на рисунке 22.

Итоговое дерево системы, со всеми внесёнными переменными, отображено на рисунке 23.

Когда все переменные в Сервере созданы, возможно создать SCADA систему и подключить её к серверу в качестве клиента. Для этого запустим программу MasterSCADA 3 и создадим новый проект.

Поскольку это учебный проект, требования о вводе пароля можно игнорировать.

В дерево системы необходимо добавить компьютер и протокол OPC UA сервер. Итоговое дерево системы отображено на рисунке 24.

В окне параметров выберем заголовок «настройки», а в открывшемся вкладке кнопку «настройки». В открывшемся окне необходимо прописать адрес сервера, в нашем случае, поскольку сервер находится на нашем компьютере, это будет 127.0.0.1 – это local host, ip адрес данного компьютера. В окне номер порта пропишем 55000, адрес порта, прописанный при конфигурации сервера. Его можно посмотреть или изменить в приложении Modbus Universal MasterOPC Server в параметрах сервера.

Тег <<COILS>> : Сброс флагов

Общие настройки	
Комментарий	
Включен в работу	True
Адрес (0x0003)	3
Тип данных в устройстве	bool
Тип данных в сервере	bool
Тип доступа	WriteOnly
Скрипт	
Разрешение выполнения скрипта перед записью	False
Дополнительно	
Наличие отдельного регистра записи	False
Чтение сразу после записи	False
Сброс команды записи	True
HDA	
HDA доступ	True
Количество записей в архиве (100 - 4080)	1000
Автоматическая запись	True
Запись по изменению значения тега	False

Рисунок 22 – Окно настроек тега «Сброс Флагов»

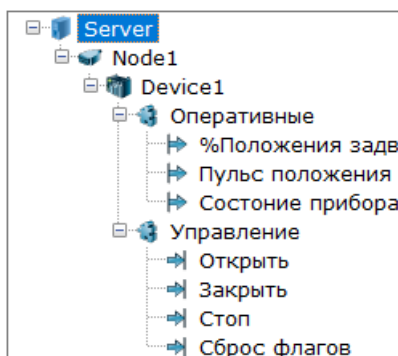


Рисунок 23 – Дерево системы в MODBUS INIVERSAL

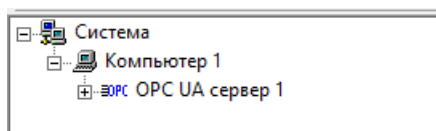


Рисунок 24 – Дерево системы в MasterSCADA

Введём данные параметры в окне настроек OPC UA клиента. Окно настроек OPC UA клиента изображено на рисунке 25.

Перейдём на вкладку «дополнительно», выберем колонку «чтение архива» и поставим галочку в ячейке «получать архивные данные». В колонке режим работы укажем пункт «чтение и подписка». Окно настроек OPC UA клиента с правильно прописанными параметрами отображено на рисунке 26.

The screenshot shows the 'Настройки OPC UA клиента' (OPC UA client settings) window with the 'Дополнительно' (Additional) tab selected. The window is divided into several sections:

- Информация о сервере** (Server information):
 - Сервер (Server): `opc.tcp://127.0.0.1:55000` with a 'Найти' (Find) button.
 - Buttons: '+ Добавить резервный' (Add backup) and '- Удалить резервный' (Remove backup).
- Настройки безопасности** (Security settings):
 - Политика безопасности (Security policy): 'None'.
 - Режим безопасности сообщений (Message security mode): 'None'.
- Настройки аутентификации** (Authentication settings):
 - Radio buttons: 'Анонимно' (Anonymous) is selected.
 - Имя пользователя (Username): 'MasterSCADAUser'.
 - Пароль (Password): empty field.
- Настройки сессии** (Session settings):
 - Имя сессии (Session name): 'Session 1'.

At the bottom right, there are three buttons: 'Ок' (OK), 'Отмена' (Cancel), and 'Справка' (Help).

Рисунок 25 – Параметры настройки OPC UA клиента

Последовательно нажмём кнопки «ОК» и «Применить».

Если всё сделано правильно, то в списке параметров высветятся переменные, содержащиеся в сервере. Обновлённое дерево системы изображено на рисунке 27.

Настройки подключения	Дополнительно
Подписывать за один запрос в количестве	<input type="text" value="100"/> узлов
Максимальное количество узлов на одну подписку	<input type="text" value="10000"/>
Постоянный опрос резервируемых соединений	<input type="checkbox"/>
Использовать кэш	<input type="checkbox"/>
Максимальное количество записей в кэше	<input type="text" value="1000"/>
Тип метки времени при чтении	Время источника ▾
Тип метки времени при записи	Не передавать ▾
Максимальное количество значений в запросе на запись	<input type="text" value="100"/>
Не записывать недостоверные данные	<input checked="" type="checkbox"/>
Запрет использования недостоверных значений при чтении	<input type="checkbox"/>

Настройки узлов	
Показывать атрибуты выделенного узла	<input checked="" type="checkbox"/>
Загружать дочерние элементы у переменных	<input type="checkbox"/>
Максимальный уровень вложенности узлов	<input type="text" value="0"/>
Не загружать узлы с данными именами	<input type="text"/>
Не загружать дочерние элементы у узлов с данными именами	<input type="text"/>

Чтение архивов	
Получать архивные данные	<input checked="" type="checkbox"/>
Режим работы	Чтение и подписк ▾
Период получения архивных данных	<input type="text" value="1000"/> мс
Таймаут чтения архивных данных	<input type="text" value="10000"/> мс
Количество возвращаемых записей	<input type="text" value="1000"/>
Глубина считываемых данных при старте	<input type="text" value="1"/> дней
Макс количество тегов в запросе чтения архива	<input type="text" value="1"/>

Рисунок 26 – Настройки OPC UA клиента

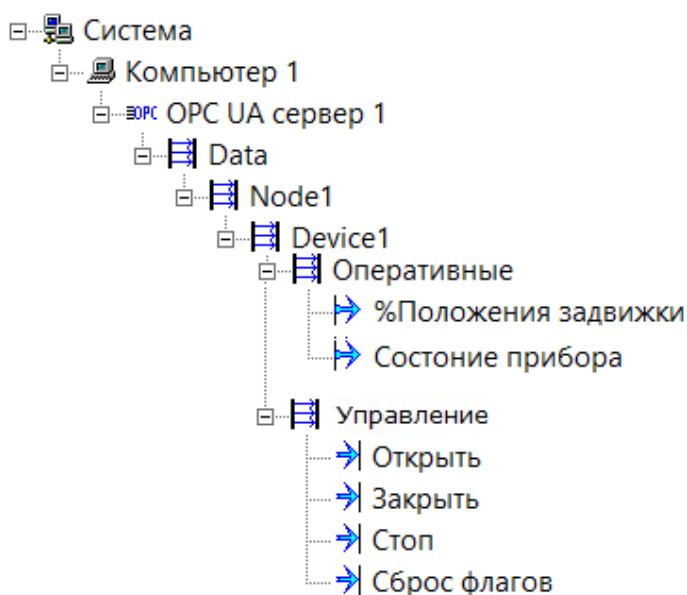


Рисунок 27 – Обновлённое дерево системы

На данном этапе лабораторной работы, студенты должны завершить установку связи между реальным объектом (стендом управления задвижкой, сервером и клиентским приложением, которым, в данном случае, является MasterSCADA 3.7. Им необходимо будет убедиться, что и в сервере, и в MasterSCADA отображаются реальные значения тегов, а также убедиться, что данные из MasterSCADA передаются ПКПИИ. Для этого необходимо будет ввести любое значение в одну из переменных управления – закрыть, открыть и стоп. Если в ответ на ввод тега задвижка начала двигаться, или же наоборот остановилась – значит связь настроена и можно приступать ко второму этапу лабораторной работы, разработке программы автоматического управления задвижкой и созданием проекта операторской панели.

Программа автоматического регулирования должна, в ответ на ввод той или иной величины, переводить задвижку в соответствующее положение. Первоначальный алгоритм программы автоматического управления представлен на рисунке 28.

В ходе разработки лабораторной работы, мы убедимся, что реализация программы автоматического управления в представленном на рисунке виде не представляется возможным, так как всегда существует задержка в отклике. Таким образом невозможно добиться мгновенной остановки задвижки в тот момент, когда она достигает заданного оператором значения. Соответственно, необходимо доработать систему автоматического регулирования с учётом мёртвой зоны, той величины, на которую положение задвижки может отклониться от заданного, не вызывая при этом реакцию программы автоматического управления.

Кроме того, представленный на рисунке 28 алгоритм недостаточно подробен и написание на его основе полноценной программы автоматического управления не представляется возможным. Поэтому, на листах 4 и 5 приложения представлена расширенная версия блок-схемы алгоритма.

Доработанный алгоритм, предусматривающий наличие мёртвых зон представлен на рисунке 29.

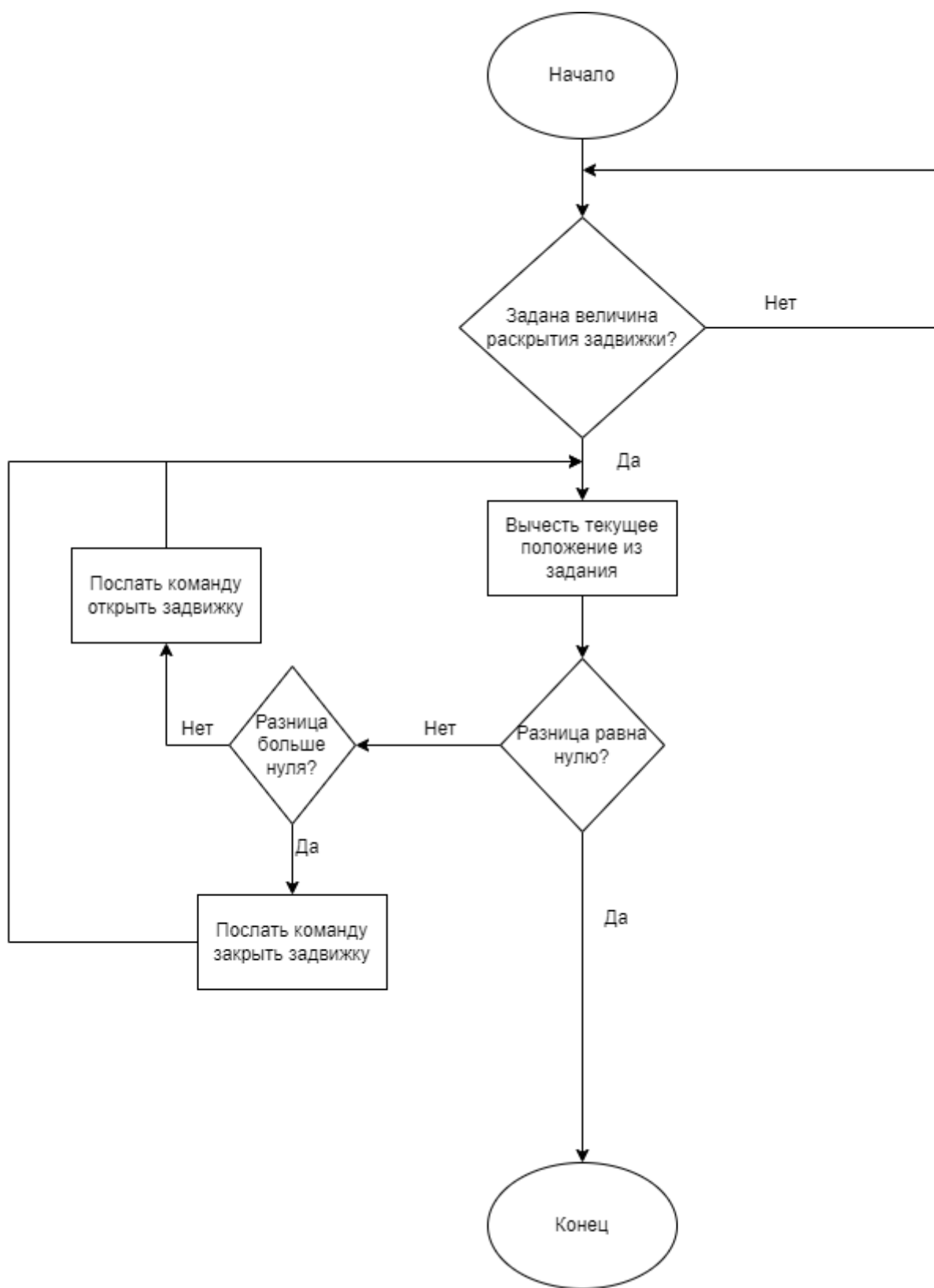


Рисунок 28 – Изначальный алгоритм автоматического управления

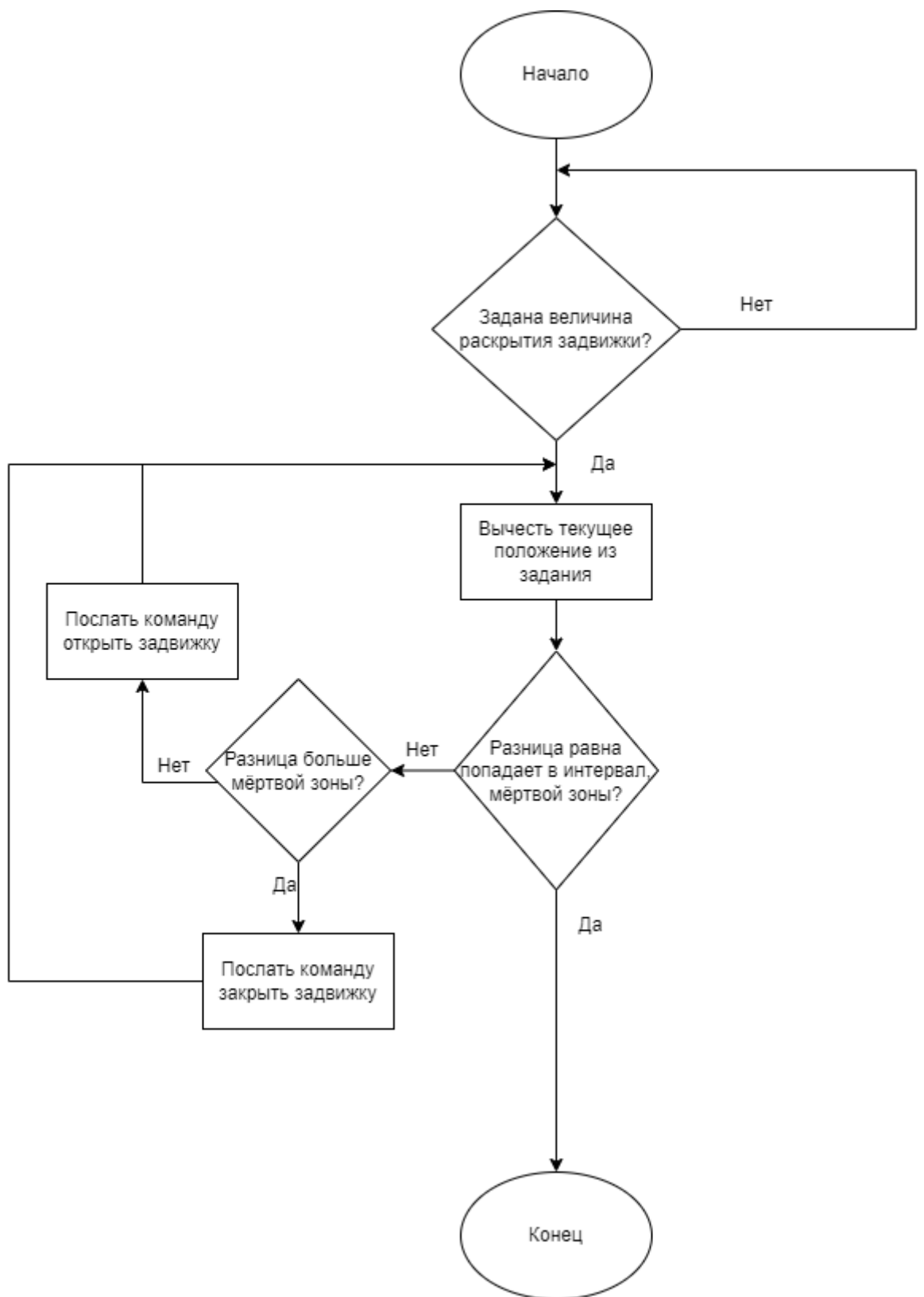


Рисунок 29 – Доработанный алгоритм работы системы автоматического регулирования

Обработка данных в MasterSCADA проводится только внутри функциональных блоков, содержащихся в дереве объектов. Поэтому, для создания автоматизированной системы управления необходимо добавить в дерево объектов объект №1 (Рисунок 30).

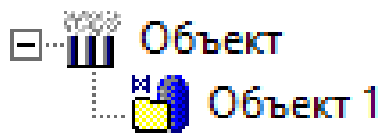


Рисунок 30 – Объект

В качестве компьютера для данного объекта выберем компьютер 1. Поскольку и клиент, и сервер в данной SCADA системе располагаются на одном и том же компьютере, то в качестве компьютера для объекта 1 прописываем компьютер 1. Поле, в котором требуется указать компьютер, располагается во вкладке «Общие», отображённом на рисунке 31.

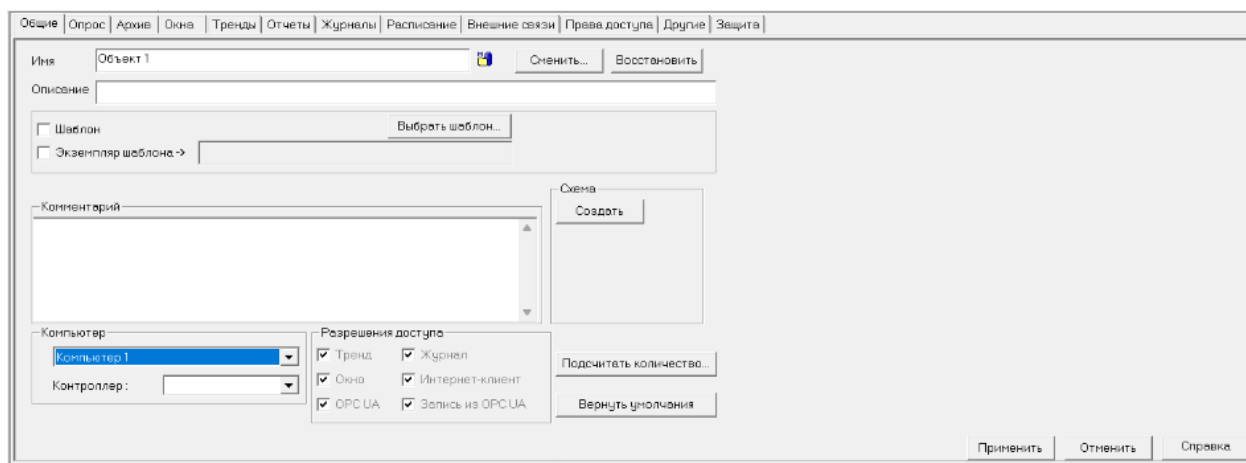


Рисунок 31 – Вкладка «общие» Объекта 1

Для того, чтобы стало возможно задать процент открытия задвижки, добавим в объект 1 значение и озаглавим его как «задание».

Добавим также в дерево объектов, в объект 1, расчёт и дадим ему имя «Положение». Данный блок будет отвечать за конвертацию значения переменной «Положение задвижки» в процентный формат. Поскольку переменная типа int 16 не способна отображать дробные значения, процент открытия

здвижки данной OPC переменной отображается как число от нуля до тысячи. Чтобы получить процентное значение необходимо значение данной переменной поделить на десять. Для это в окне параметров перейдём на вкладку функция.

В открывшееся окне формул перенесём OPC переменную «% положения задвижки» и пропишем функцию «%/положения задвижки/10». Окно формул функционального блока «Положение» отображено на рисунке 32.

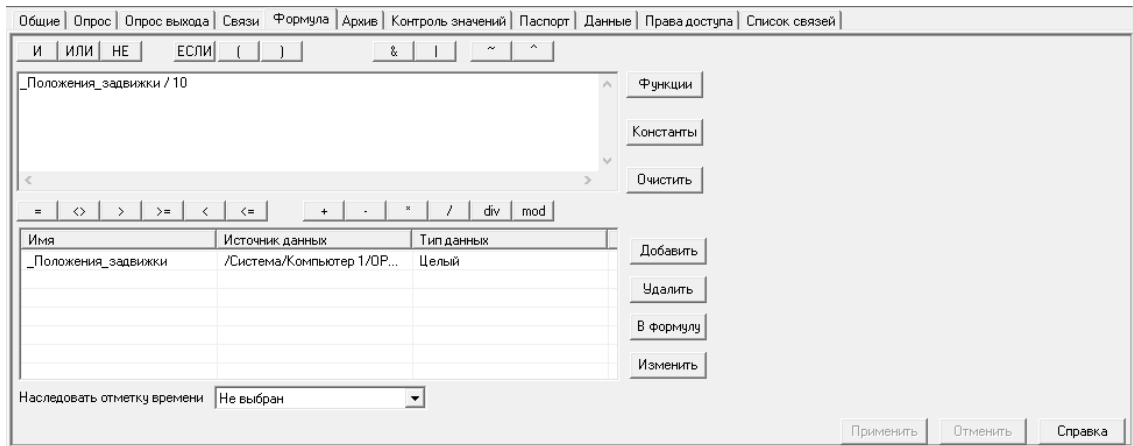


Рисунок 32 – Окно формул Положения.

Для того, чтобы присвоить данному расчёту имя, необходимо перейти на вкладку общие данного функционального блока и ввести в строку имён желаемое наименование. Данная вкладка изображена на рисунке 33.

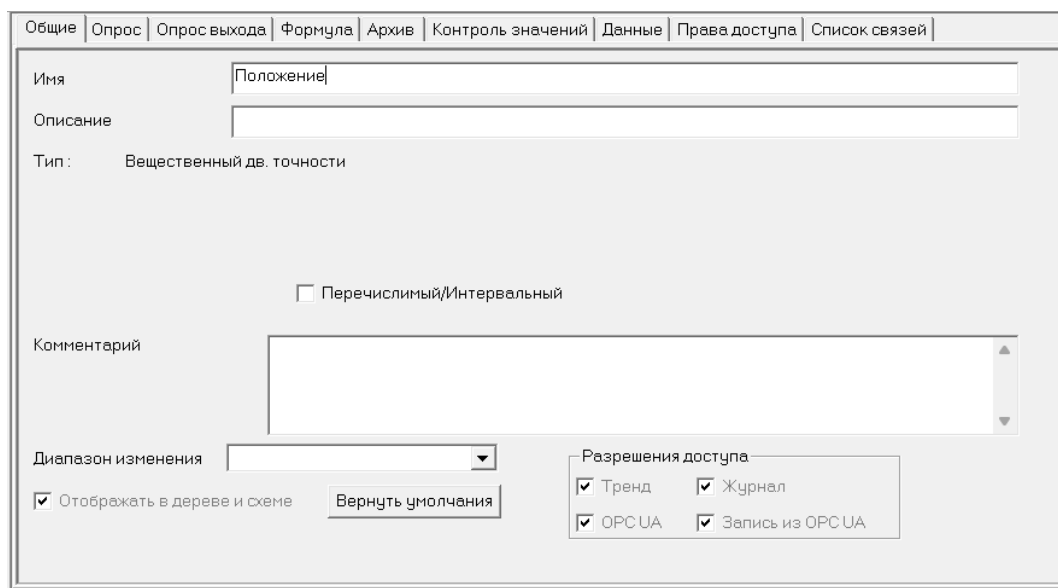


Рисунок 33 – Присвоение имени функциональному блоку расчёт

Для того, чтобы считать необходимый пункт из графы «состояние прибора», необходимо применить к данной переменной битовую маску. Осуществить это в третьей версии MasterSCADA невозможно, однако визуально функциональный блок «распаковка аналогового 32бит_значения», расположенный во вкладке вычисления – рисунок 34 – способен преобразовать любую переменную, объемом 32 бита или меньше в 32 логических переменных.

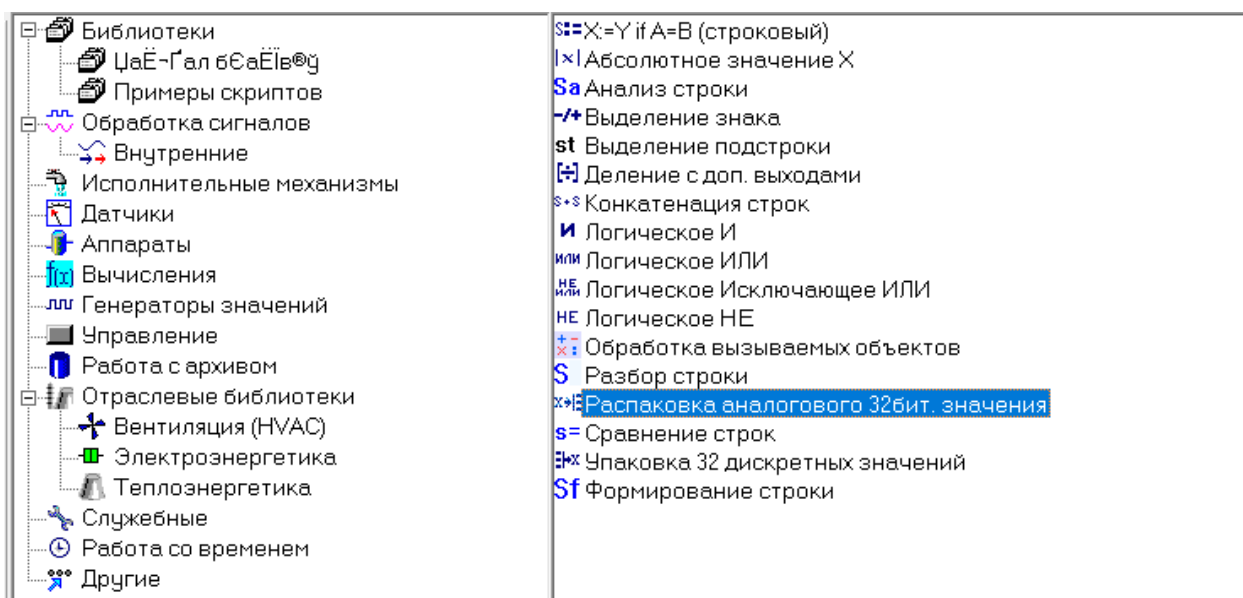


Рисунок 34 – Библиотека функциональных и визуально-функциональных блоков

Разместим данный блок в дереве объектов. В настройках данного блока изменим число выходов с тридцати двух на восемь. Окно настроек с правильными параметрами изображено на рисунке 35.

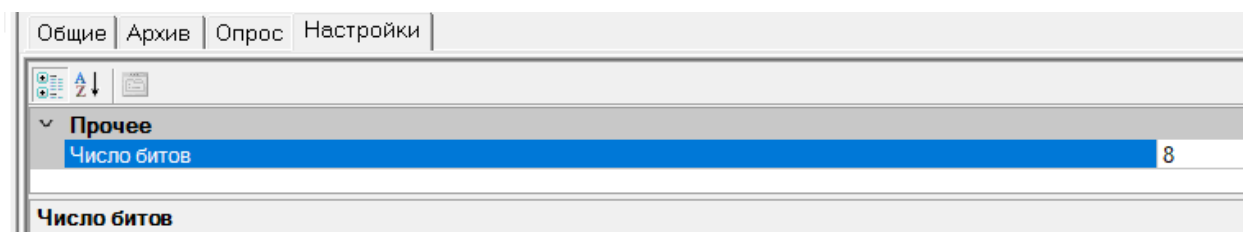


Рисунок 35 – Настройки параметров функционального блока «Распаковка аналогового 32бит_значения»

Свяжем вход данного функционального блока с ОРС переменной «состояние прибора».

Для того, чтобы определить в какую сторону требуется сдвинуть задвижку, дабы раскрыть задвижку на требуемую величину, необходимо высчитать ошибку регулирования, представляющую собой разницу между текущим положением задвижки и заданием.

Добавим в дерево объектов функциональный блок расчёт, озаглавленный как Ошибка Регулирования и введём следующую формулу:

Положение – Задание

Данная формула представлена на рисунке 36

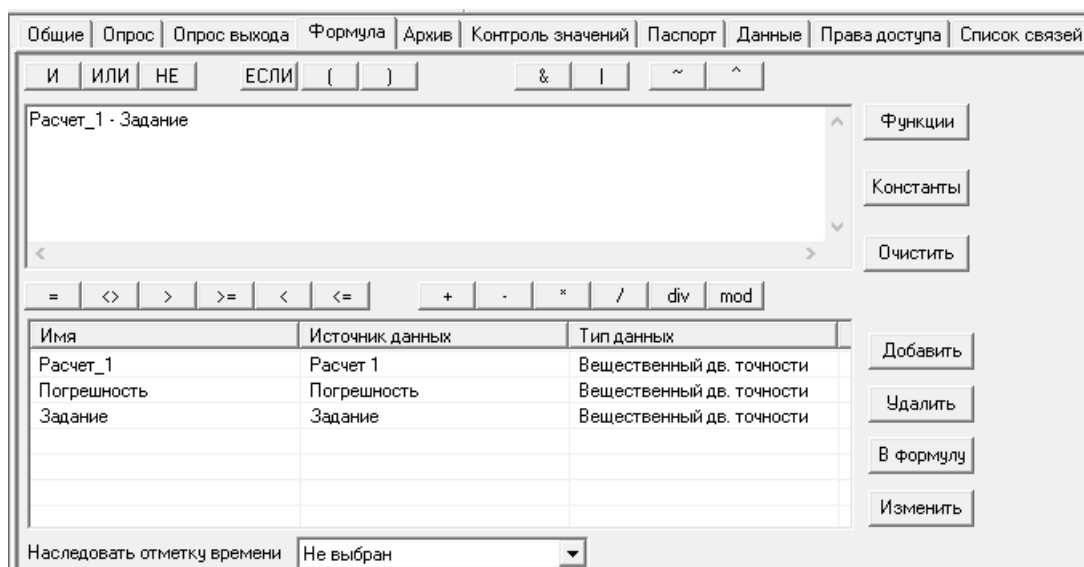


Рисунок 36 – Формула расчёта «Задание»

Добавим ещё три блока расчёт, ответственных за автоматическую подачу стенду сигналов об открытии задвижки, закрытии задвижки и остановке движения задвижки.

ОРС переменные Открыть, Закрыть и Стоп являются логическими, а потому данные блоки тоже должны формировать логический сигнал. Одной из формул, формирующих логический сигнал, является неравенство, так что наиболее подходящим для нашей программы автоматического управления

будут формулы, в которых значение ошибки регулирования сравнивается со значением мёртвой зоны:

Студенты должны будут экспериментальным путём определить, мёртвую зону, в нашем случае, в связи с высокой задержкой, она оказалась достаточно широкой, $\pm 3\%$.

Потому, в качестве формулы для расчёта сигнала открытия запишем:

Ошибка регулирования < -3

Окно формул данного блока изображено на рисунке 37

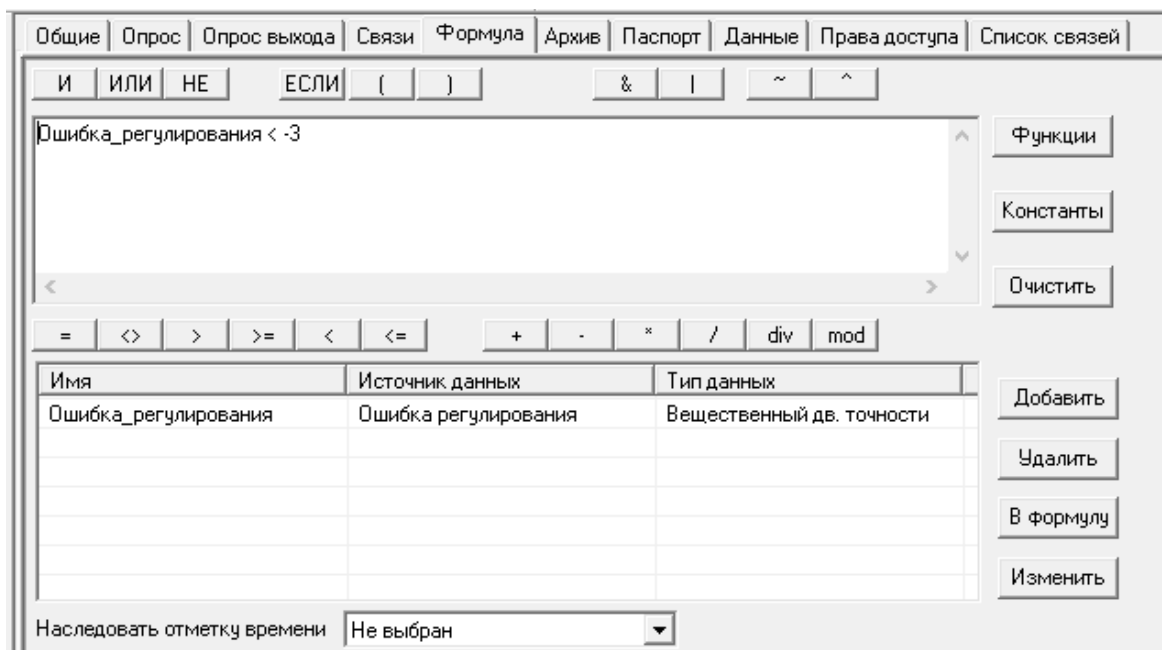


Рисунок 37 – Формула расчёта «Открыть»

В окне формул расчёта «Закрыть» введём формулу:

Ошибка регулирования > 3

Окно формул изображено на рисунке 38.

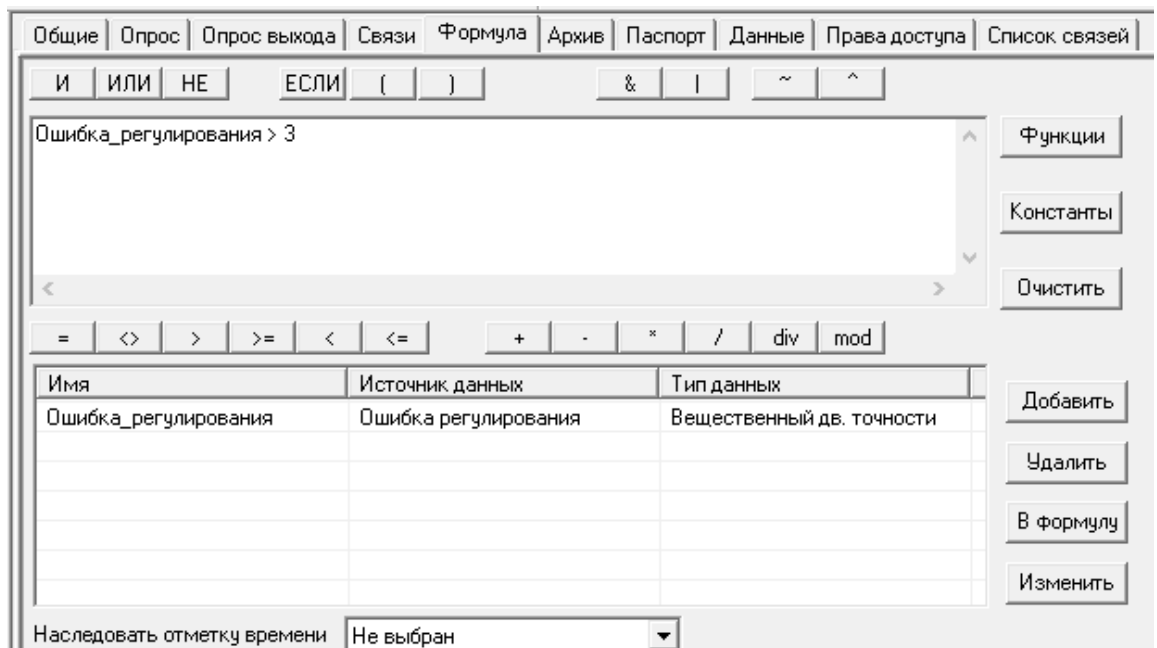


Рисунок 38 – Формула расчёта «Закреть»

Добавим функциональный блок «Расчёт Остановки», с формулой:

$$\text{abs}(\text{Ошибка регулирования}) < 3$$

Окно формул данного функционального блока изображено на рисунке

39.

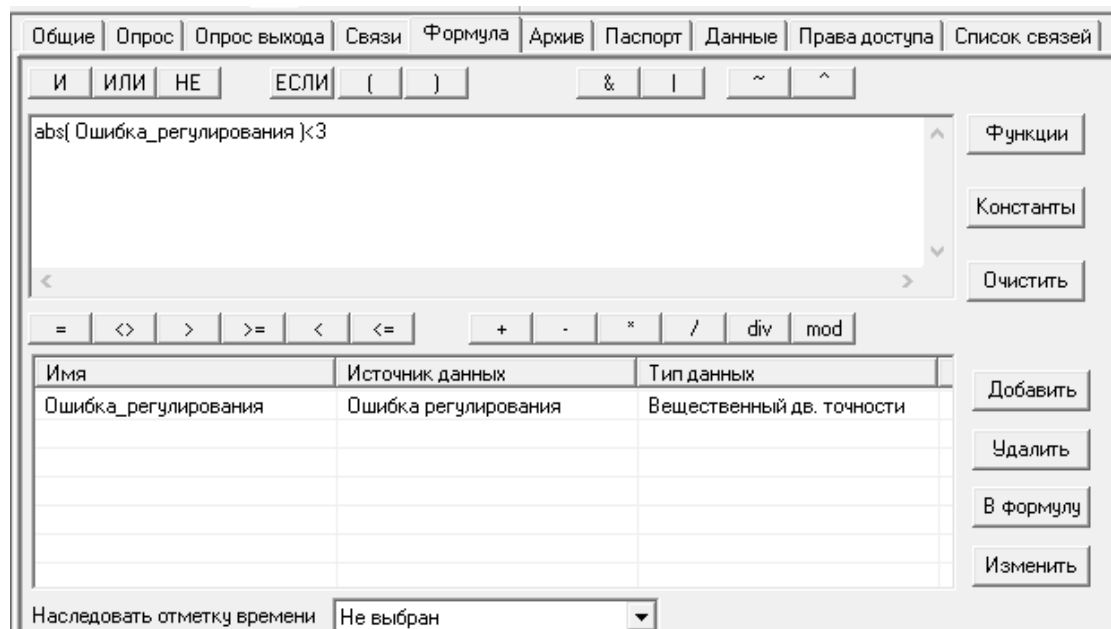


Рисунок 39 – Окно формул расчёта Остановки

Поместим в дерево объекта три значения: открыть(авто), закрыть(авто) и стоп(авто). Сегмент дерева объекта с созданными значениями изображён на рисунке 39.

Установим связь между выходом данного значения и входами соответствующих OPC переменных.

Как только будет установлена связь между выходом OPC переменной и значением появится предупреждение о том, что установлена связь между двумя архивируемыми параметрами и будет предложено снять архивацию с одного из параметров (рисунок 41). Снимем архивацию со Значений.

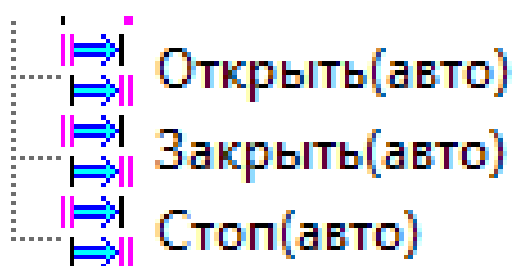


Рисунок 40 – Значения.

Установлена связь между двумя архивируемыми параметрами. Это вызовет дублирование архивов. Рекомендуется снять архивирование с одной из них.

Рисунок 41 – Предупреждение

Необходимо убедиться, что типы OPC переменной и тип созданного значения совпадают, поскольку если это будет не так, то мы не сможем установить связь между выходным сигналом кнопки и OPC переменной. Убедимся, что переменные «Открыть(авто)», «Заккрыть(авто)» и «Стоп(авто)» имеют логический тип, как показано на рисунке 42. Если связь была установлена неправильно, то окно настроек значений примет вид, показанный на рисунке 43.

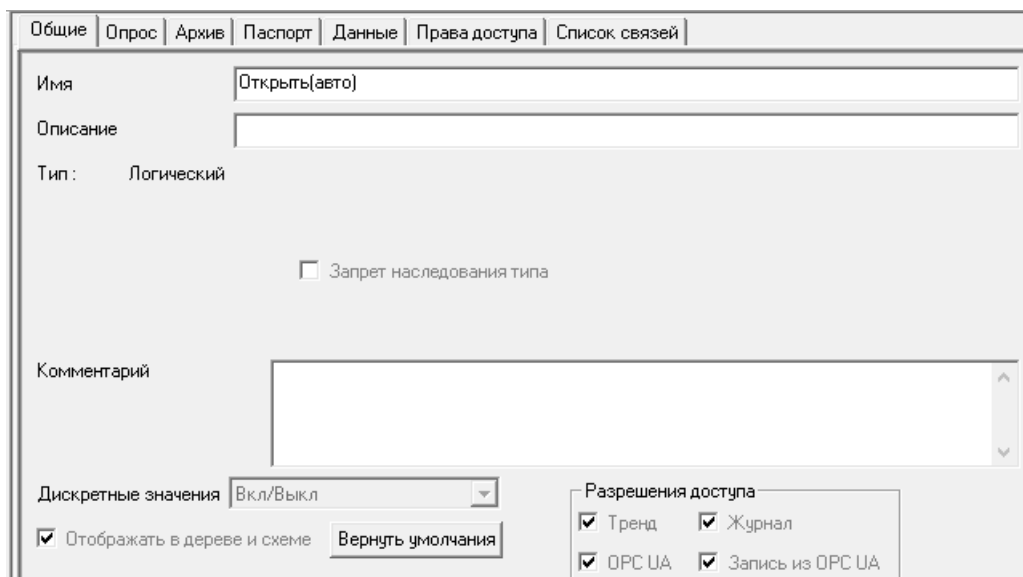


Рисунок 42 – Окно настроек значений

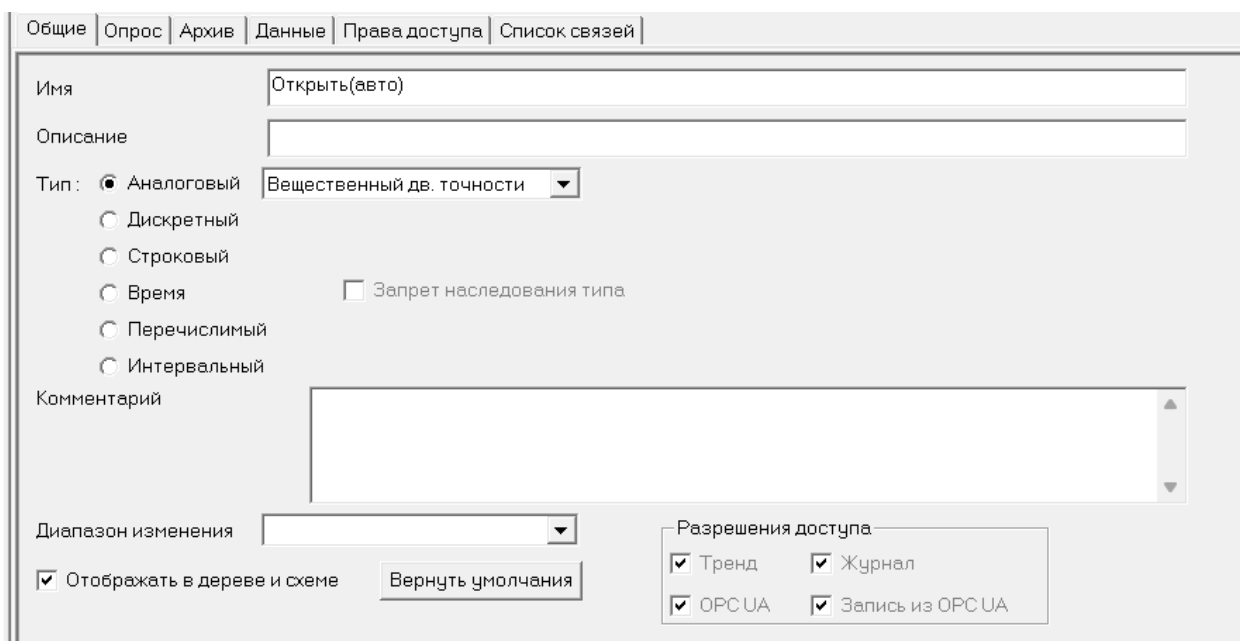


Рисунок 43 – Неправильное окно настроек значений

SCADA система содержит не только функциональные блоки, но и визуально-функциональные блоки, переводящие значения переменных в понятную оператору форму, а также позволяющие взаимодействовать с переменными при помощи кнопок и тумблеров, а не непосредственным изменением значений переменных в окне объектов. Совокупность визуально-функциональных блоков, ответственных за управление конкретным объек-

том, размещается на соответствующей мнемосхеме. Перейдём на вкладке Окна в окне параметров Объекта 1 и добавим мнемосхему (окно создания мнемосхемы представлено на рисунке 44). Откроем её.

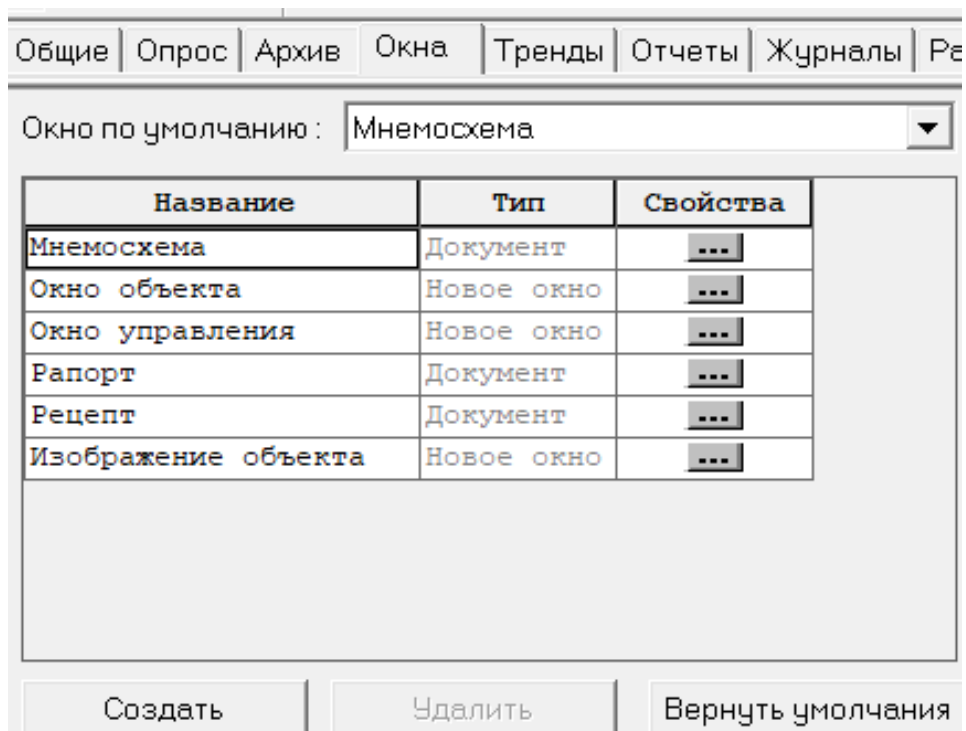


Рисунок 44 – Окно создания мнемосхемы

Для отображения переменной «положение», полученной при помощи расчёт в соответствующем функциональном блоке, разместим на мнемосхеме стрелочный прибор (рисунок 45). Расположенный в палитре, во вкладке «управление» (Рисунок 46).

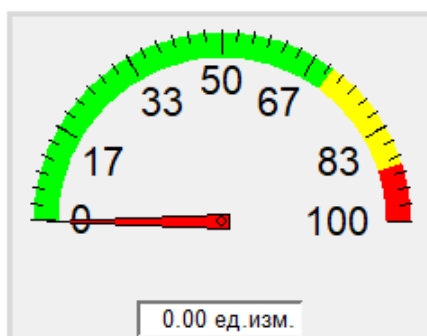


Рисунок 45 – Стрелочный прибор



Рисунок 46 – Вкладка «управление» окна «палитра»

На вкладке свойства (расположенной рядом со вкладкой палитра на мнемосхеме) выберем заголовок «динамизация входных значений» (рисунок 47) и переместим в строку «величина значения» значение переменной «положение».

Для работы оператора необходимо визуально отображать информацию о состоянии прибора. Поскольку каждая из переменных состояния имеет всего два значения, то отобразить информацию о приборе можно при помощи восьми лампочек. Каждый бит переменной «Состояние Прибора» содержит информацию об одном из восьми параметров.

Добавим к объекту №1 объект «Лампочки» и перенесём в него восемь визуально-функциональных блоков «Индикатор Состояния».

Индикатор Состояния является визуальной функциональным блоком, содержащим один или множество входов и один выход. В зависимости от сигнала на каждом из входов, визуальная составляющая блока будет менять

цвет. Максимальное отображаемое число цветов равно $2n$, где n – число входов. Он расположен в библиотеке функциональных блоков во вкладке датчики (рисунок 48)

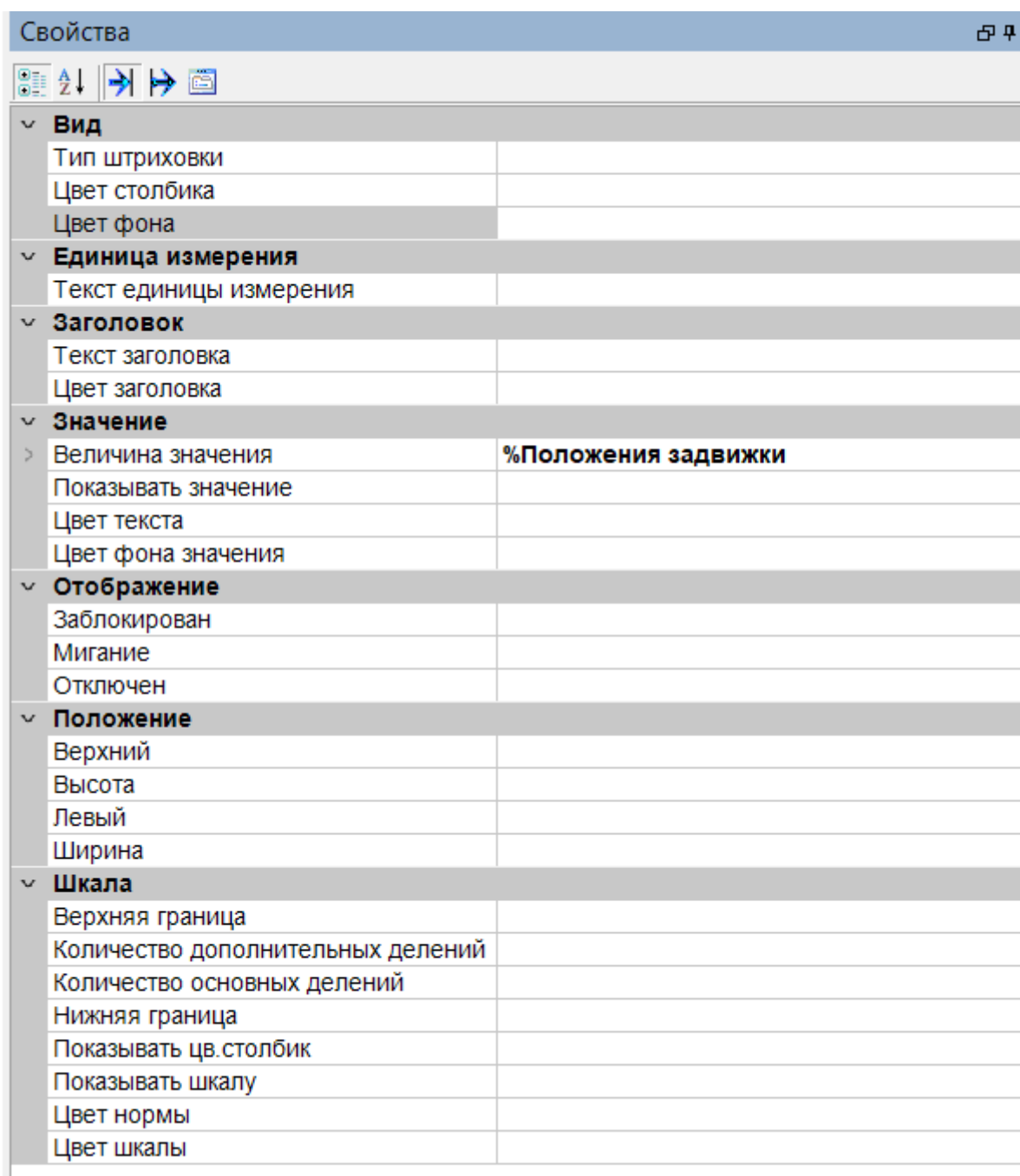


Рисунок 47 – Окно настройки свойств визуально-функционального блока «Стрелочный Прибор»

При размещении на мнемосхеме Индикатор Состояния выглядит как красный круг, настройка позволяет изменить положение функционального

блока на мнемосхеме или его размер, но не цвет. Используемые настройки цвета будут отображаться только тогда, когда проект MasterSCADA будет переведён в режим отображения.

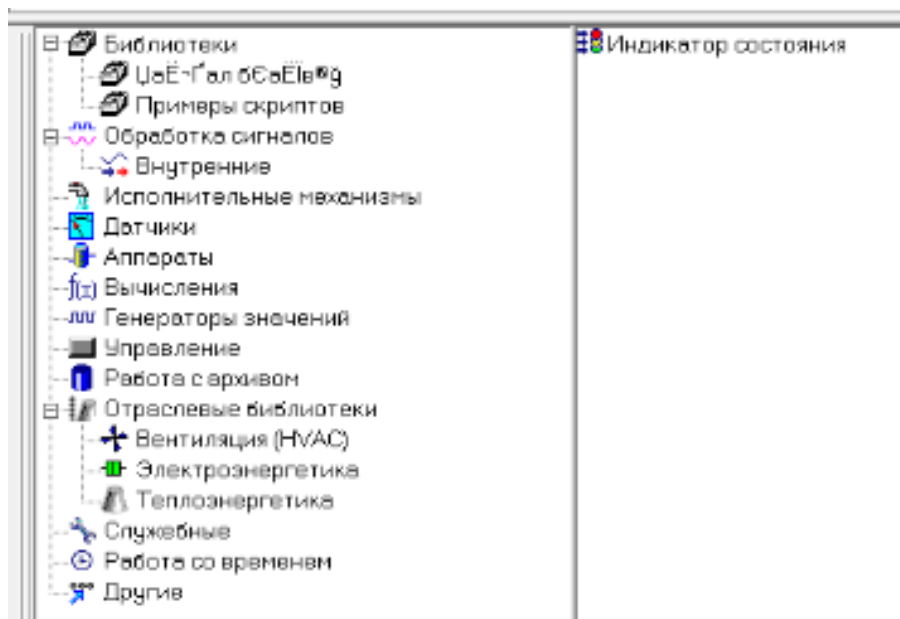


Рисунок 48 – Библиотека функциональных блоков

В настройках визуально-функционального блока выберем число входов: «1» и настроим связь соответствующих выходов блока «распаковка 32_бит значения» со входами индикаторов состояния. Вкладка «настройки» располагается в окне параметров и изображена на рисунке 49.

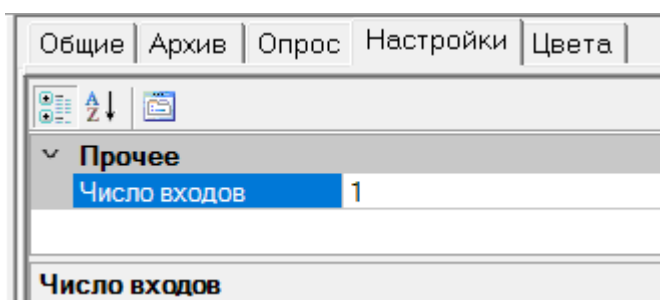


Рисунок 49 – Окно настроек Индикатора Состояния

Озаглавим индикатор состояния, связанный с первым выходом, как «полностью закрыто» и разместим его на мнемосхеме. Изменить название можно на вкладке «общие» окна параметров визуально-функционального

блока (рисунок 50). Данный индикатор состояния должен уведомлять оператора о том, что задвижка полностью закрыта и дальнейшее закрытие невозможно. Поскольку это штатная ситуация, в качестве цвета по умолчанию выберем тёмно-зелёный, а цвета 0, тревожного цвета – яркий светло-зелёный. Настройка цвета проводится на вкладке «Цвета», изображённой на рисунке 51.

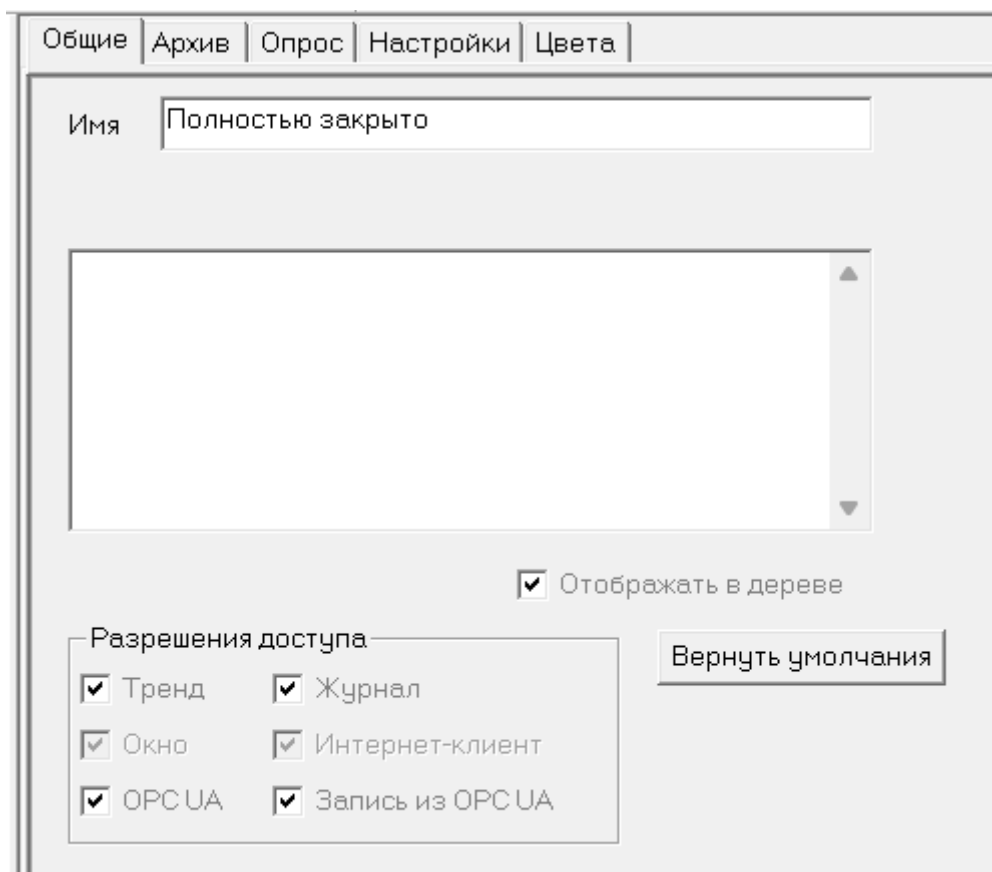


Рисунок 50 – Присвоение функциональному блоку индикатор состояния имени

Те же цвета будут использованы и при настройке индикаторов состояния со второго по четвёртый, так как они отвечают за уведомление оператора SCADA системы о полном открытии задвижки, а также об направлении движения задвижки. Ничто из этого не является опасной ситуацией, требующей немедленного вмешательства.

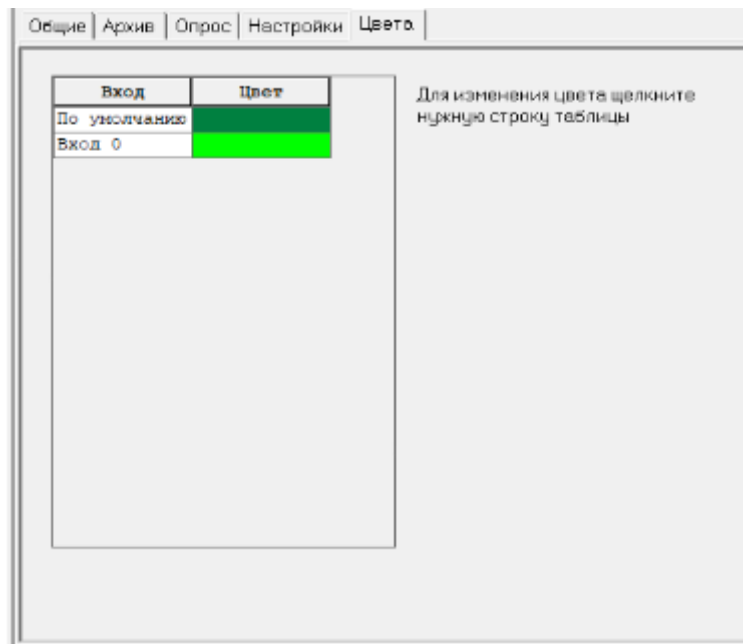


Рисунок 51 – Настройка цветов Индикаторов штатной работы станда

Пятый, шестой и седьмой индикатор состояния информируют оператора о том, что произошла авария, задвижка перегружена или произошло проскальзывание соответственно. Любое из данных состояний считается критическим, так что зелёный цвет индикации не подойдёт, вместо этого в качестве основного цвета выберем серо-красный, а в качестве цвета ноль – рубиновый. Выбранный цвета изображены на рисунке 52.

Восьмой же выход функционального блока «распаковка 32_бит значения» содержит булеву переменную, уведомляющего оператора о том, что запущена внутренняя программа автоматического управления контролера. В нашей Лабораторной Работе студенты будут писать программу автоматического управления внутри SCADA системы и, соответственно, возгорание данного индикатора – должно сообщить оператору SCADA системы о том, что прибор настроен на выполнение иной программы. Поэтому, в качестве тревожного цвета выберем рубиновый, а цвета по умолчанию – серо-красный.

Разместим данные блоки на мнемосхеме в столбик по правую сторону от стрелочного прибора. Для равномерного распределения индикаторов со-

стояния по вертикальной оси выделим все восемь размещённых кнопок и нажмём на кнопку выравнивания по вертикальной оси. Положение данной кнопки на оси настроек мнемосхемы продемонстрировано на рисунке 53

Напротив каждого из индикаторов состояния текст с соответствующей надписью. Блок Индикаторов состояния стенда изображён на рисунке 54.

Разместим на мнемосхеме четыре кнопки без фиксации, озаглавив их как «открыть», «закрыть» «стоп» и «авто». Кнопки без фиксации расположены на мнемосхеме, в окне палитра во вкладке управление. Положение данных кнопок изображено на рисунке 55.

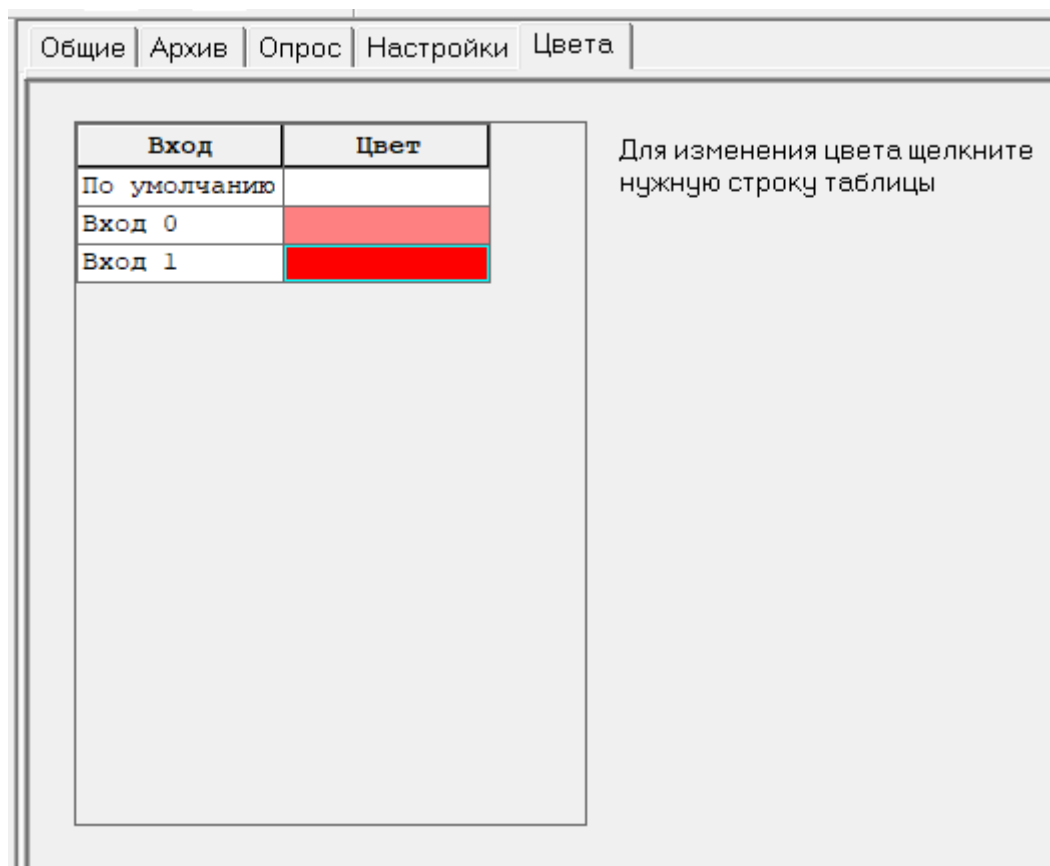


Рисунок 52 – Настройка цветов индикаторов ошибок и проскальзывания

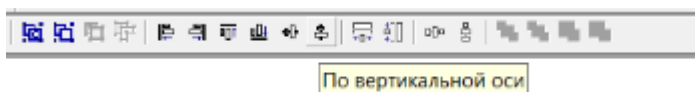


Рисунок 53 – Выравниванием по вертикальной оси

Откроем вкладку свойств, находящуюся рядом с вкладкой «палитра» и нажмём на вкладку «динамизация выходных значений». Перенесём OPC переменную «Открыто» из дерева системы в строку «кнопка нажата» кнопки Открыть. Настройка данной кнопки изображена на рисунке 56.

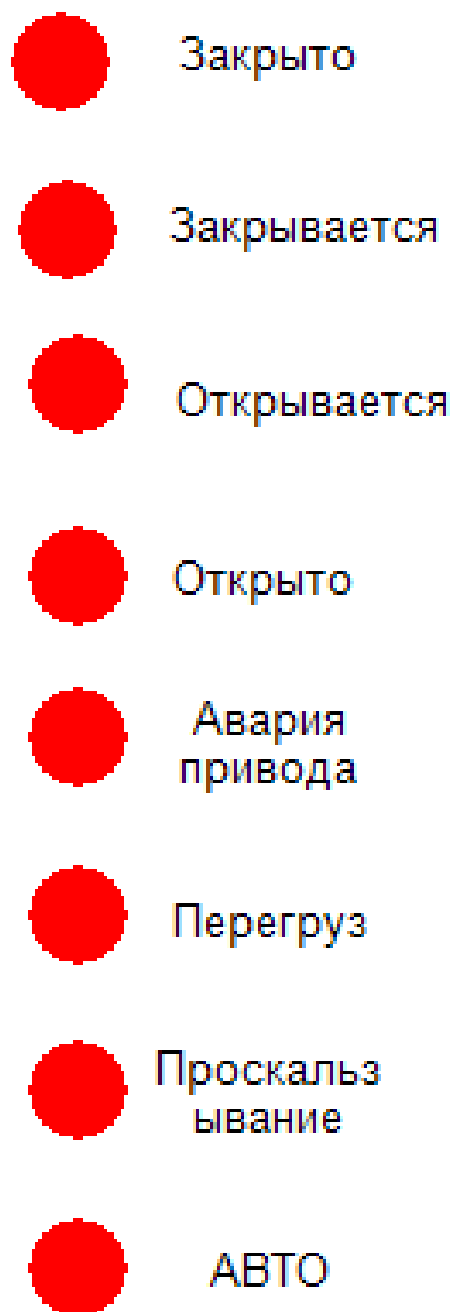


Рисунок 54 – Индикаторы Состояния



Рисунок 55 – Вкладка «управление» окна «палитра»

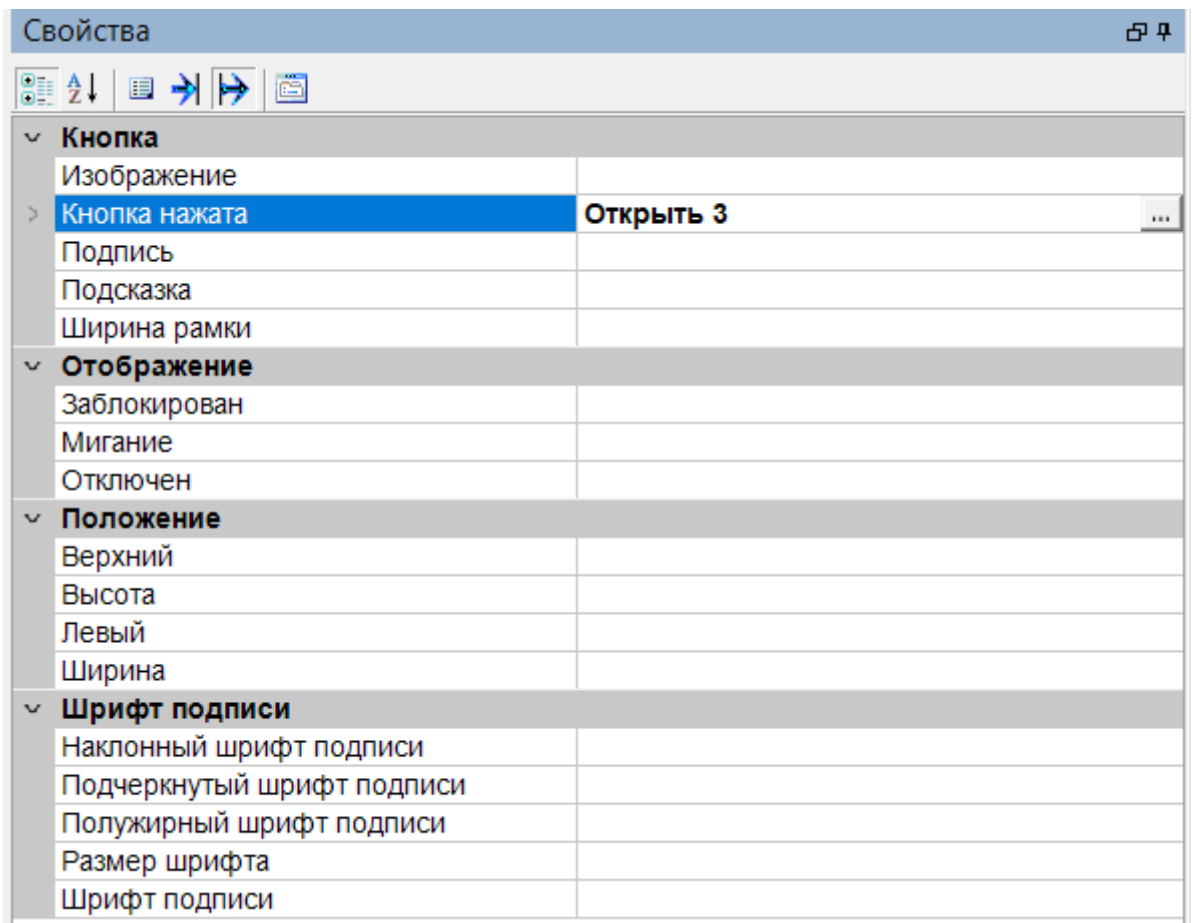


Рисунок 56 – Окно настроек кнопки Открыть

Для удобного задания процента открытия задвижки необходимо разместить на мнемосхеме задатчик, располагающийся в окне палитра, во вкладке управление (рисунок 57) Нажмите правой кнопкой мышки на блок свойств и в появившемся окне выберите вид задатчика «Горизонтальный». Окно редактирования свойств показана на рисунке 58.

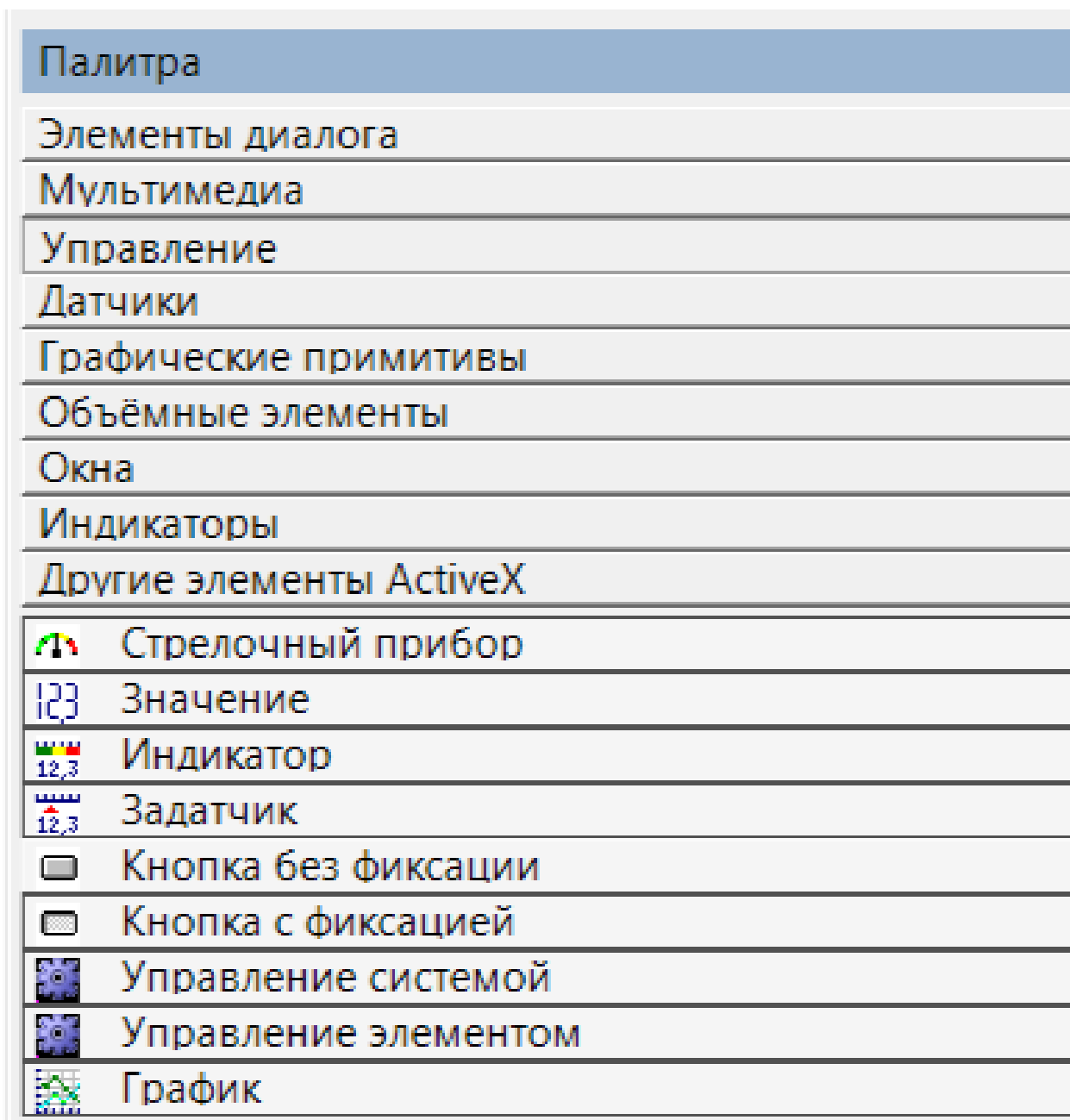


Рисунок 57 - Вкладка «управление» окна «палитра»

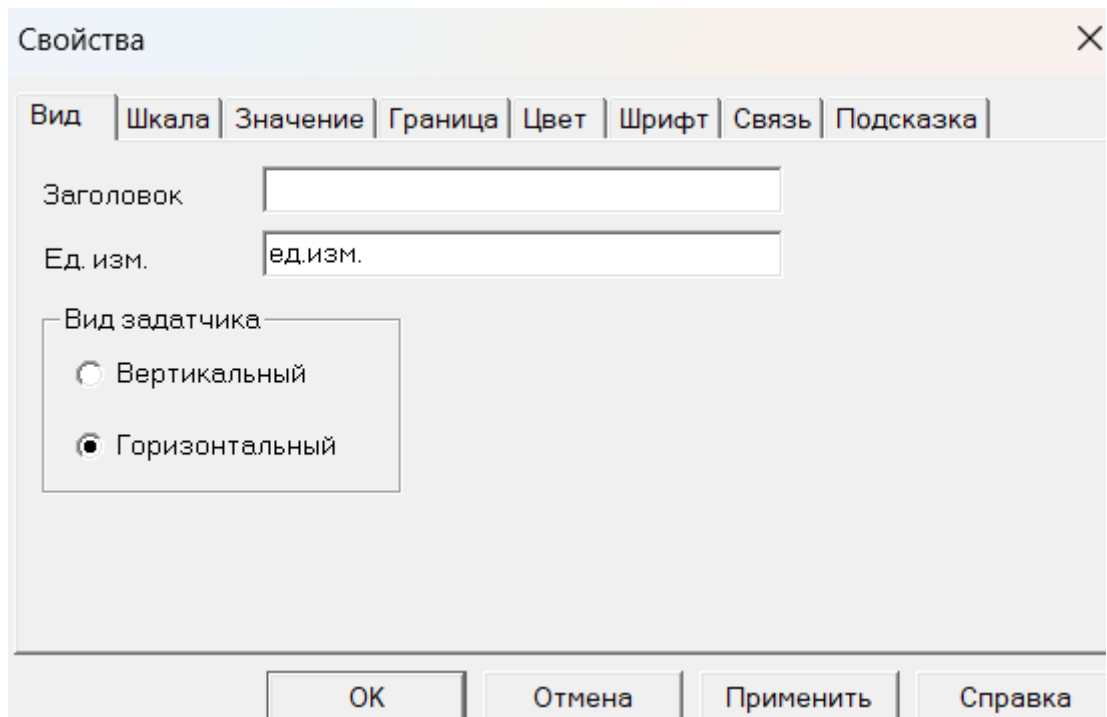


Рисунок 58 – Настройка свойств визуального блока задатчик

Откроем вкладку динамизация выходных значений и установим связь между OPC переменной «Задание» и строкой Величина в строке выходных значений.

Мнемосхема, которую студент должен будет создать при выполнении лабораторной работы показана на рисунке 59, а итоговое дерево объектов – 60.

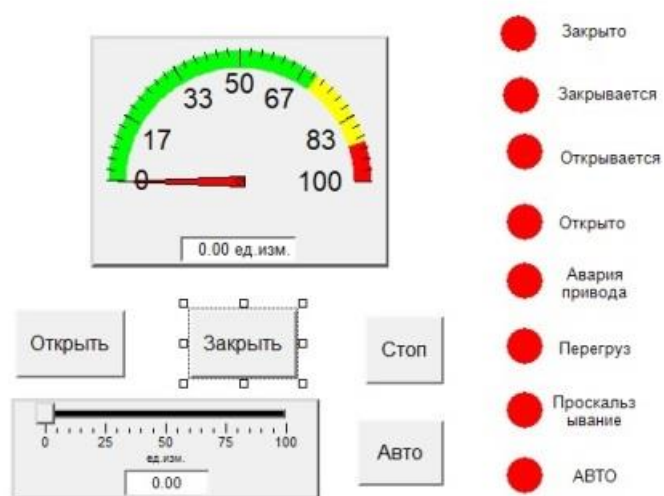


Рисунок 59 – Разработанная мнемосхема

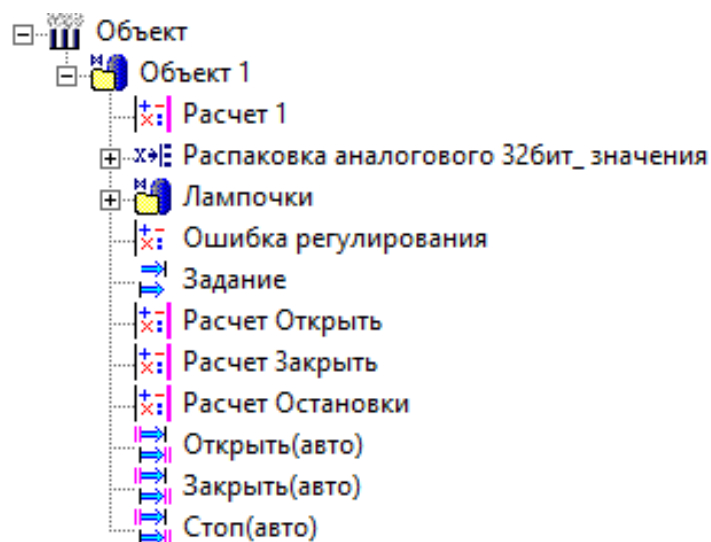


Рисунок 60 – Дерево объектов

3.4 Установка связи с удалённым OPC UA сервером посредством сети интернет

В ходе выполнения третьей лабораторной работы студенты должны будут установить обмен данными между двумя компьютерами, располагающимися в одной сети.

Для этого необходимо включить оба компьютера и подключить их к роутеру.

Студенты должны узнать IP адрес компьютера, выполняющего роль сервера в данной сети. Для этого необходимо открыть Панель Управления, перейти по пути Суть и Интернет > Центр управления > Центр управления сетями и общим доступом >Изменение параметров адаптера. На экране откроется вкладка сетевые подключения. Далее необходимо дважды нажать левой кнопкой мышки на иконку, содержащую название сети и, в открывшемся окне, нажать на кнопку «сведения».

В открывшемся окне можно узнать IP4 адрес компьютера в данной сети.

Необходимо убедиться, что обмен данными между сервером и компьютером возможен. Для этого на компьютере-клиенте требуется открыть ко-

мандную строку, путём ввода комбинации клавиш Win+R, а в открывшемся окне прописать «cmd». Скриншот данного окна изображён на рисунке 61.

Как только команда введена, необходимо подтвердить ввод команды нажатием кнопки «ОК». Это откроет командную строку. Для того, чтобы проверить обмен данными между компьютерами необходимо ввести команду ping «IP4server»

Вместо «IP4server» команда должна содержать IP адрес компьютера-сервера. Пример написания команды помещён на рисунке 62.

Если обмен пакетами проведён успешно, то можно перейти к следующему шагу. Если нет – потребуется открыть настройки параметров сети и разрешить обоим компьютерам, и серверу, и клиенту, отображаться в частных сетях.

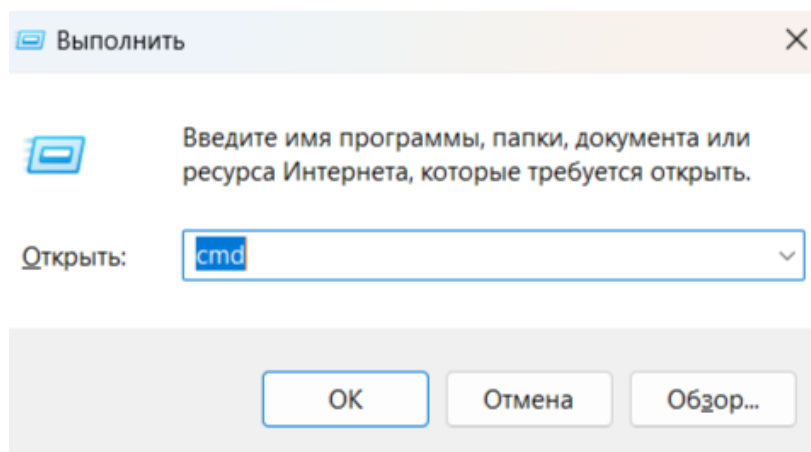


Рисунок 61 – Открытие командной строки

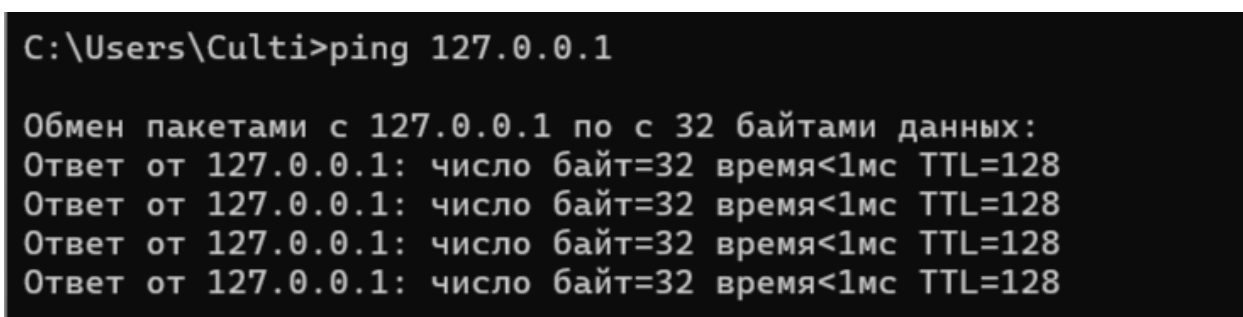


Рисунок 62 – Обмен пакетами с компьютером в сети

Как только мы убедились в возможности обмена данными между компьютером-клиентом и компьютером-сервером, можно приступить к выполнению лабораторной работы. Студент должен открыть на компьютере-сервере Modbus Universal MasterOPC Server, запустите файл конфигурации сервера, использующийся при выполнении второй лабораторной работы, и запустить его.

На компьютере-клиенте же необходимо открыть MasterScada 3.7, добавить в дерево системы компьютер и протокол OPC UA сервер. Как и в предыдущих работах, необходимо провести настройку OPC UA клиента, в качестве адреса указав IP адрес компьютера-сервера. Окно настроек OPC UA клиента изображено на рисунке 63.

Настройки OPC UA клиента

Настройки подключения **Дополнительно**

Информация о сервере

Сервер

Настройки безопасности

Политика безопасности

Режим безопасности сообщений

Настройки аутентификации

Анонимно

Имя пользователя

Пароль

Настройки сессии

Имя сессии

Рисунок 63 – настройки OPC UA клиента

Добавим к дереву тренд, чтобы наблюдать за зависимостью положения задвижки от времени на экране компьютера. Для этого добавим к объекту тренд и перенесём на ось абцисс переменную «положение задвижки».

4 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

4.1 Безопасность

Обучение студентов работе с OPC UA серверами будет происходить в 306-ой аудитории шестого корпуса Амурского Государственного Университета. Поскольку в ходе данной работы студенты будут непосредственно взаимодействовать с электрооборудованием – стендом, включающим в себя электродвигатель, перемещающий задвижку – то они обязаны будут пройти инструктаж по технике безопасности при работе с электроприборами в начале семестра.

Согласно пункту три главы два точка пять приказа Минэнерго РФ[18] «Правила Технической эксплуатации электроустановок потребителей» на электродвигателе, а также на приводимой им в движение задвижке, нанесены стрелки, указывающие направление движения подвижных элементов.

Пункт 2.5.5 вышеназванного приказа гласит, что, при кратковременном перерыве электропитания электродвигателей, должен быть обеспечен самозапуск ответственных механизмов, если от их работы зависит безопасность технического процесса. Поскольку задвижка стенда к подобному оборудованию не относится, то проведение самозапуска не только не является обязательным, но и противопоказанно, так как подвергает риску жизнь и здоровье людей, работающих с оборудованием после отключения электропитания.

Также, двигатель должен, через установленные проектом производства работ промежутки времени, проходить испытания, в ходе которых проверяется сопротивление изоляции, сопротивление постоянному току, измерение зазоров в подшипниках скольжения и их вибрация, и другие параметры. Оборудование, эксплуатирующееся в лаборатории триста шесть шестого корпуса, проходит данную проверку ежегодно, в августе, перед началом учебного года. Ответственным в лаборатории за функциональность оборудования является доцент Рыбалёв Андрей Николаевич.

Электродвигатель является не единственным оборудованием, входящим в состав стенда. Помимо него, на стенде также размещён откалиброванный амперметр и вольтметр, на которых нанесены символы, соответствующие отображаемым величинам.

В триста шестой аудитории шестого корпуса освещение осуществляется при помощи шести газоразрядных ламп. Мощность газоразрядных ламп соответствуют проектной, а электропитание осветительных приборов выполняется по отдельным линиям, изолированных от линии электроснабжения прочего электрооборудования. Величину падения напряжения на наиболее удалённых лампах сети внутреннего освещения проверить не удалось, однако, если освещение соответствует стандарту, то падение не должно превышать 5%.

Лампы располагаются на высоте трёх метров от уровня пола, что допускает их обслуживание с приставных лестниц и стремянок, так как, согласно вышесказанному приказу, необходимость в использовании кранов или иного подъёмного оборудования возникает только в том случае, если осветительные приборы находятся на расстоянии более пяти метров от уровня пола.

4.2 Экологичность

В ходе выполнения данной лабораторной работы, студенты будут взаимодействовать с двумя персональными компьютерами, а освещение в данном помещении будет производиться при помощи шести газоразрядных лам.

Согласно пункту 15 главы 2.12 Приказа Минэнерго РФ «Правила технической эксплуатации электроустановок потребителей» утратившие работоспособность люминесцентные лампы, лампы типа ДРЛ и другие источники, содержащие ртуть (отходы первого класса опасности), должны храниться в специальном помещении, в ожидании вывоза для последующего обезвреживания и утилизации. За исполнением озвученных выше требований действующего приказа следит профессор и кандидат технических наук Скрипко Оль-

га Валерьевна. Электрики снимают вышедшие из строя газоразрядные лампы и переносят их на склад первого корпуса. Через какое-то время данные лампы отправляются на дезактивацию и переработку на один из заводов, расположенных в Приморском крае.

Утилизация вышедших из строя персональных компьютеров проходит без нарушений: в соответствии с пунктом 1.1 статьи. 20 Федерального закона от 26.03.1998 № 41-ФЗ[19] лом и отходы драгоценных металлов подлежат сбору во всех организациях, в которых они образуются. Собранные лом и отходы обязательно учитываются.

Поскольку при производстве компьютеров в малых количествах используются драгоценные металлы, то утилизацию данной техники должен осуществлять уполномоченная организация, способная осуществлять переработку подобных отходов, коей, в данном случае выступает Благовещенская Свалка.

Кроме того, сами персональные компьютера являются отходами четвёртого класса опасности, чья утилизация, хоть и не столь явно регламентируется, как утилизация отходов, но всё же должна проводиться специально уполномоченными организациями. В частности, регулированию подлежит сжигание отходов четвёртого класса опасности. Закон не запрещает сжигание данных материалов, однако регламентирует, что данный процесс возможно осуществлять только при наличии специальных фильтров, препятствующих выбросу продуктов сжигания в атмосферу.

4.3 Чрезвычайные ситуации

Любая работа с электрооборудованием предполагает определённый риск, в частности связанный с пожарами. Однако, согласно статье 32 Федерального Закона о техническом регламенте и требованиям к пожарной безопасности от 30 Апреля 2009-го года[20], риск возникновения пожара не должен превышать одной миллионной в год. Для понижения величины данного параметра, а также для обеспечения безопасности всех потенциальных

жертв, в данном законе разработаны определённые требования, перечисленные в вышеназванном законе.

Здание Амурского Государственного Университета, в котором располагается аудитория со стендом и будет проходить обучение студентов работе с OPC UA серверами классифицируется как здание категории Ф4.2 Все студенты обязаны прослушать инструктаж по пожарной безопасности, который проводится Рыбалёвым Андреем Николаевичем и поставить роспись в журнале регистрации инструктажа по пожарной безопасности.

Выход из учебной аудитории находится на расстоянии 5 метров от лестничной клетки категории Л1, что укладывается в требуемые Федеральным Законом характеристики эвакуационных путей. Данные пути не должны быть захламлены мебелью или оборудованием, препятствующим эвакуации. Данное требование выполняется, ширина коридора удовлетворяет предъявленными в законе требованиям, что позволяет, в случае чрезвычайной ситуации, провести эвакуацию учащихся и преподавателей в соответствующие сроки.

Спустившись по данным лестницам, эвакуирующиеся попадают в коридор первого этажа, дальнейшее перемещение по которому приведёт к холлу. Таким образом соблюдается требование федерального закона о том, что выходы лестниц должны вести либо непосредственно к выходу из здания, либо к коридорами ведущим к выходу, расположенному на первом этаже здания.

Для тушения пожаров класса Е «Пожары электроустановок под напряжением» в лаборатории 306 располагается два порошковых огнетушителя. Данный тип огнетушителей является одним из немногих, чьё применение в адрес горящего электрооборудования является допустимым.

Однако, шансы на то, что возникнет ситуация, требующая тушения электрооборудования под напряжением не велики, поскольку, как предписывает статья 142 данного федерального закона, всё оборудование снабжено ав-

томатическими выключателями, так что при возникновении короткого замыкания устройство будет немедленно отсоединено от сети электроснабжения и данный исклучением не является.

При этом, после того как электрооборудование – в данном случае любой из стендов, располагающихся в 306 аудитории 6 корпуса, а также один из многочисленных персональных компьютеров – будет отключено от электросети, будет возможно провести тушение другими, более доступными средствами пожаротушения – песком и водой.

Также, меры обеспечения пожарной безопасности коснулись и электропроводки. Все питающие кабели покрыты изоляцией, что препятствует соприкосновению оголённых проводов и, как следствие, возникновению короткого замыкания.

Помимо этого, сами изолированные проводники находятся в стенах, внутри металлического короба. Таким образом, даже если изоляция в процессе эксплуатации будет повреждена, то воспламенение не распространится за пределы короба.

Однако, следует понимать, что при длительном температурном воздействии, сам короб может расплавиться или нагреться до высокой температуры, что в конечном итоге может привести к разрушению материалов стены. Поэтому, необходимо осуществить своевременное пожаротушение и эвакуацию.

ЗАКЛЮЧЕНИЕ

В ходе выполнения бакалаврской работы был проведён сбор информации о стандартах OPC, в частности о стандарте OPC UA и его преимуществах по сравнению со стандартом OPC DA. Были разработаны три лабораторных работы, с помощью которых студенты смогут ознакомиться с особенностью OPC UA серверов, а также с созданием конфигурационных файлов сервера в программе Modbus Universal Master OPC server и созданием проектов в клиентском приложении в MasterSCADA 3.7.

В ходе подготовке материала для написания лабораторной работы была разработана блок-схема алгоритм программы автоматического управления и принципиальная схема стенда управления задвижкой (в ходе выполнения одной из лабораторных работ студенты должны будут подключиться к реальному объекту по протоколу Modbus. Была обработана информация о работе с MasterSCADA 3.7 и Modbus Universal Master OPC Server.

Были разработаны три файла-конфигурации OPC UA, содержащие 14 различных тегов, а также три проекта MasterSCADA 3.7, один из которых содержит полноценный проект операторской панели и программу автоматического управления задвижкой. Создана виртуальная локальная сеть, устанавливающая обмен данными между двумя удалёнными компьютерами, один из которых содержит клиентское приложение, а второй выступает сервером.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

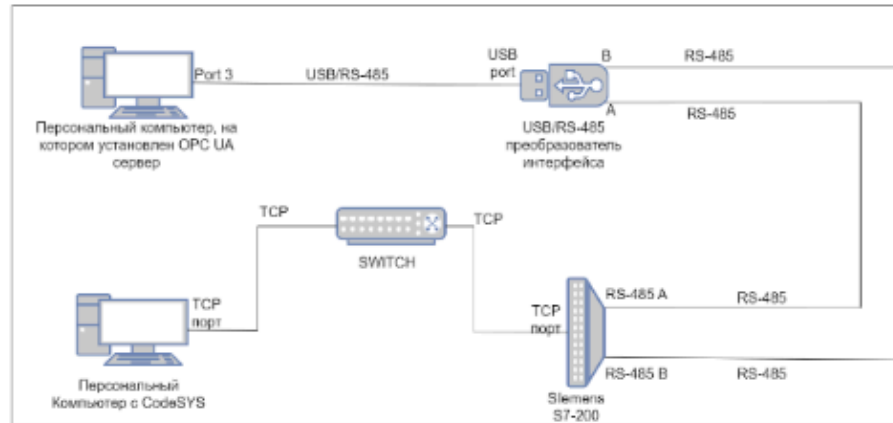
- 1 МЭК, 2015, IEC 62541-7:2015 (OPC unified architecture – Part 7: Profiles [Электронный ресурс]. – Режим доступа: <https://webstore.iec.ch/publication/21992>. – 16.03.2024
- 2 МЭК, 2015, IEC 62541-5:2015 OPC unified architecture – Part 5: Information model [Электронный ресурс]. – Режим доступа: <https://webstore.iec.ch/publication/21994>. – 16.03.2024
- 3 OPC Foundation, Спецификация OPC UA, Binary Data Encoding [Электронный ресурс]. – Режим доступа: <https://reference.opcfoundation.org/Core/Part6/v104/docs/5.2#:~:text=The%20OPC%20UA%20Binary%20DataEncoding,read%20from%20a%20binary%20stream>. – 16.03.2024
- 4 Microsoft, 1998 [MS-DCOM]: Distributed Component Object Model (DCOM) Remote Protocol. [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-dcom/86b9cf84-df2e-4f0b-ac22-1b957627e1ca. – 12.04.2024)
- 5 МЭК, 2015, IEC 62541-3:2015 (OPC unified architecture – Part 3: Address space model [Электронный ресурс]. – Режим доступа: <https://webstore.iec.ch/publication/21996>. – 16.03.2024
- 6 МЭК, 2015, IEC 62541-6:2015 OPC unified architecture – Part 6: Mappings [Электронный ресурс]. – Режим доступа: <https://webstore.iec.ch/publication/21993>. – 16.03.2024
- 7 МЭК, 2015, IEC 62541-4:2015 OPC unified architecture – Part 4: Services [Электронный ресурс]. – Режим доступа: <https://webstore.iec.ch/publication/21995>. – 16.03.2024
- 8 МЭК, IEC 62541-12 OPC unified architecture – Part 12: Discovery [Электронный ресурс]. – Режим доступа: <https://webstore.iec.ch/publication/29087>. – 16.03.2024

- 9 Федеральное агентство по техническому регулированию и метрологии, предварительный национальный стандарт российской федерации. Умноное производство. Унифицированная архитектура OPC [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200174815>. – 16.03.2024
- 10 Издание Control Engineering от 8 августа 2014-го года. [Электронный ресурс]. – Режим доступа: <https://controlengrussia.com/> . – 22.04.2024
- 11 Инсат, 2017, Руководство пользователя MasterSCADA версия 3.X . – [Электронный ресурс]. – Режим доступа: https://drive.google.com/uc?export=download&id=1Fz7auGhH-kIkWTZEXc3Y6_LoYmgKWFe5 . – 17.05.2024
- 12 А.Н. Рыбалёв, программируемые логические контроллеры и аппаратура управления лабораторный практикум. Часть 6. Приборы контроля положения исполнительных механизмов. [Текст], Благовещенск 2016, 70 страниц.
- 13 MODBUS, Modbus Technical Resources, [Электронный ресурс]. – Режим доступа: <https://www.modbus.org/tech.php> . – 22.05.2024
- 14 MODBUS, 2012, MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3, [Электронный ресурс]. – Режим доступа: https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf. – 22.05.2024
- 15 MODBUS, Modbus Specifications and Implementation Guides, [Электронный ресурс]. – Режим доступа: <https://www.modbus.org/specs.php>. – 22.05.2024
- 16 MODBUS.ORG, 2006, MODBUS over Serial Line Specification and Implementation Guide V1.02 , [Электронный ресурс]. – Режим доступа: https://modbus.org/docs/Modbus_over_serial_line_V1_02.pdf. – 22.05.2024
- 17 MODICON, Inc., Industrial Automation Systems One High Street North Andover, Massachusetts 01845, June 1996, Modicon Modbus Protocol Ref-

erence Guide , [Электронный ресурс]. – Режим доступа:
https://modbus.org/docs/PI_MBUS_300.pdf. – 17.05.2024

- 18 Приказ Минэнерго РФ «Правила Технической эксплуатации электроустановок потребителей» [Текст]. – Москва, 2017. – 340 страниц
- 19 Федеральный закон от 26.03.1998 № 41-ФЗ [Текст]. – Москва, 1998. – 120 страниц
- 20 Федеральный Закон о техническом регламенте и требованиях к пожарной безопасности от 30 Апреля 2009-го года. [Текст]. – Москва, 1998. – 67 страниц

Схема подключения



Настройка коммуникационного узла

Редактирование коммуникационного узла

Имя узла: Node1

- Общие настройки**
 - Комментарий
 - Включен в работу: True
 - Тип узла: COM
- Настройки COM**
 - Порт: 3
 - Скорость: 9600
 - Данные: 8
 - Контроль четности: Не используется
 - Стоп биты: 1
 - Максимальный таймаут (мс): 0
 - Использовать режим ASCII: False
 - Использовать модем: False
- Скрипт**
 - Выполнение скрипта: False
- Дополнительные настройки**
 - Слабое подключение: False
 - Принудительная разрыв соединения в каждом цикле: False

Настройка устройства

Редактирование устройства

Имя устройства: Device1

- Общие настройки**
 - Комментарий
 - Включен в работу: True
 - Тип устройства: MODBUS
 - Адрес: (3000) 0
 - Время сканирования (мс): 1000
 - Повтор при ошибке: 3
 - Сброс номера канала при разрыве соединения: True
 - Повторное сканирование после ошибки канала (с): 10
 - Реинициализация узла при ошибке: False
 - Порядок сканирования: 1000
 - Размерность команды адреса: 8
 - Исходная фаза: 0
 - Реинициализация: 100
 - Старт после запуска: True
 - Задержка запроса после получения ответа (мс): 4
 - Перестановка байтов в команде: выключ редактора перестановки байтов...
- Скрипт**
 - Выполнение скрипта: False
- Настройка запросов**
 - Максимальное количество команд в запросе чтения: 128
 - Максимальное количество битов (команда) в запросе чтения: 128
 - Использовать команду WRITE_SINGLE_COIL (RW5): True
 - Использовать команду WRITE_SINGLE_REGISTER (RW6): True
 - Максимальное допустимое значение в запросе чтения: 0
 - Использовать триггер: False
 - Использовать реинициализацию: False

Настройка параметров тега B

Редактирование тега

Имя тега: B

- Общие настройки**
 - Комментарий
 - Включен в работу: True
 - Режим: HOLDING_REGISTER
 - Адрес: (0x0001) 1
 - Тип данных в устройстве: int16
 - Тип данных в сервере: int32
 - Тип доступа: Read/Write
 - Использовать перестановку байтов устройства: True
 - Последний тег в групповом запросе: False
 - Порядок (A*16 + B): False
- Скрипт**
 - Разрешение выполнения скрипта после чтения: False
 - Разрешение выполнения скрипта перед записью: False
- Дополнительно**
 - Изменение бита из данных: False
 - Наличие отдельного регистра записи: False
 - Чтение сразу после записи: False
 - Обрат команда записи: True
 - Предупреждение записи командой 0: False
- HDA**
 - HDA доступ: True
 - Количество записей в архиве (100 - 4096): 2000
 - Автоматическая запись: False
 - Запись по изменению значащих битов: True

Окна MasterSCADA 3.7

The screenshot shows the MasterSCADA 3.7 interface with four numbered red boxes highlighting specific areas: 1 (top left), 2 (bottom left), 3 (top right), and 4 (bottom right).

Настройка OPC сервера

Настройка OPC UA клиента

Настройка подключения: **дополнительно**

Информация о сервере

Сервер: oesr://127.0.0.1:53000 Найти

+ Добавить резервный ✗ Удалить резервный

Настройки безопасности

Политика безопасности: None

Режим безопасности сообщений: None

Настройки аутентификации

Анонимно

Имя пользователя: MastersCADUSER

Пароль:

Настройка сессии

Имя сессии: Session 1

✓ OK ✗ Отмена ↻ Справка

Окна Master OPC Universal Modbus Server

Master OPC Universal Modbus Server

Мастер OPC Universal Modbus Server (B200) - USB

1

2

3

Общие настройки

Скрипт

- Общие настройки
 - Комментарий
 - Получен в работу: True
 - Парольный доступ к редактору: False
 - Буферный режим: Отключить/Включить: True
 - Кристалл: 8
 - Таймаут драйвера: 100
- Модуль
 - Разрешение доступа: False
- OPCUA Сервер
 - Отображение адреса при отсутствии клиента: False
 - Имя: (не задано)
 - Парольный доступ: False
 - При чтении архива всегда создавать Событийный файл: True
 - Контроль паролем: True
 - Валидация запросов (с): 18
 - Имя файла: False

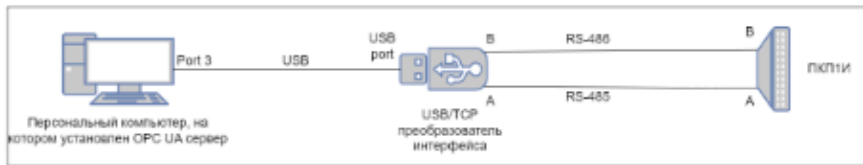
Область клиента

Тег: 1 Адрес: 0 Значение: 0

Имя: Конфигурирование

				ВКР.204-012.150304.П.П		
Имя/Лист	№ докум.	Лист	Дата	Методика выполнения лабораторной работы №1		Лист 1 / Листов 6
Разраб.	Исполнитель Т.В.	Провер.	2024.04.02			
Лист	Сервис ОВ	Лист	2024.04.02			
Техник	Сервис ОВ	Лист	2024.04.02			
Исполн.	Сервис ОВ	Лист	2024.04.02			
Студ.	Сервис ОВ	Лист	2024.04.02			
				Инициальное надзорное систем управления на базе технологии OPC		
				Формат А1		

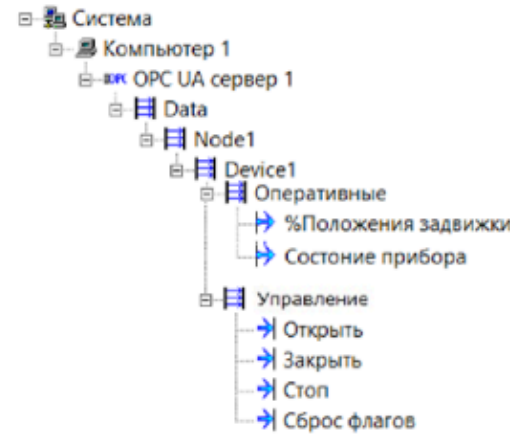
Схема подключения



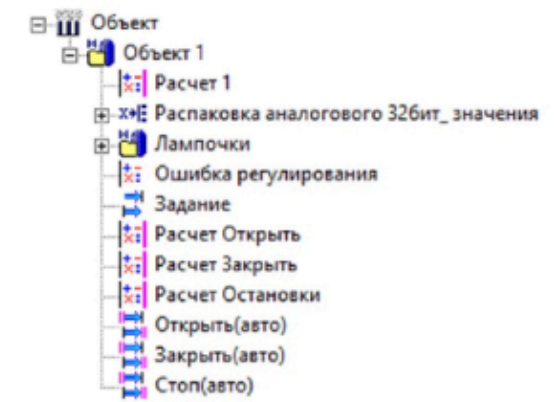
Библиотека визуально-функциональных блоков

Библиотеки	X=У и A=B (строковый)
ЦаЕ-Гал 6ЕаЕ1в@g	x Абсолютное значение X
Примеры скриптов	Sa Анализ строки
Обработка сигналов	^a Выделение знака
Внутренние	st Выделение подстроки
Исполнительные механизмы	hN Деление с доп. выходами
Датчики	q-? Конкатенация строк
Аппараты	и Логическое И
Вычисления	mM Логическое ИЛИ
Генераторы значений	kk Логическое Исключающее ИЛИ
Управление	ne Логическое НЕ
Работа с архивом	stt Обработка вызываемых объектов
Отраслевые библиотеки	S Разбор строки
Вентиляция (HVAC)	x-E Распаковка аналогового 32бит_значения
Электроэнергетика	s- Сравнение строк
Теплоэнергетика	z- Упаковка 32 дискретных значений
Служебные	St Формирование строки
Работа со временем	
Другие	

Дерево системы



Дерево объектов



Расчёт положения задвижки

Общие | Опрос | Опрос выхода | Связи | Формула | Архив | Контроль значений | Паспорт | Данные | Права дост

И | ИЛИ | НЕ | ЕСЛИ | () | & | | ~ | ^

Положения_задвижки / 10

Функции | Константы | Очистить

= < > >= <= + - * / div mod

Расчёт ошибки регулирования

Общие | Опрос | Опрос выхода | Формула | Архив | Контроль значений | Паспорт | Данные

И | ИЛИ | НЕ | ЕСЛИ | () | & | | ~ | ^

Расчет_1 - Задание

= < > >= <= + - * / div mod

Расчёт открытия

Общие | Опрос | Опрос выхода | Связи | Формула | Архив | Паспорт | Данные | Права доступа

И | ИЛИ | НЕ | ЕСЛИ | () | & | | ~ | ^

Ошибка_регулирования < -3

= < > >= <= + - * / div mod

Мнемосхема

0 17 33 50 67 83 100

0.00 ед. изм.

Открыть Закрыть Стоп

0 25 50 75 100

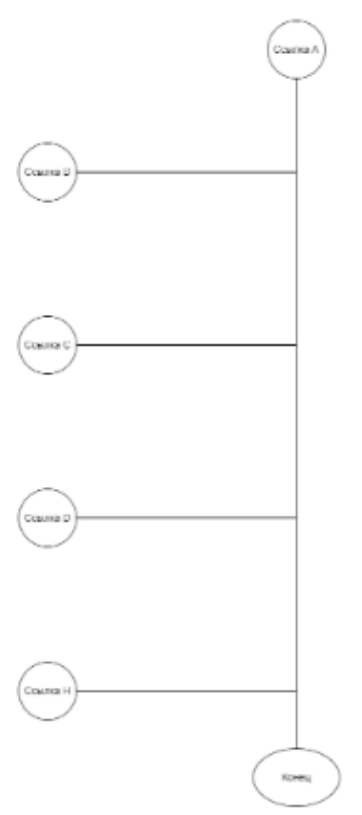
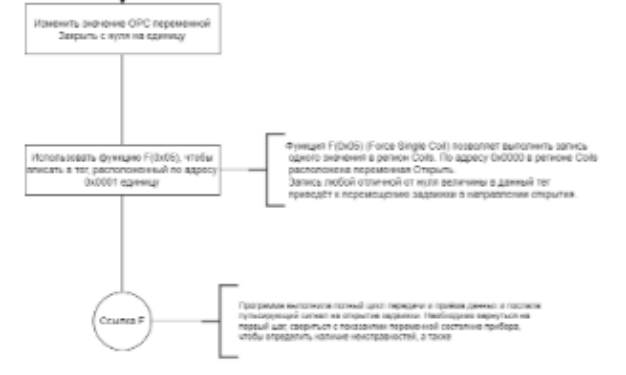
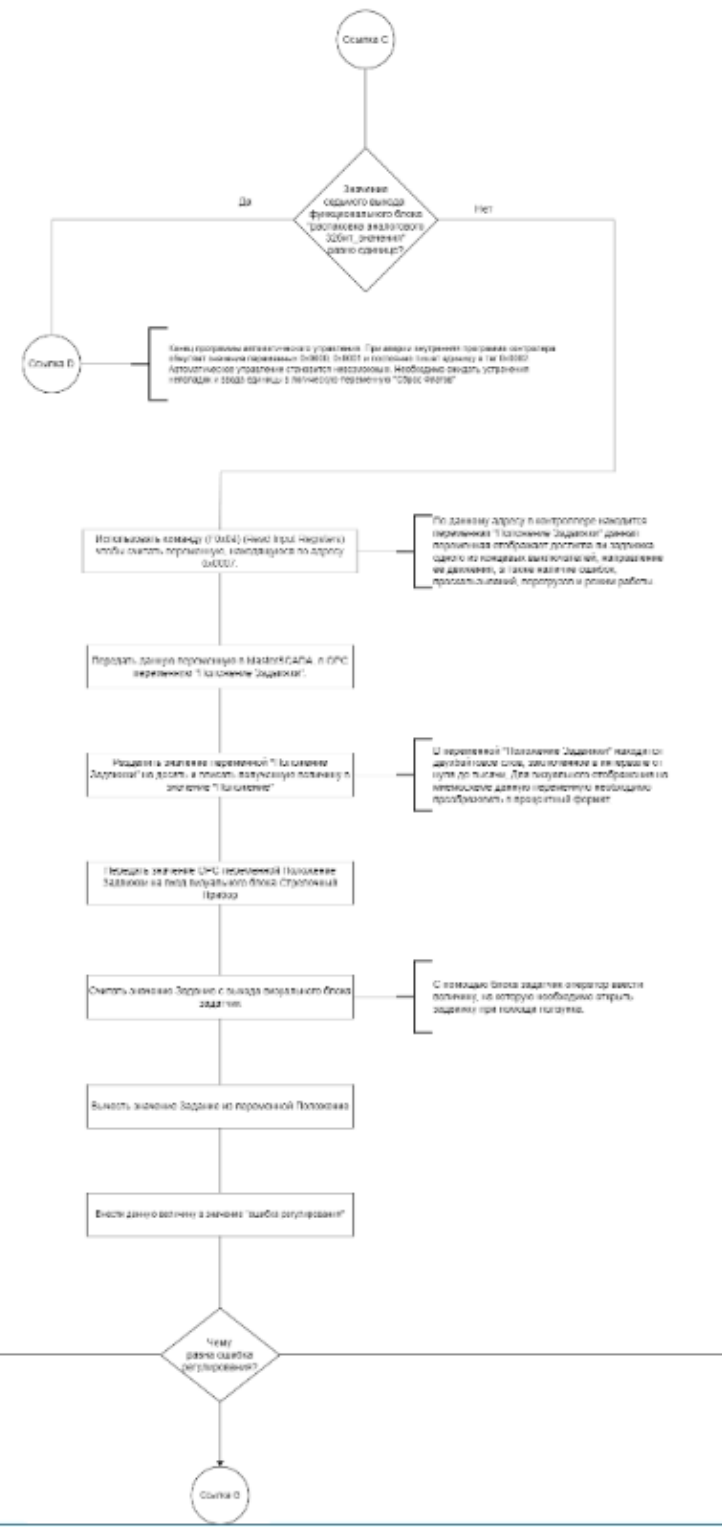
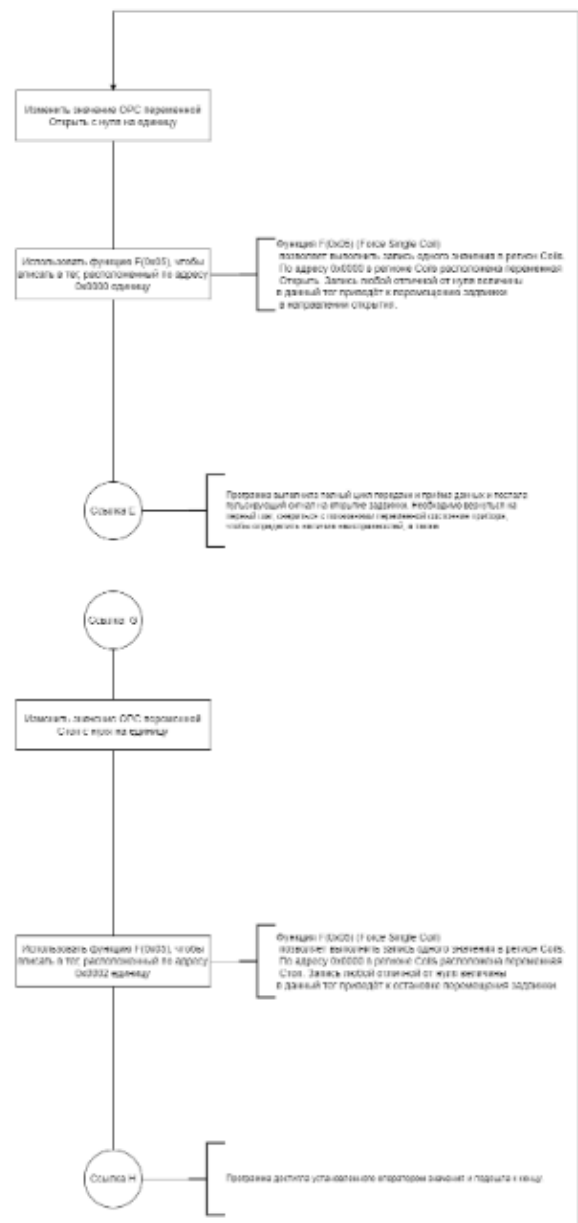
ед. изм.

0.00

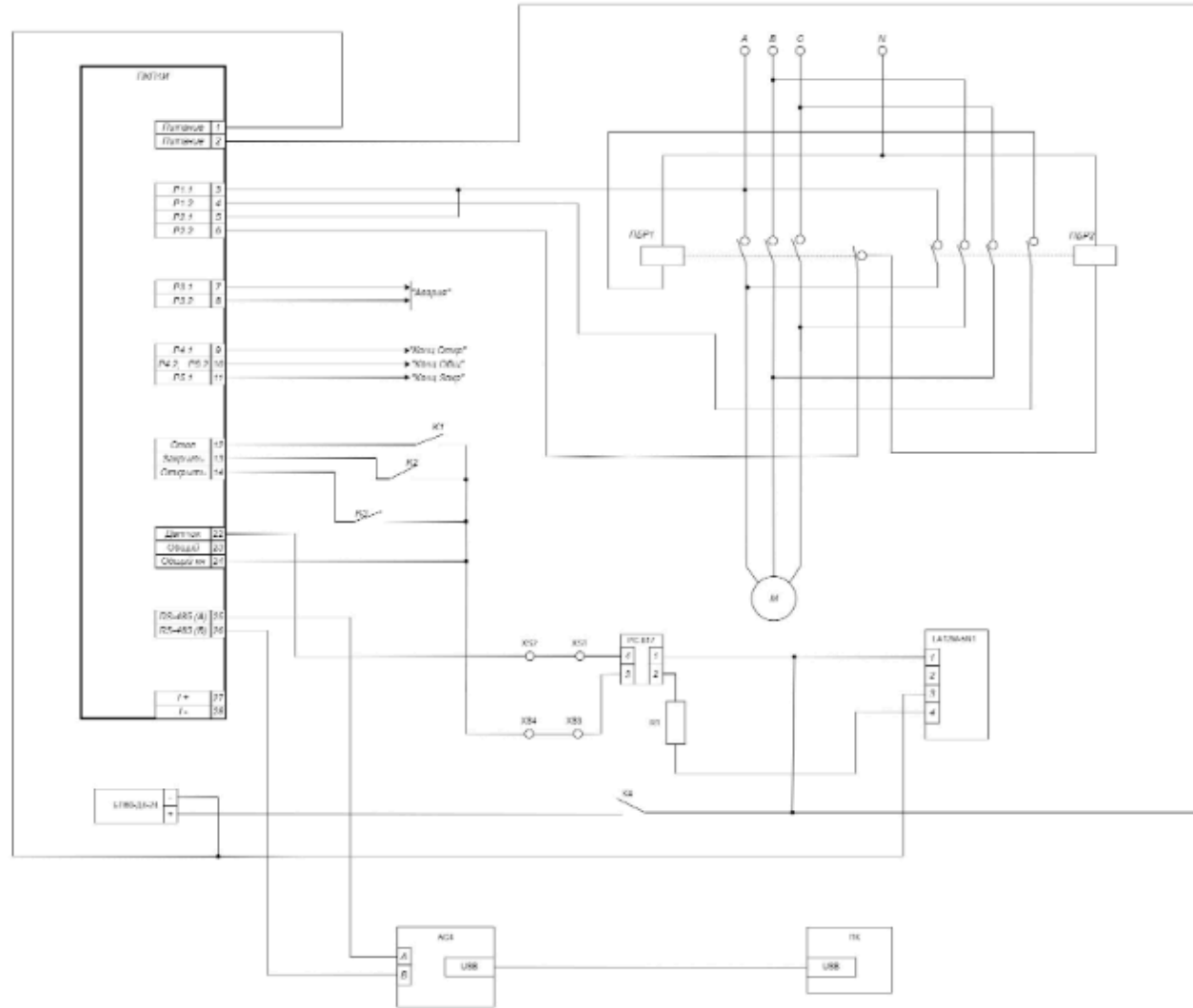
- Закрыто
- Закрывается
- Открывается
- Открыто
- Авария привода
- Перегруз
- Проскальзывание
- АВТО

				КВР.204.012.150304.ПЛ		
Методика выполнения лабораторной работы №2	Лист 2	Листов 6	Лист 2	Листов 6	Лист 2	Листов 6
Инициальное надлежание систем управления на базе технологий OPC						АМГУ, гр. 041 од
Копировать						47

Итого: 10 листов. В том числе: 10 листов. В том числе: 10 листов. В том числе: 10 листов.



				ВКР.204012.150304.CX		
Изм/Лист	№ Докум.	Лист	Дата	Блок-схема алгоритм		
Разраб	Михайлов ИВ	18	11.08.11	Лист	Масса	Масштаб
Проб	Сирота ОВ	18	11.08.11	у		
Технар	Сирота ОВ	18	11.08.11	Лист 4 из 6		
Исполн	Сирота ОВ	18	11.08.11	Инициальное надобление систем		
Знат	Сирота ОВ	18	11.08.11	управления на базе технологии OPC		
				АМГУ, гр 041 од		



Обозн.	Адрес	Обозначение	Наименование	Кол.	Примеч.
			Резисторы		
		R1	Индуктор сопротивления 480 Ом	1	
			Вычислительные устройства		
		ПК/ПЛИ	ОВЕН ПК/ПЛИ	1	
		ПК	Ноутбук MSI		
			Исполнительные механизмы		
		ПЕР12 М	Исполнительный электрический прибор МЭО 40/83-0.63-82	2	
			Устройства сложения		
		PC 817 LA12M-SN1	Транзисторный силовой PC 817 КИРРОВОР LA12M-SN1	1	
			Источники питания		
		БП60-Д4-24	ОВЕН БП60-Д4-24	1	
			Переключатели		
		К1-К4	4SWD-13D	4	

ВКР.204012.150304.СХ

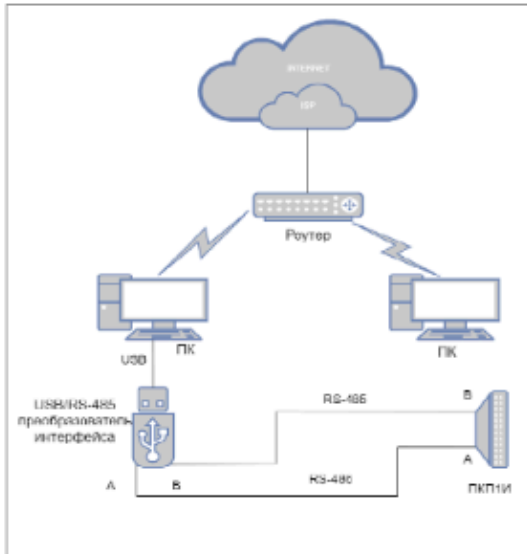
Изм./Лист	№ Эскиз	Лист	Этап	Итого	Лист	Масса	Масштаб
Разработ	Муромов Т.А.	1/1	Схемат	1/1	5		
Проект	Сварко Д.В.	2/2	Схемат	2/2	5		
Технический	Сварко Д.В.	3/3	Схемат	3/3	5		
Исполнитель	Сварко Д.В.	4/4	Схемат	4/4	5		
Смет	Сварко Д.В.	5/5	Схемат	5/5	5		

Принципиальная схема
стенда управления задвижкой

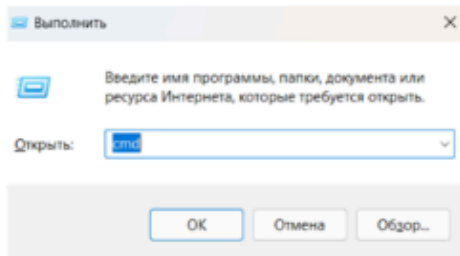
Инициальное моделирование систем
управления на базе технологии OPC

Лист 5
Листов 6
АМГУ, гр. 041 об

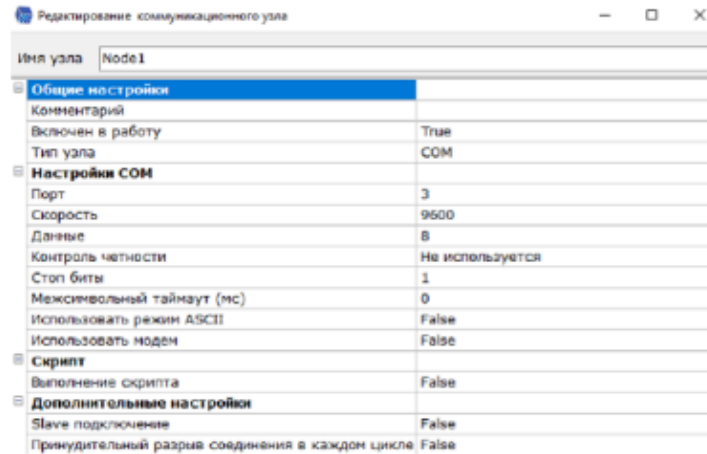
Схема соединения



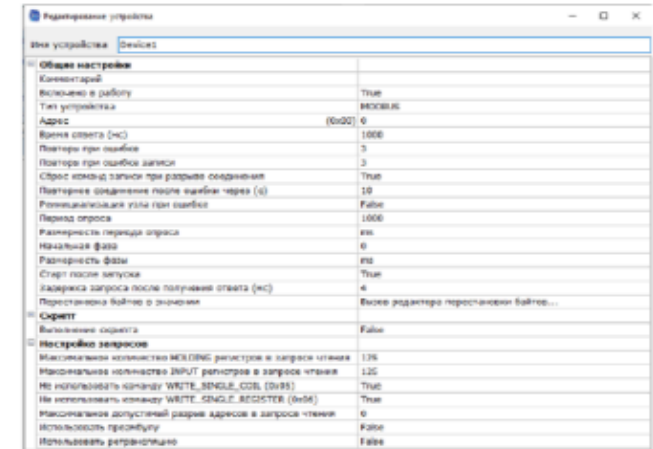
Вызов командной строки



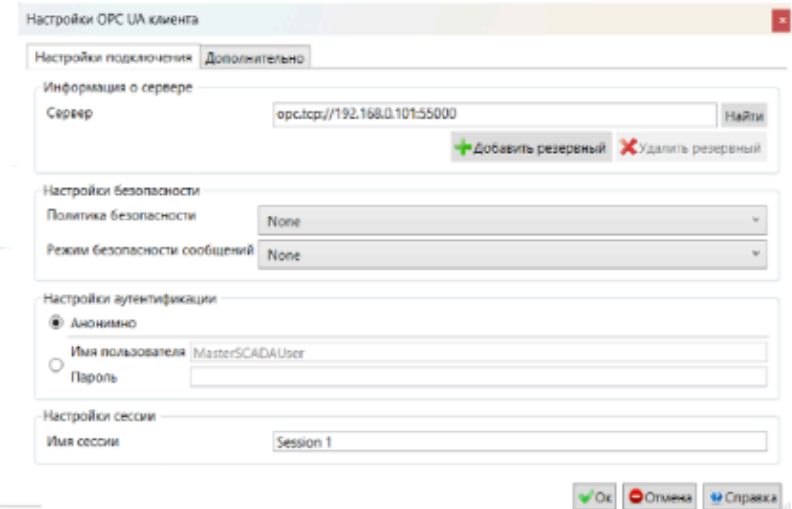
Настройка параметров узла



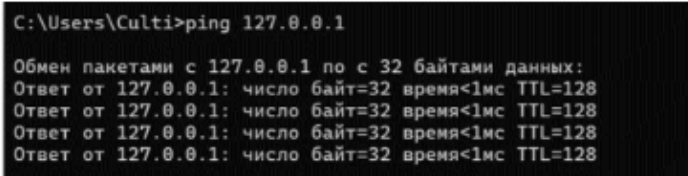
Настройка параметров устройства



Настройка OPC UA клиента



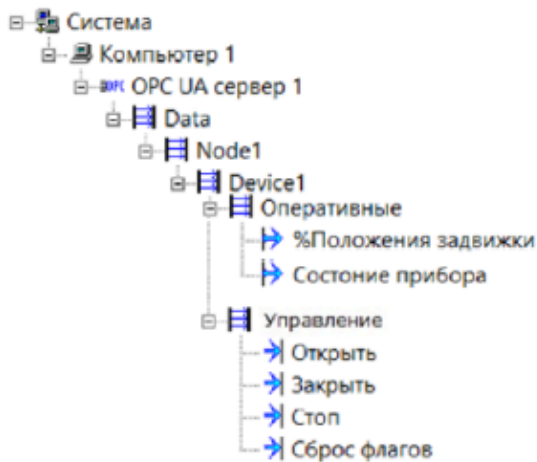
Команда PING



Тренд



Дерево системы



				ВКР.204012.150304.ПЛ		
Имя/Лист	№ документа	Полн.	Дата	Лист	Масса	Масштаб
Разработ.	Муромов Т.В.	И.С.Х.	11.05.2014	6		
Проект.	Сережа О.В.	И.С.Х.		6		
Утверд.	Сережа О.В.	И.С.Х.		6		
Исполн.	Сережа О.В.	И.С.Х.		6		
Стат.	Сережа О.В.	И.С.Х.		6		

Методика выполнения лабораторной работы №3
Имитационное моделирование систем управления на базе технологии OPC
Лист 6 из 6
АМГУ, гр. 041 од