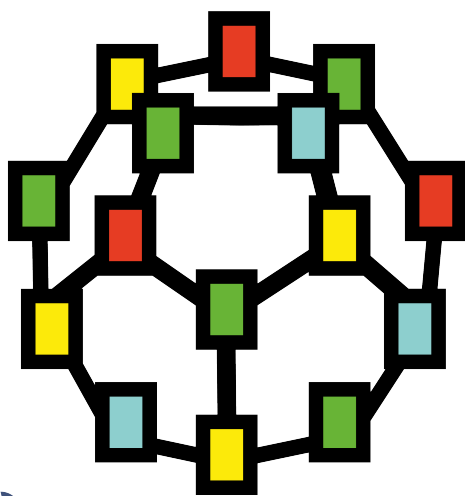




Библиотека раIЕС850



Руководство пользователя

07.2024
версия 1.2

Содержание

Используемые термины и сокращения	3
Введение	4
1 Общие сведения	5
1.1 Основные сведения о стандартах МЭК 61850	5
1.2 Информационная модель подстанции	5
1.3 Сервисы ACSI	7
2 Библиотека raIEC850	8
2.1 Сервер МЭК 61850 (IEC61850srv)	9
2.1.1 Свойства узлов дерева описания устройств IEC61850srv	13
2.2 Клиент МЭК 61850 (IEC61850cli)	14
2.2.1 Свойства узлов дерева описания устройств IEC61850cli	17
2.2.2 Вспомогательный модуль выполнения команды (IEC61850CmdCSWI)	19
2.3 Квалификатор (IEC61850BitStrToQuality/IEC61850TransQuality)	19
3 Пример настройки обмена по протоколу МЭК 61850 в режиме сервера	21
3.1 Настройка логических узлов LN	22
3.1.1 Настройка контролируемого логического узла класса SPC	22
3.1.2 Настройка логического узла класса MV – измеряемая величина	25
3.1.3 Настройка логического узла класса SPS – состояние	29
3.1.4 Настройка пользовательского логического узла	31
3.2 Формирование наборов данных DS	33
3.3 Обмен данными с клиентом IEDScout	35
3.4 Обмен данными с клиентом Multi-Protocol MasterOPC Server	44
4 Пример настройки обмена по протоколу МЭК 61850 в режиме клиента	48
5 Диагностика обмена	56
5.1 Диагностика обмена сервера IEC61850srv	56
5.2 Диагностика обмена клиента IEC61850cli	56
ПРИЛОЖЕНИЕ А. Поддерживаемые сервисы ACSI	60
ПРИЛОЖЕНИЕ Б. Функциональные ограничения (Functional Constraints)	62

Используемые термины и сокращения

АРМ – автоматизированное рабочее место.

ОС – операционная система.

ПЛК – программируемый логический контроллер.

ТН – трансформатор напряжения.

ТС – телесигнализация.

ТТ – трансформатор тока.

ТУ – телеуправление.

Физическое устройство – устройство, эквивалентное **IED**.

ACSI (Abstract Communication Service Interface) – абстрактный интерфейс услуг связи.

DA (Data Attribute) – атрибут данных.

DO (Data Object) – объект данных.

DS (Data Set) – набор данных (отчет).

FC (Functional Constraint) – функциональное ограничение. Указывает те сервисы, с которыми разрешено работать на определенном атрибуте данных **DA**.

GOOSE (Generic Object Oriented Substation Events) – объектно-ориентированный протокол для связи между устройствами релейной защиты и автоматики.

IED (Intelligent Electronic Device) – любое устройство, имеющее один или несколько процессоров и способное получать или направлять данные и управляющие воздействия от внешнего источника либо на внешний источник.

LD (Logical Device) – логическое устройство – объект, представляющий набор типичных функций подстанции.

LN (Logical Node) – логический узел – наименьшая часть функции, обменивающаяся данными.

MMS (Manufacturing Message Specification) – протокол обмена сообщениями стандарта ISO 9506.

SBO (Select Before Operate) – выбор перед выполнением.

SCL (Substation Configuration Language) – язык описания конфигурации подстанции.

SV (Sampled Value) – протокол передачи мгновенных значений тока и напряжения.

Введение

Настоящее руководство описывает настройку обмена данными по протоколу **МЭК 61850** для контроллеров ОВЕН, программируемых в среде **Полигон**. Предполагается, что читатель обладает базовыми навыками работы с **Полигон**, поэтому общие вопросы (например, создание и загрузка проектов) в данном документе не рассматриваются – они подробно описаны в документах [Руководство по программированию](#), [Библиотека raCore](#) и [Быстрый старт](#).

Настройка обмена данными по протоколу **МЭК 61850** в среде **Полигон** осуществляется с помощью функциональных блоков из библиотеки **paIEC850**. Данная библиотека доступна для работы при наличии соответствующей лицензии runtime (см. описание лицензионных пакетов [на странице среды разработки Полигон](#)).

Примеры в документе актуальны для версии среды **Полигон – 1917**, версии библиотеки **paIEC850 – 81** и выше.

1 Общие сведения

1.1 Основные сведения о стандартах МЭК 61850

МЭК 61850 – набор стандартов «Сети и системы связи на подстанциях», которые описывают архитектуру сетей на подстанциях. Эти стандарты обеспечивают совместимость работы АСУ-оборудования различных производителей.

В **МЭК 61850** определены абстрактные сервисы взаимодействия и классы данных так, чтобы эти спецификации не зависели от конкретных стеков протоколов, интерфейсов и ОС. Стандарт также определяет требования к надежности передачи данных.

Первая часть стандарта была опубликована в 2003 году.

МЭК 61850 состоит из разделов:

- **1 Introduction and overview** – введение и общий обзор;
- **2 Glossary** – глоссарий терминов;
- **3 General requirements** – основные требования;
- **4 System and project management** – управление системой и проектированием;
- **5 Communication requirements for functions and device models** – требования к связи для функций и моделей устройств;
- **6 Configuration language for communication in electrical substations related to IEDs** – язык описания конфигурации связи между микропроцессорными электронными устройствами подстанций;
- **7 Basic communication structure – Principles and models** – основная структура связи для оборудования подстанции и питающей линии (4 части);
- **8-1 Specific communication service mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3** – описание специфического сервиса связи (SCSM) – описание передачи данных по протоколу MMS (ИСО/МЭК 9506 – часть 1 и часть 2) и по протоколу ИСО/МЭК 8802-3;
- **9-2 Specific communication service mapping (SCSM) – Sampled values over ISO/IEC 8802-3** – описание специфического сервиса связи (SCSM) – выборочные значения по ИСО/МЭК 8802-3;
- **10 Conformance testing** – проверка на совместимость.

В качестве общей шины для обмена по протоколам **МЭК 61850** выбрана сеть [Ethernet](#).

1.2 Информационная модель подстанции

Одна из основных задач стандарта **МЭК 61850** – определение единого базового набора терминов и связывающей их информационной модели.

В стандарте есть понятие **логического узла LN** (Logical Node) – это наименьшая часть функции, которая может обмениваться данными.

Логические узлы разделены на группы. Всего стандарт предусматривает девятнадцать групп логических узлов.

Таблица 1.1 – Основные группы логических узлов

Указатель группы	Имя группы
L	Системные логические узлы
P	Функции защит
R	Вспомогательные функции защит
C	Диспетчерское управление
G	Общие функции
I	Интерфейсы и архивирование
A	Автоматическое управление
M	Учет и измерения
S	Диспетчерское управление и мониторинг
X	Коммутационные аппараты
T	Измерительные трансформаторы и датчики
Y	Силовые трансформаторы и связанные функции
Z	Резерв

Обозначение каждого логического узла состоит из четырех символов: первый символ обозначает группу логического узла, остальные три формируются как аббревиатура обозначаемой функции.

Например, **PDIF** – дифференциальная защита, **XCBR** – выключатель, **MMXU** – измерительное устройство и т. п.

Каждый логический узел содержит один или несколько объектов данных, каждый объект данных имеет стандартизованную структуру.

Например, логический узел **XCBR** содержит такие обязательные объекты данных, как:

- **Loc** – режим управления (локальный/дистанционный);
- **OpCnt** – счетчик операций;
- **Pos** – положение;
- **BlkOpn** – блокировка команд отключения выключателя;
- **BlkCls** – блокировка команд включения выключателя;
- **CBOPCap** – возможность срабатывания выключателя.

Для объектов данных задается функциональное ограничение **FC** (Functional Constraint), которое определяет, какие сервисы **ACSI** применимы к этому объекту данных. Например, для объекта **Loc** задано функциональное ограничение **ST** – информация о состоянии. Соответственно, данный объект клиент может читать, но не может записывать.

Список основных функциональных ограничений по **ГОСТ Р МЭК 61850-7-2**, используемых в среде **Полигон**, приведен в **Приложении Б**.

Объект данных **Loc** имеет структуру класса **SPS** (Single Point State), которая в свою очередь содержит такие обязательные атрибуты, как:

- **stVal** – состояние (bool);
- **q** – качество;
- **t** – временная метка.

Таким образом, полное уникальное имя атрибута данных внутри логического устройства **LD** (Logical Device) состоит из следующих полей:

- Имя логического устройства **LD**;
- Логический узел **LN**;
- Функциональное ограничение **FC**;
- Объект данных;
- Атрибут.



Рисунок 1.1 – Полное имя атрибута данных

Для описания информационной модели **МЭК 61850** используется стандартизированный язык **SCL** (Substation Configuration Language), основанный на языке разметки **XML**.

Основные типы файлов **SCL**:

- **.ssd** – топология подстанции (однолинейная схема);
- **.icd** – функциональные возможности интеллектуального электронного устройства **IED** (Intelligent Electronic Device);
- **.cid** – описание функционала **IED** (для каждого устройства свой файл, поставляемый производителем);
- **.scd** – полное описание подстанции.

1.3 Сервисы ACSI

Модели абстрактного интерфейса сервиса связи **ACSI** (Abstract Communication Service Interface) по стандарту **МЭК 61850** определяют набор сервисов, которые позволяют **IED** функционировать в сети аналогично соответствующим функциям энергосистемы.

Стандарт **МЭК 61850** описывает преобразование абстрактных сервисов в сетевые протоколы.

В стандарте предусматривается два вида цифровых шин:

- Шина подстанции – обмен между АРМ, серверами и **IED**;
- Шина процесса – обмен между полевыми устройствами и **IED**.

МЭК 61850 описывает следующие протоколы сетевого взаимодействия:

- **MMS** – клиент-серверная технология (АРМ – клиент, **IED** – сервер), обеспечивающая обмен некритичными ко времени передачи данными (уровень шины подстанции);
- **GOOSE** – технология, основанная на модели подписчик-издатель, для обмена критичными ко времени передачи (высокоприоритетными) данными между **IED** (уровень шины подстанции);
- **SV (Sampled Value)** – цифровая информация от ТТ и ТН (уровень шины процесса).

Для организации обмена между устройствами используются наборы данных **DS** (DataSet). Каждый **DS** имеет свое имя, в него помещаются ссылки на передаваемые атрибуты. Такие наборы данных используются для формирования отчетов для передачи на АРМ операторов или организации обмена между **IED**.

Для осуществления телеуправления стандарт определяет следующие модели управления:

- Сценарий 1: Прямое управление с нормальной безопасностью (**direct-operate**);
- Сценарий 2: SBO управление с нормальной безопасностью (**operate-once** или **operate-many**);
- Сценарий 3: Прямое управление с улучшенным уровнем безопасности (**direct-operate**);
- Сценарий 4: SBO управление с улучшенным уровнем безопасности (**operate-once** или **operate-many**).

Поддерживаемые сервисы **ACSI** по **ГОСТ Р МЭК 61850-7-2** в среде **Полигон** приведены в [Приложении А](#).

2 Библиотека paEC850

Библиотека **paEC850** содержит функциональные блоки для реализации обмена по протоколу **МЭК 61850**.

В структуру библиотеки входят блоки **IEC61850cli** и **IEC61850srv**, с помощью которых настраивают обмен по протоколу **МЭК 61850** в режимах **клиент** и **сервер**.

Остальные блоки библиотеки являются вспомогательными.

Для добавления библиотеки **paEC850** в проект следует:

1. Перейти в меню **Окна/Проекты**. В появившемся окне отобразится текущий проект и добавленные библиотеки.

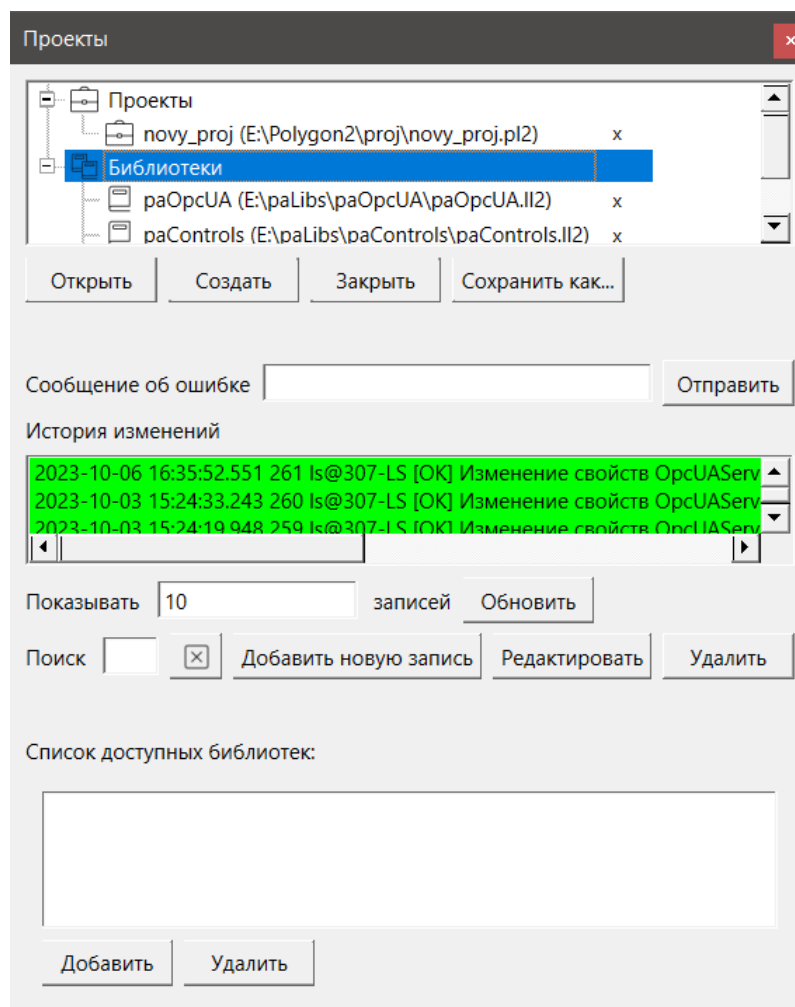


Рисунок 2.1 – Добавление библиотеки paEC850 в проект

2. Нажать кнопку **Открыть** и перейти в папку с файлами библиотеки, которую необходимо добавить. Затем в выпадающем списке выбрать тип файла **Библиотека Полигон 2 (*.II2)**.

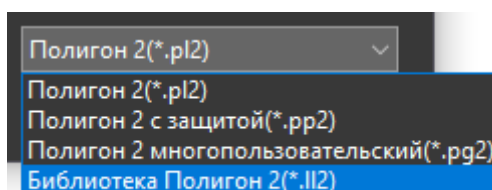


Рисунок 2.2 – Добавление библиотеки paEC850 в проект

3. В окне появится файл библиотеки с расширением **.II2**. Следует выбрать его и нажать **Открыть**.

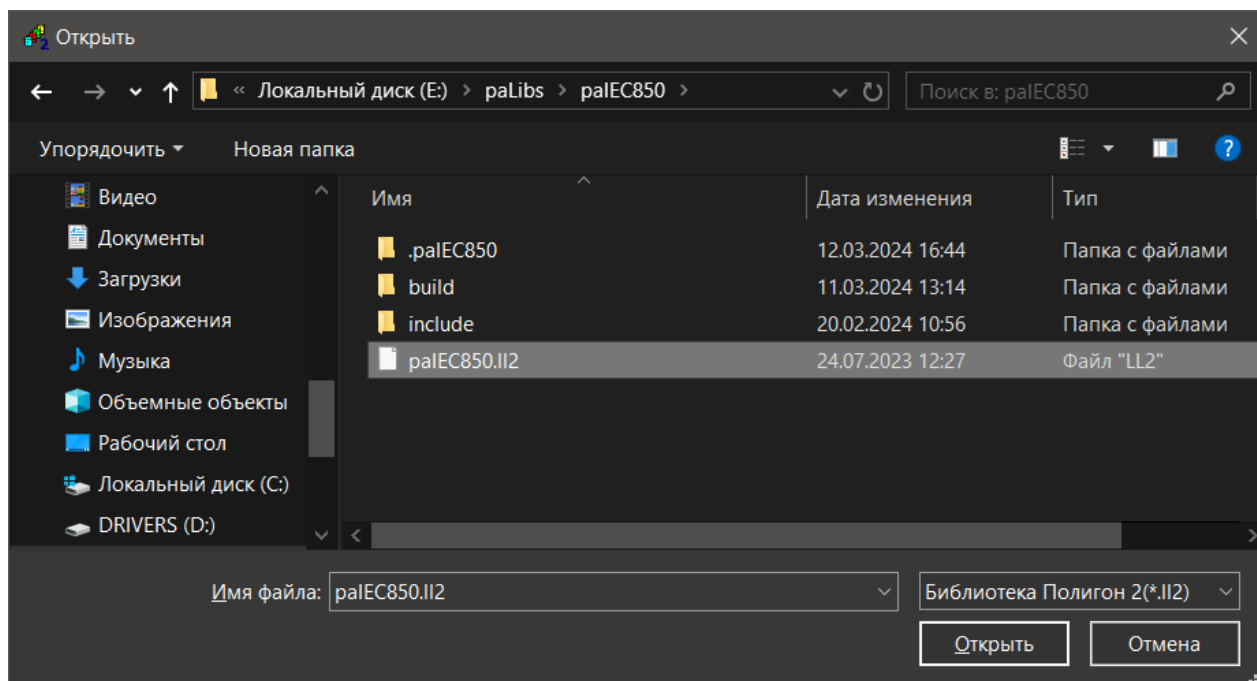


Рисунок 2.3 – Добавление библиотеки palEC850 в проект

Добавленная библиотека отобразится в окне *Проекты*.

2.1 Сервер МЭК 61850 (IEC61850srv)

Блок *IEC61850srv* реализует протокол стандарта **МЭК 61850 MMS** и выполняет роль сервера.

Данный блок следует размещать только в **Фоне**, потому что работа блока занимает значительное время.

Таблица 2.1 – Назначение входов и выходов IEC61850srv

Элемент	Описание
Входы	
enb	Разрешение работы блока
tmp	Период вызова вспомогательных функций блока, мс (см. рисунок 2.5)
sdr	Сетевой стек для ПЛК ОВЕН "I"
cfg	Конфигурация (битовая маска): Бит 4 – активирует принудительное обновление значений атрибутов mag из значения instMag . Таймаут задается соответствующим свойством в узле CIED
ctl	Управление диагностикой (битовая маска): Бит 7 – активировать диагностический модуль (для разработчиков); Биты 8...15 – битовая маска, определяющая, какие диагностические сообщения будут выводиться в системную консоль (установка 0xFF00 разрешает вывод всех типов сообщений, см. раздел 5.1)
Выходы	
sts	Статус устройства: 0 – нет связи с клиентом; 1 – соединение с клиентом установлено
dsts	Не используется

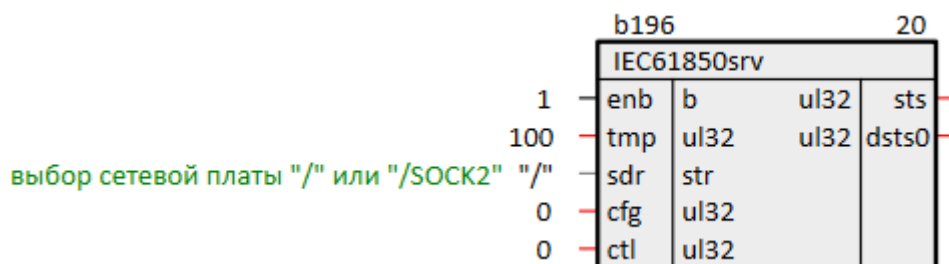


Рисунок 2.4 – Блок IEC61850srv

Работа протокола делится на три основные подпрограммы (см. рисунок 2.5).

При **enb = 0** происходит деактивация подпрограмм протокола.

Вход **tmp** определяет период вызова вспомогательных функций блока, выполняющихся в фоновом потоке.

На входе **cfg** можно задать **бит 4**. Если в модели данных есть классы **MV** (измеряемая величина), то при **cfg = 4** активируется принудительное обновление значения **mag** (измеряемая величина после мертвой зоны) из значения **instMag** (реальное значение измеряемого параметра). Это полезно, если измеряемая величина долгое время не изменяется из-за мертвой зоны.

Интервал обновления значений **mag** задается свойством в узле **CIED** (см. раздел 2.1.1).



Рисунок 2.5 – Работа IEC61850srv: разделение на потоки и последовательность вызова подпрограмм

Для функционирования протокола следует определить абстрактную модель данных внутри узла блока **IEC61850srv** в дереве проекта. Для этого применяются шаблоны из библиотеки **palEC850**.

Корневым узлом модели данных является физическое устройство **IED**. Внутри него добавляются логические устройства **LD** и логические узлы **LN**. Логические узлы, в свою очередь, содержат объекты данных с атрибутами.

Каждый узел в дереве проекта имеет по крайней мере один параметр (свойство) – **Имя**.

Для узла **IED** также задаются следующие свойства:

- **IP-адрес** – локальный IP-адрес сервера (если отличается от заданного в свойстве модуля);
- **Порт** – сетевой порт для подключения клиента;

- **МЭК61850: Таймаут принудительного обновления mag (мс)** – таймаут принудительного обновления значений атрибутов **mag** из значения **instMag** (должен быть активирован **бит 4** на входе **cfg**).

Свойство	Значение
IP адрес	"10.2.12.12"
Имя	IED01
Имя типа	CIED
МЭК61850: Таймаут принудительного обновления mag (мс)	60000
Номер	0
Порт	102
Индекс	34
Принадлежит	33

Рисунок 2.6 – Свойства узла IED

Для объекта данных дополнительно можно задать свойство **МЭК61850: функциональное ограничение**. Оно определяет, какие сервисы клиент может применять к данному объекту данных.

Свойство	Значение
Имя	MV0
Имя типа	CIEC61850DO
МЭК61850: функциональное ограничение	MX
Номер	0
Индекс	75
Принадлежит	74

Рисунок 2.7 – Свойства узла объекта данных

Для атрибута данных задаются свойства:

- **МЭК61850: базовый тип данных** – базовый тип данных атрибута, определяется явно протоколом;
- **МЭК61850: значение по умолчанию** – значение, присваиваемое данному атрибуту в начале работы протокола (при отсутствии проведенной связи к данному атрибуту).



ПРИМЕЧАНИЕ

В данной реализации протокола используются два вида атрибутов данных: структурированный (**Struct**) и строго типизированный (тип данных определяется протоколом).

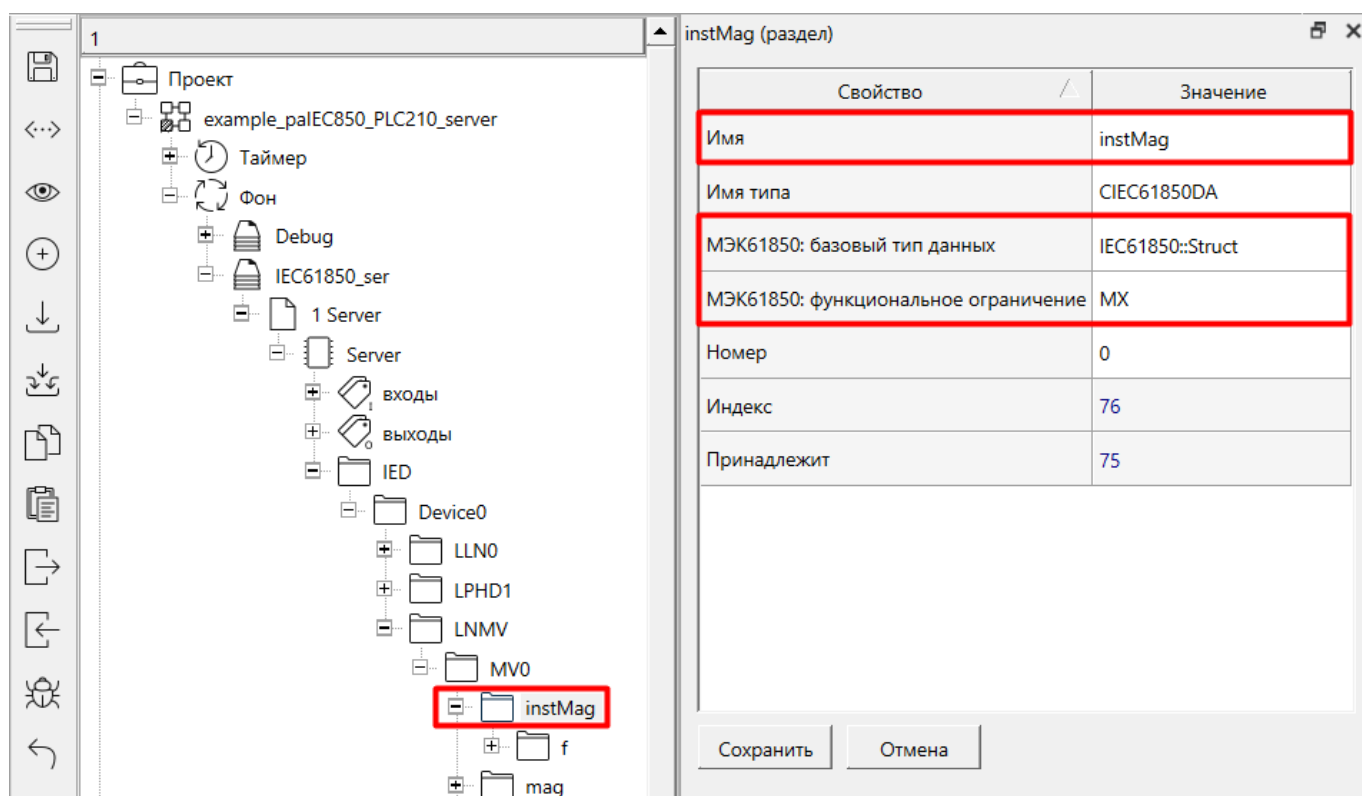


Рисунок 2.8 – Структурированный атрибут данных. Свойства

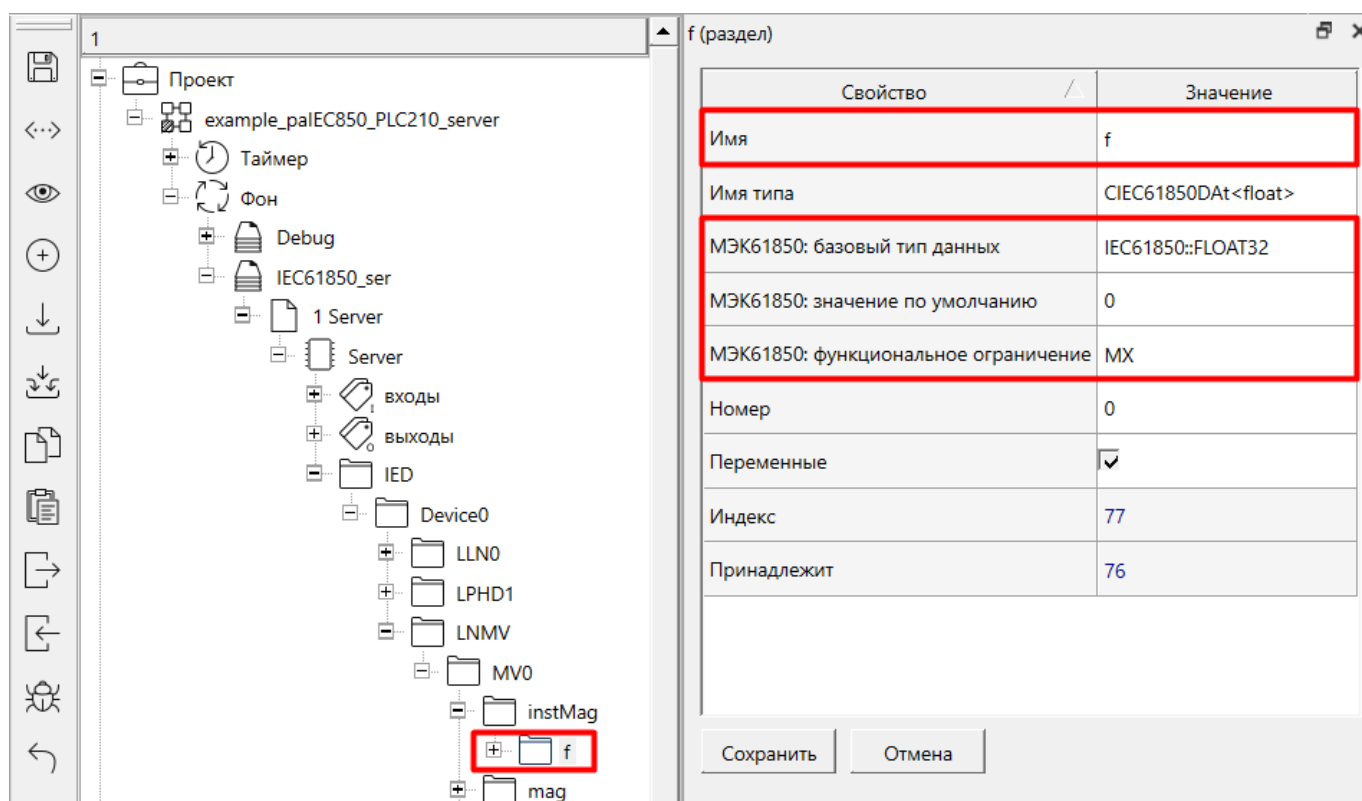


Рисунок 2.9 – Строго типизированный атрибут данных. Свойства

Для управляемых атрибутов данных дополнительно задается свойство **МЭК61850: модель управления**. Возможные значения см. в справке среды и в [разделе 2.1.1](#).

Для узла управления набором данных (отчетом) определяются причины передачи отчета. Активированные в сервере причины будут также активированы по умолчанию у клиента (при необходимости их можно поменять).

Свойство **МЭК61850: название набора данных** определяет, каким набором данных управляет данный узел.

Свойство **МЭК61850: буферированный отчет** включает буферизацию данных отчета. Если клиент потеряет связь с сервером, отчеты будут сохраняться в буфер и передадутся клиенту при возобновлении подключения.

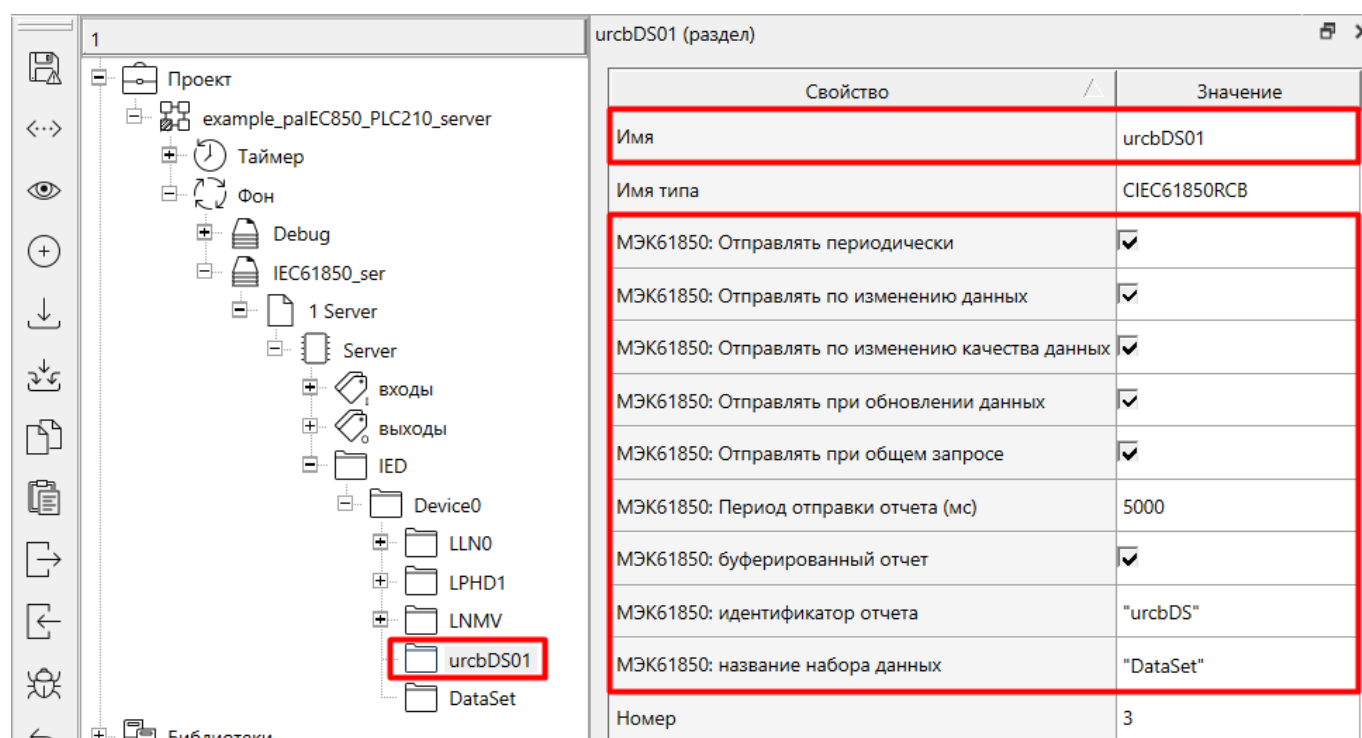


Рисунок 2.10 – Свойства узла управления набором данных (отчетом)

Перечень используемых свойств узлов дерева описания устройств **IEC61850srv** приведен в справке среды программирования и в [разделе 2.1.1](#).

Пример работы с блоком приведен в [разделе 3](#).

2.1.1 Свойства узлов дерева описания устройств IEC61850srv

В данном разделе приведены свойства узлов, используемые в дереве описания устройств блока [IEC61850srv](#).

Таблица 2.2 – Описание свойств узлов дерева описания устройств IEC61850srv

Свойство	Идентификатор свойства	Назначение
Класс CIED. Корневой узел дерева описания устройства		
IP адрес	prop_ip	Локальный сетевой адрес устройства 61850 (необязательное свойство)
Порт	prop_inter_port	Сетевой порт, на котором устройство ожидает подключения
МЭК61850: Таймаут принудительного обновления mag (мс)	prop_61850_mag_force_update_timeout	Таймаут принудительного обновления значений атрибутов mag из значения instMag (должен быть активирован бит 4 на входе cfg)
Класс CIEC61850DO. Объект данных		
МЭК61850: Функциональное ограничение	prop_FC	Функциональное ограничение, определенное протоколом
Класс CIEC61850DA. Атрибут данных		
МЭК61850: Функциональное ограничение	prop_FC	Функциональное ограничение, определенное протоколом
МЭК61850: Базовый тип данных	prop_bType	Базовый тип данных, определенный протоколом

Продолжение таблицы 2.2

Свойство	Идентификатор свойства	Назначение
МЭК61850: значение по умолчанию	prop_defVal	Значение, которое присваивается атрибуту в начале работы протокола (при отсутствии связи)
МЭК61850: Модель управления	prop_ctlModel	Модель управления, определенная протоколом. Только для управляемых атрибутов данных. Возможные значения: <ul style="list-style-type: none"> • status_only; • direct_with_normal_security; • sbo_with_normal_security; • direct_with_enhanced_security; • sbo_with_enhanced_security; • not_controllable
Класс IEC61850RCB. Блок управления набором данных (отчетом)		
МЭК61850: Отправлять по изменению данных	prop_trg_data_changed	Условие отправки отчета
МЭК61850: Отправлять при общем опросе	prop_trg_general_inter	Условие отправки отчета
МЭК61850: Отправлять при обновлении данных	prop_trg_general_inter	Условие отправки отчета
МЭК61850: Отправлять по изменению качества данных	prop_trg_qual_changed	Условие отправки отчета
МЭК61850: Отправлять периодически	prop_trg_integrity	Условие отправки отчета
МЭК61850: Период отправки отчета (мс)	prop_trg_integrity_period_ms	Период отправки отчета в случае установки свойства prop_trg_integrity
МЭК61850: Буферированный отчет	prop_isBuffered	Маркер буферированного отчета
МЭК61850: идентификатор отчета	prop_ReportID	Идентификатор отчета
МЭК61850: название набора данных	prop_dataSetName	Имя набора данных, которым управляет узел

2.2 Клиент МЭК 61850 (IEC61850cli)

Блок *IEC61850cli* реализует протокол стандарта **МЭК 61850 MMS** и выполняет роль клиента.

Так как работа блока занимает значительное время, может быть размещен только в **Фоне**.

Таблица 2.3 – Назначение входов и выходов IEC61850cli

Элемент	Описание
Входы	
enb	Разрешение работы блока
tmp	Период вызова вспомогательных функций блока, мс (см. рисунок 2.12)
sdr	Сетевой стек, для ПЛК ОВЕН "I"
cfg	Не используется
ctl	Управление (битовая маска): Бит 0 – инициировать команду общего опроса устройств 61850 ; Бит 7 – активировать диагностический модуль (для разработчиков); Биты 8...15 – битовая маска, определяющая, какие диагностические сообщения будут выводиться в системную консоль (установка 0xFF00 разрешает вывод всех типов сообщений, см. раздел 5.2)
bo	Вход для подключения вспомогательных блоков (циклический)

Продолжение таблицы 2.3

Элемент	Описание
Выходы	
sts	Не используется
dsts	Выход статуса устройства. Добавляется для каждого устройства 61850 (циклический): 0 – нет связи; 1 – есть связь

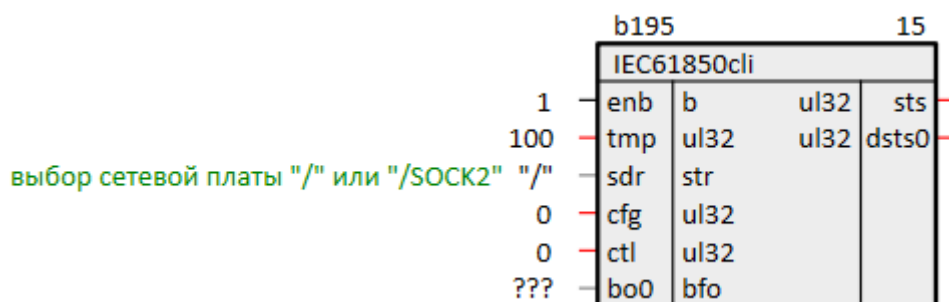


Рисунок 2.11 – Блок IEC61850cli

Работа протокола делится на три основные подпрограммы (см. рисунок 2.12).

При **enb = 0** происходит деактивация подпрограмм протокола.

Вход **tmp** определяет период вызова вспомогательных функций блока, выполняющихся в фоновом потоке.

Бит 0 на входе **ctl** активирует общий опрос всех устройств в дереве клиента.

Циклические входы **bo** служат для подключения вспомогательных блоков.

Для каждого устройства в дереве клиента можно создать выход статуса устройства **dsts**.



Рисунок 2.12 – Работа IEC61850cli: разделение на потоки и последовательность вызова подпрограмм

Для функционирования протокола абстрактная модель данных сервера импортируется в клиент из файла конфигурации *.cid (предоставляется производителем устройства).

Корневым узлом модели данных является физическое устройство **IED**. Внутри него добавляются логические устройства **LD** и логические узлы **LN**. Логические узлы, в свою очередь, содержат объекты данных с атрибутами.

Для узла **IED** задаются следующие свойства:

- **IP адрес** – IP адрес устройства для подключения клиента;
- **Порт** – сетевой порт устройства для подключения клиента;
- **МЭК61850: Периодичность попыток соединения (мс)** – определяет период попыток подключения к серверу;
- **МЭК61850: Период опроса входов (мс)** – определяет частоту формирования запросов на чтение атрибутов сервера (к таким атрибутам следует провести связь из проекта);
- **МЭК61850: Приоритет процесса** – определяет приоритет потока взаимодействия протокола с runtime (см. рисунок 2.12).

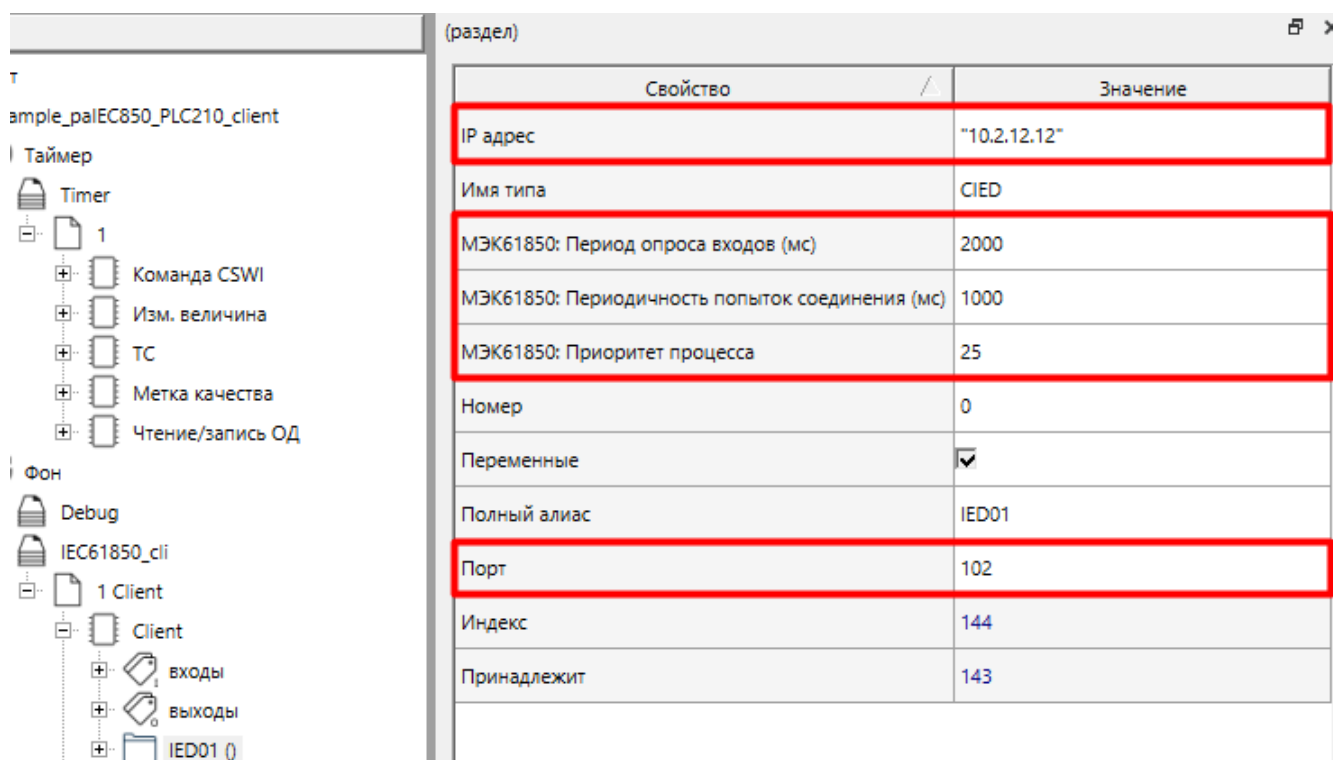


Рисунок 2.13 – Свойства узла IED

Для узла управления набором данных (отчетом) определяются причины передачи отчета.

Активированные в сервере причины передачи будут также активированы по умолчанию у клиента (при необходимости их можно поменять).

С помощью свойства **МЭК61850: включить отчет** можно управлять активацией отчета.

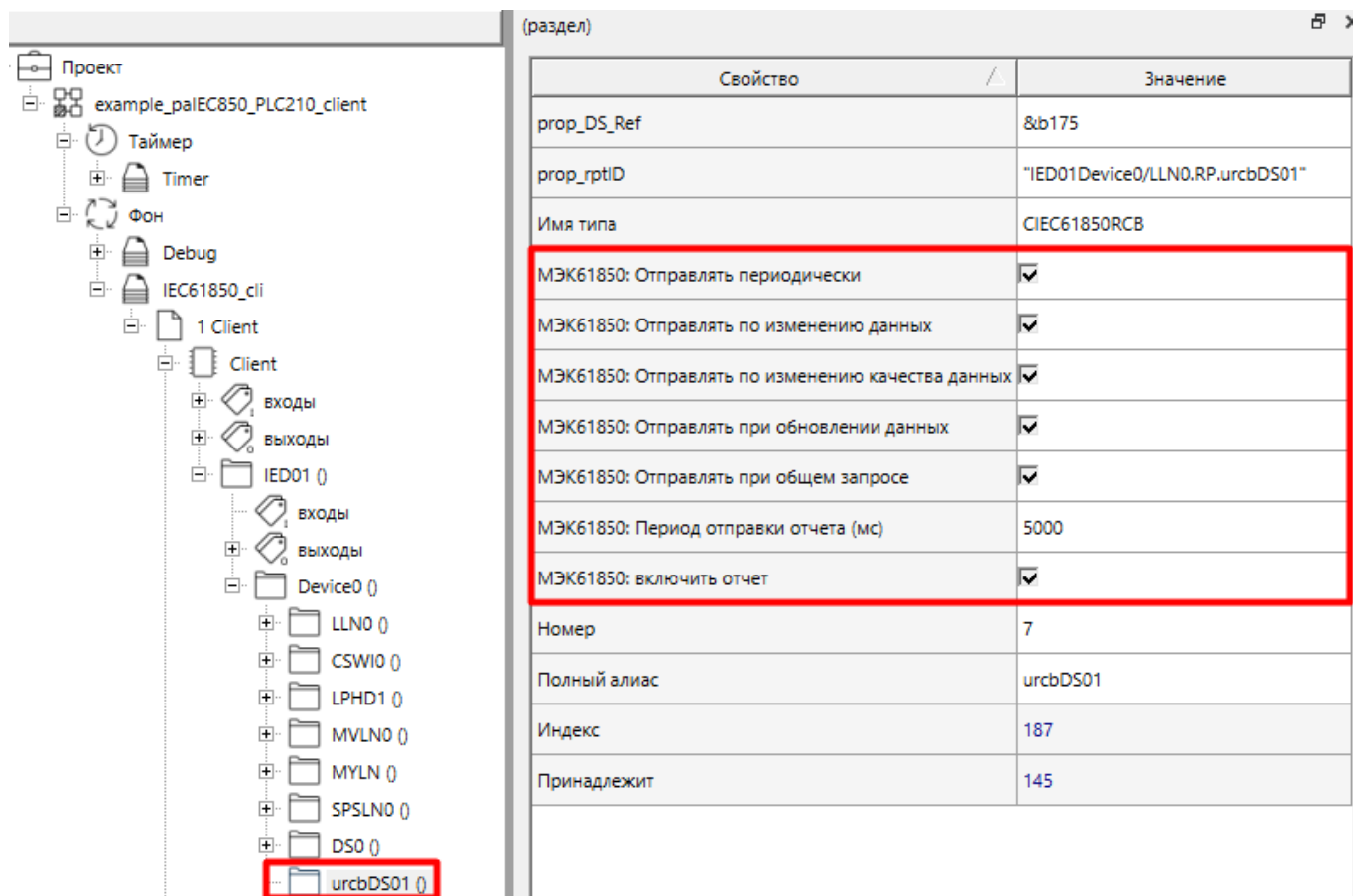


Рисунок 2.14 – Свойства узла управления набором данных (отчетом)

Полный перечень свойств узлов дерева описания устройств *IEC61850cli* приведен в справке среды и в [разделе 2.2.1](#).

Пример работы с блоком приведен в [разделе 4](#).

2.2.1 Свойства узлов дерева описания устройств IEC61850cli

В данном разделе приведены свойства узлов, используемые в дереве описания устройств блока [IEC61850cli](#).

Таблица 2.4 – Описание свойств узлов дерева описания устройств IEC61850cli

Свойство	Идентификатор свойства	Назначение
Класс CIED. Корневой узел дерева описания устройства		
IP адрес	prop_ip	Сетевой адрес устройства 61850
Порт	prop_inter_port	Сетевой порт устройства 61850
МЭК61850: Периодичность попыток соединения (мс)	prop_conn_period	Время ожидания следующей попытки подключения к устройству после неудачной
МЭК61850: Период опроса входов	prop_readin_period	Период отправки запросов чтения к устройству
МЭК61850: Приоритет процесса	prop_thread_priority	Приоритет системного процесса, выполняющего программные функции данного устройства
Пользовательское свойство 00	prop_0	Локальный IP-адрес
Пользовательское свойство 01	prop_1	Локальный TCP-порт
Класс CIEC61850LN. Логический узел		
	prop_object_indx	Индекс относительно узла-владельца

Продолжение таблицы 2.4

Свойство	Идентификатор свойства	Назначение
Класс CIEC61850DO. Объект данных		
	prop_object_indx	Индекс относительно узла-владельца
Класс CIEC61850DA. Атрибут данных		
	prop_object_indx	Индекс относительно узла-владельца
Класс CIEC61850Input. Входы		
	prop_object_indx	Индекс относительно узла-владельца
	prop_bType	Базовый тип данных, определенный протоколом
МЭК61850: Функциональное ограничение	prop_FC	Функциональное ограничение, определенное протоколом
Класс CIEC61850Output. Выходы		
	prop_object_indx	Индекс относительно узла-владельца
	prop_bType	Базовый тип данных, определенный протоколом
МЭК61850: Функциональное ограничение	prop_FC	Функциональное ограничение, определенное протоколом
МЭК61850: Модель управления	prop_ctlModel	Модель управления, определенная протоколом. Только для управляемых атрибутов данных. Возможные значения: <ul style="list-style-type: none"> • status_only; • direct_with_normal_security; • sbo_with_normal_security; • direct_with_enhanced_security; • sbo_with_enhanced_security; • not_controllable
Класс CIEC61850RCB. Блок управления набором данных (отчетом)		
	prop_DS_Ref	Ссылка на блок класса CIEC61850DS, которым управляет данный блок
	prop_rptID	Строка, содержащая полный алиас данного блока (например, "testIEDtestDevice0/LLNRP. urcbMX01")
МЭК61850: Отправлять по изменению данных	prop_trg_data_changed	Условие отправки отчета
МЭК61850: Отправлять при общем опросе	prop_trg_general_inter	Условие отправки отчета
МЭК61850: Отправлять при обновлении данных	prop_trg_general_inter	Условие отправки отчета
МЭК61850: Отправлять по изменению качества данных	prop_trg_qual_changed	Условие отправки отчета
МЭК61850: Отправлять периодически	prop_trg_integrity	Условие отправки отчета

Продолжение таблицы 2.4

Свойство	Идентификатор свойства	Назначение
МЭК61850: Период отправки отчета (мс)	prop_trg_integrity_period_ms	Период отправки отчета в случае установки свойства prop_trg_integrity
МЭК61850: включить отчет	prop_rcb_enabled	Включить отчет

2.2.2 Вспомогательный модуль выполнения команды (IEC61850CmdCSWI)

Блок *IEC61850CmdCSWI* является вспомогательным модулем выполнения команды контроля узла **CSWI**.

Таблица 2.5 – Назначение входов и выходов IEC61850CmdCSWI

Элемент	Описание
Входы	
in	Вход данных
select	Инициировать команду «Выбрать»
execute	Инициировать команду «Выполнить»
cancel	Инициировать команду «Отменить»
Выходы	
bo	Выход для подключения ко входу bo клиента
ref	Назначенный атрибут данных Pos из узла CSWI устройства 61850
err	Результат выполнения команды в виде текстовой строки
ierr	Результат выполнения команды в виде целого числа: 0 – нет ошибки
opSts	Статус выполнения команды: 1 – команда выполнена

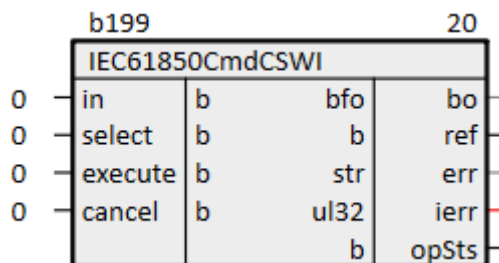


Рисунок 2.15 – Блок IEC61850CmdCSWI

Пример использования блока в проекте приведен в [разделе 4](#).

2.3 Квалификатор (IEC61850BitStrToQuality/IEC61850TransQuality)

Блок *IEC61850BitStrToQuality* является вспомогательным модулем преобразования списка битовых полей в битовую строку **q**.

Названия входов блока соответствуют значениям полей, которые определены в протоколе.

На выход блока **q** назначается атрибут данных из дерева устройства **61850**.

Пример использования блока в проекте приведен в [разделе 3.1](#).

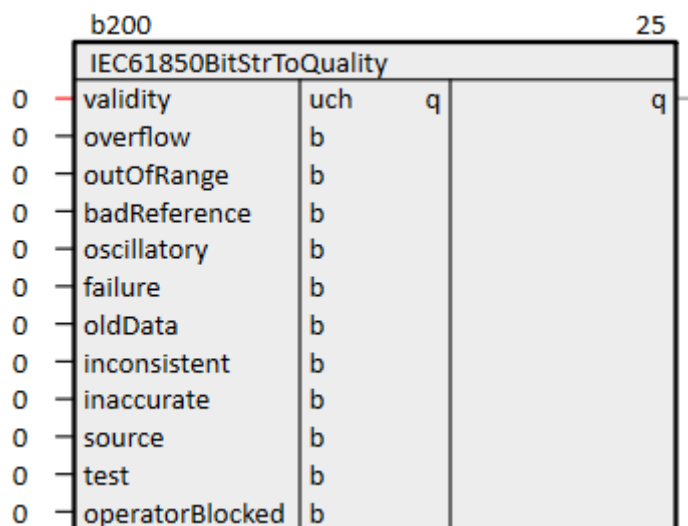


Рисунок 2.16 – Блок IEC61850BitStrToQuality

Блок **IEC61850TransQuality** является вспомогательным модулем преобразования битовой строки **q** в список битовых полей.

На вход блока **q** назначается атрибут данных из дерева устройства **61850**.

Названия выходов блока соответствуют значениям полей, которые определены в протоколе.

Пример использования блока в проекте приведен в [разделе 4](#).

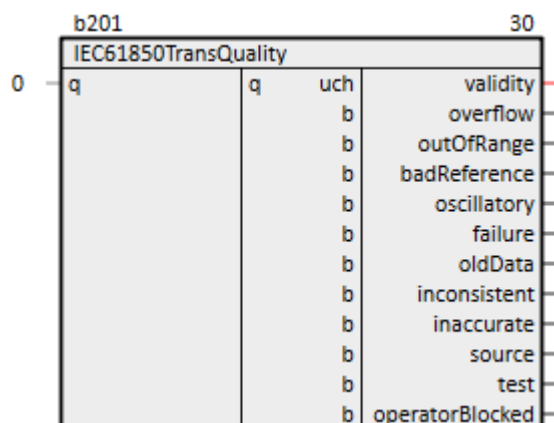


Рисунок 2.17 – Блок IEC61850TransQuality

3 Пример настройки обмена по протоколу МЭК 61850 в режиме сервера

Данный пример доступен для скачивания на [сайте](#). Пароль для доступа к отладчику – 1.

Для настройки обмена по протоколу **МЭК 61850 MMS** в режиме сервера следует:

1. Добавить на любую страницу места работы **Фон** блок **IEC61850srv**.

выбор сетевой платы "/" или "/SOCK2" "/"

	Server		5
	IEC61850srv		
1	enb	b	ul32 sts
100	tmp	ul32	ul32 dsts0
	sdr	str	
0	cfg	ul32	
0	ctl	ul32	

Рисунок 3.1 – Блок IEC61850srv

2. Чтобы протокол функционировал, определить абстрактную модель данных внутри узла блока **IEC61850srv** в дереве проекта. Для этого применяются шаблоны из библиотеки **palIEC850**. Для копирования шаблонов в дерево проекта следует открыть библиотеку **palIEC850** в представлении «дерево».

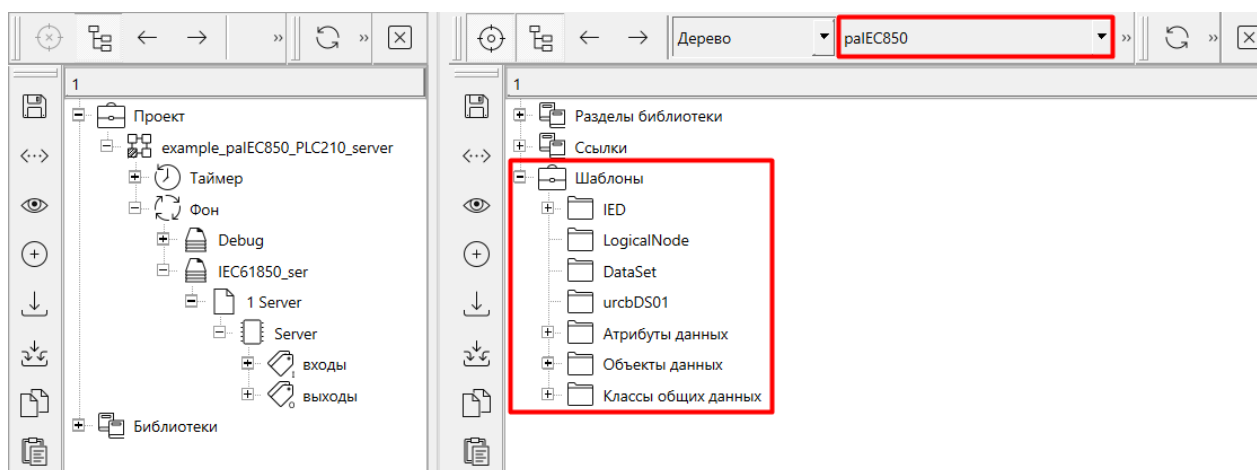


Рисунок 3.2 – Шаблоны библиотеки palIEC850

3. Корневым узлом модели данных является физическое устройство **IED**. Внутри него добавляются логические устройства **LD** и логические узлы **LN**. Логические узлы, в свою очередь, содержат объекты данных с атрибутами. Для копирования необходимого узла из раздела **Шаблоны** библиотеки **palIEC850** следует захватить его мышью, перетащить на нужный узел внутри узла блока **IEC61850srv** в дереве проекта и нажать **Копировать**. Затем нужно скопировать шаблон физического устройства **IED** и присвоить ему имя – **IED01**. Внутри **IED01** находится одно логическое устройство **Device0**. Внутри устройства добавлены обязательные узлы **LLN0** и **LPHD1**.

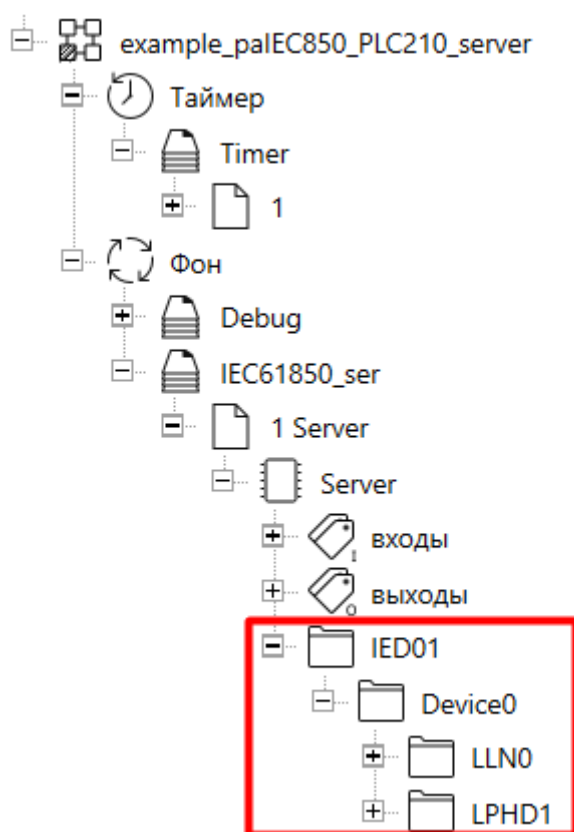


Рисунок 3.3 – Узел IED01 в дереве проекта

3.1 Настройка логических узлов LN

Для настройки логического узла **LN** внутри логического устройства **LD** следует добавить в него шаблон **LogicalNode** из библиотеки *palEC850*.

3.1.1 Настройка контролируемого логического узла класса SPC

Для того, чтобы добавить внутрь логического устройства **LD** контролируемый логический узел, следует:

1. Добавить в узел **Device0** логический узел, копируя шаблон **LogicalNode**, и назначить имя **LN** – **CSWI0**.
2. Скопировать в **CSWI0** контролируемый объект данных класса **SPC**: шаблон **Controllable DataObject (BOOLEAN)** из раздела **Объекты данных**. Присвоить имя добавленному объекту данных – **Pos**.



ПРИМЕЧАНИЕ

Для контролируемых объектов данных атрибуты данных создаются автоматически.

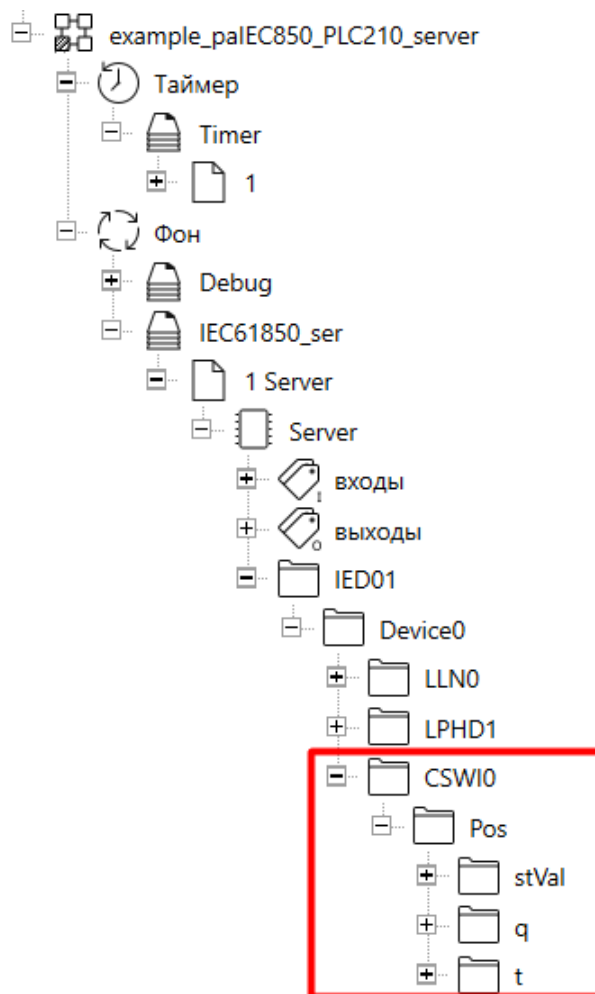


Рисунок 3.4 – Узел CSWI0 в дереве проекта

3. Задать для управляемых атрибутов данных свойство **МЭК61850: модель управления**. Возможные значения см. в справке среды программирования и в [разделе 2.1.1](#).

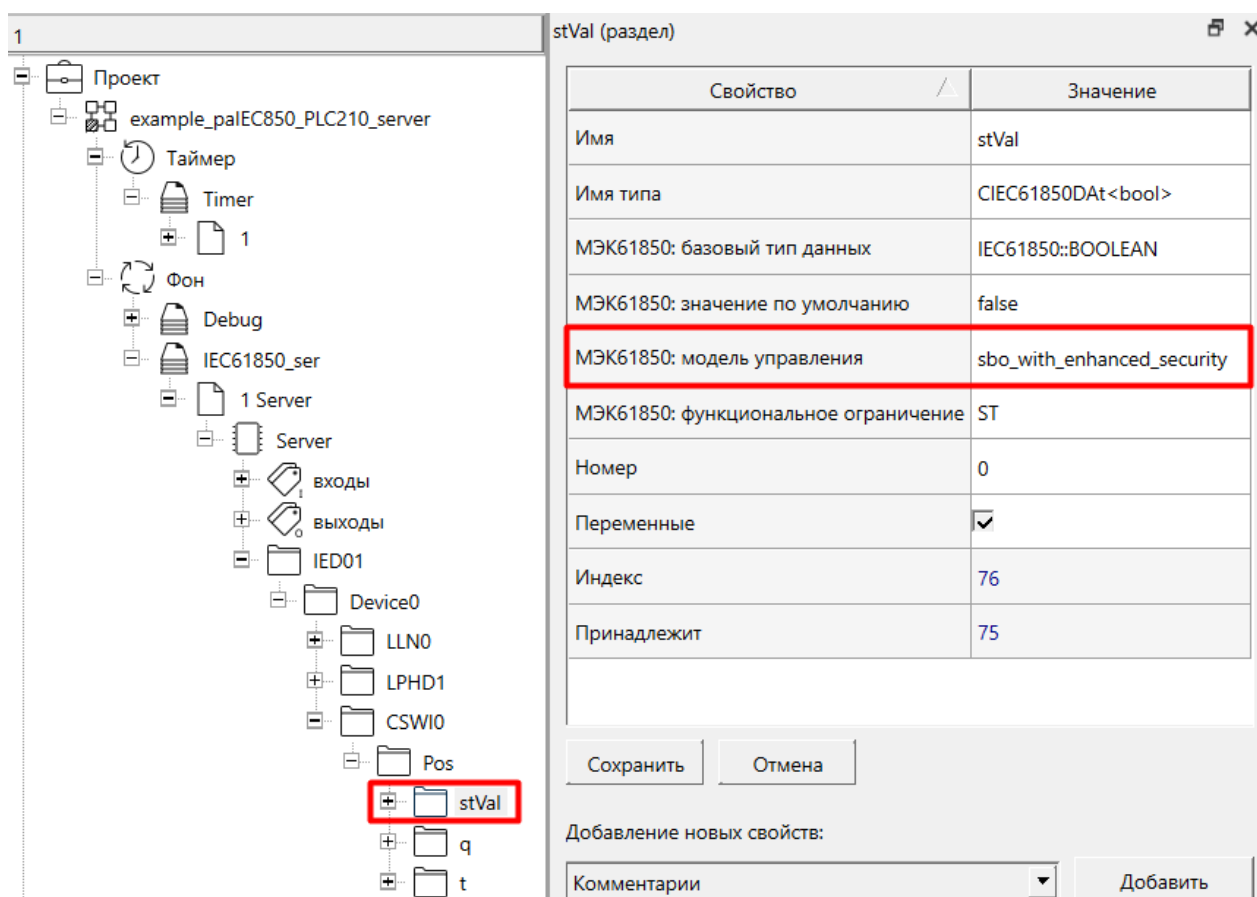


Рисунок 3.5 – Свойства управляемого атрибута данных

4. Подать значение атрибута на вход блока **TransBit** из библиотеки **paCore**, чтобы получить его в программе. Для этого следует захватить вход блока **TransBit** с зажатым **Ctrl**, перетащить на раздел **stVal** и в выпадающем меню выбрать **Добавить**.

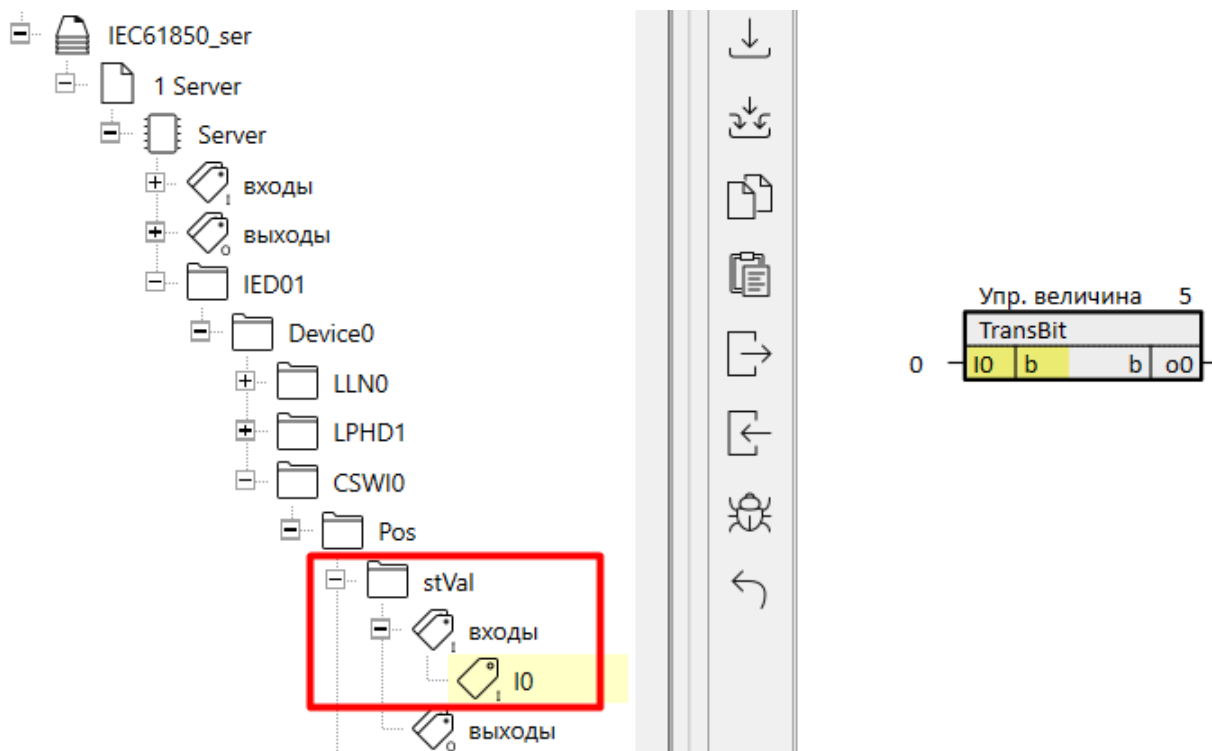


Рисунок 3.6 – Связь управляемого атрибута данных с TransBit

3.1.2 Настройка логического узла класса **MV** – измеряемая величина

Внутри логического устройства можно создать логический узел типа «измеряемая величина **MV**».

Для этого следует:

1. Добавить внутрь **Device0** логический узел, копируя шаблон библиотеки **LogicalNode**, и назначить имя **LN – MVLN0**.
2. Скопировать в **MVLN0** класс данных **MV**: шаблон **MV (Набор 1)** из раздела **Классы общих данных**. Присвоить имя добавленному объекту данных – **MV**.

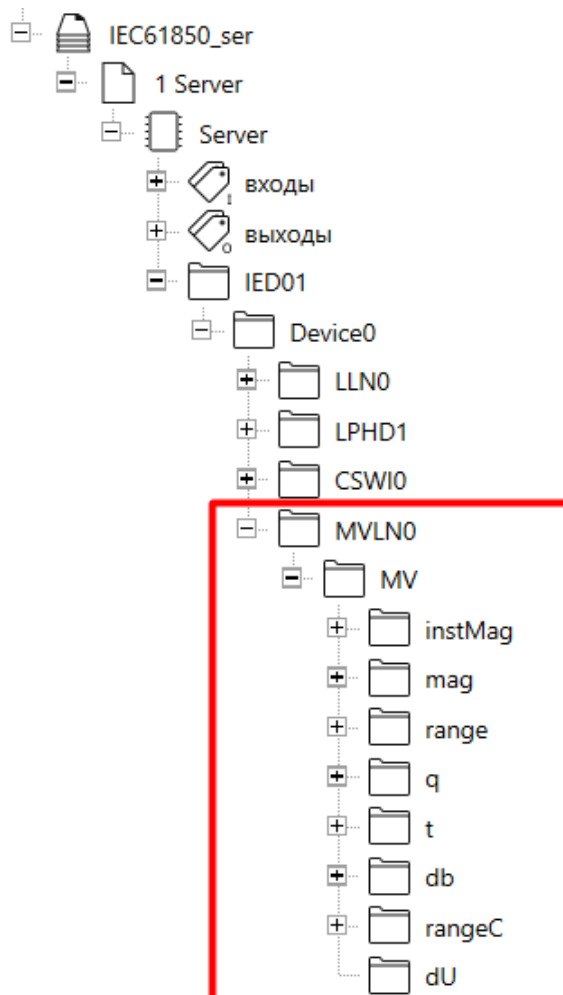


Рисунок 3.7 – Узел **MVLN0** в дереве проекта

3. Для эмуляции сигнала подать синусоиду на атрибут данных **instMag**.

Для этого соединить блоки **GenSign** и **TransFit** из библиотеки **paCore**, как показано на рисунке ниже. Добавить в раздел внутри **instMag** выход блока **TransFit**.

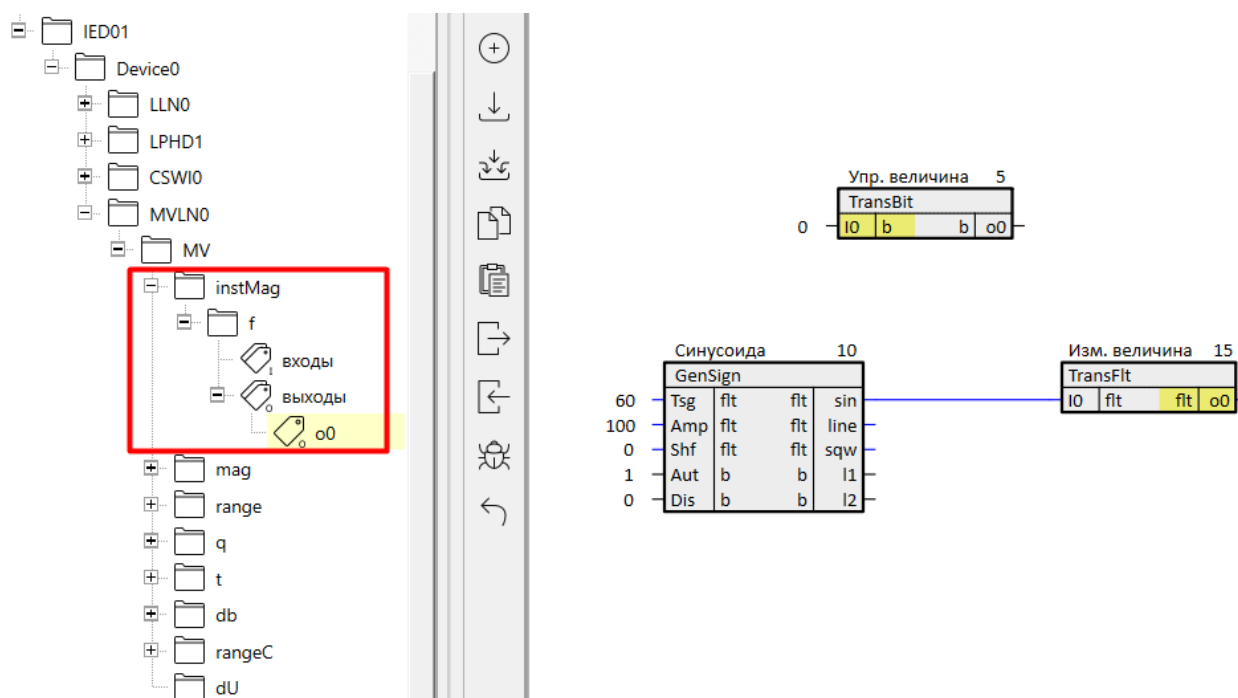


Рисунок 3.8 – Связь атрибута instMag с TransFlt

4. Атрибут **mag** изменяется в соответствии с **instMag** и зоной нечувствительности, которая задается атрибутом **db**.

Зона нечувствительности **db** задается как тысячные процента от диапазона измеряемой величины.

Атрибуты **min** и **max** определяют диапазон измеряемой величины.

Так как амплитуда **GenSign = 100**, следует задать диапазон **min = -100**, **max = 100**.

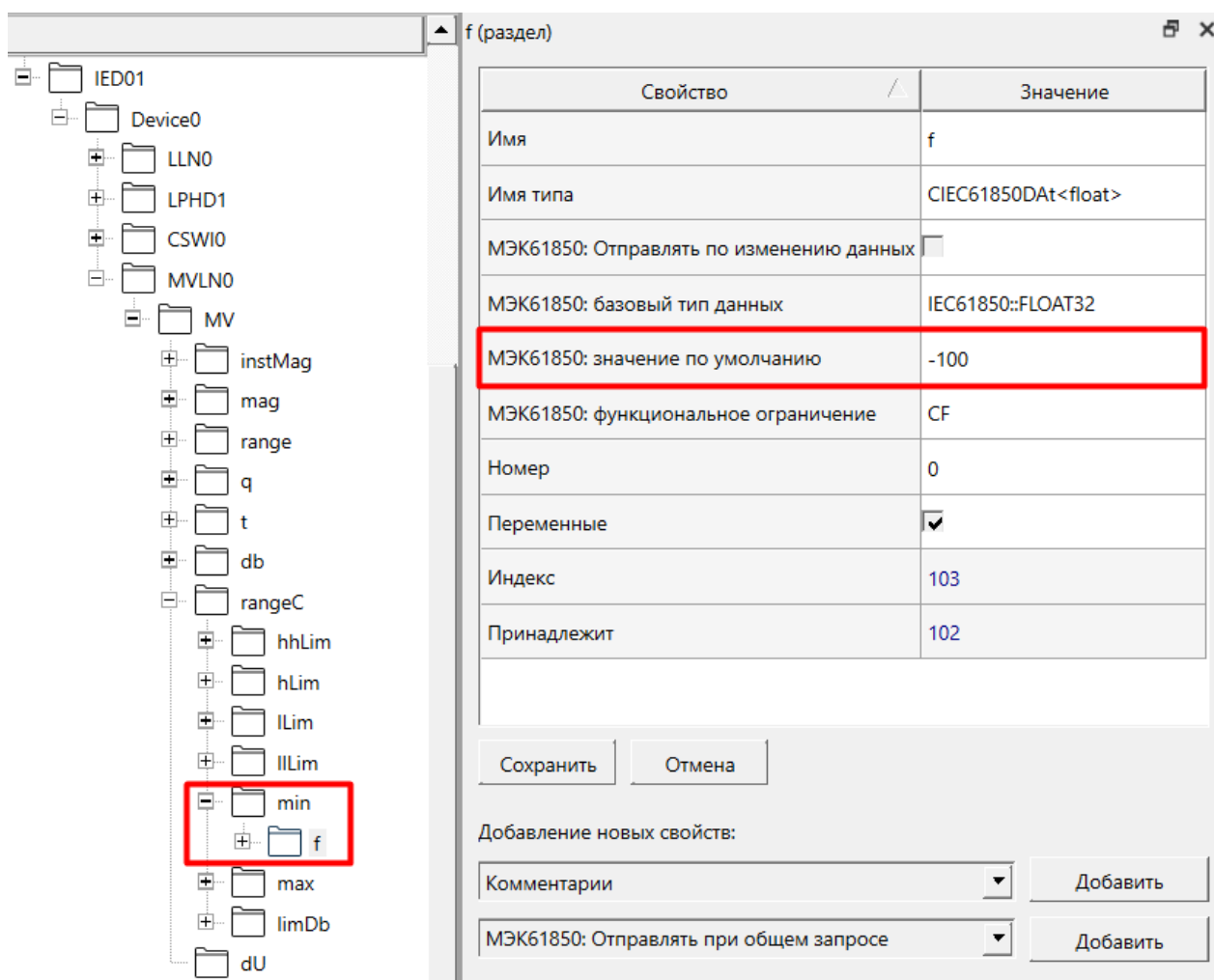


Рисунок 3.9 – Задание диапазона измеряемой величины: атрибут min

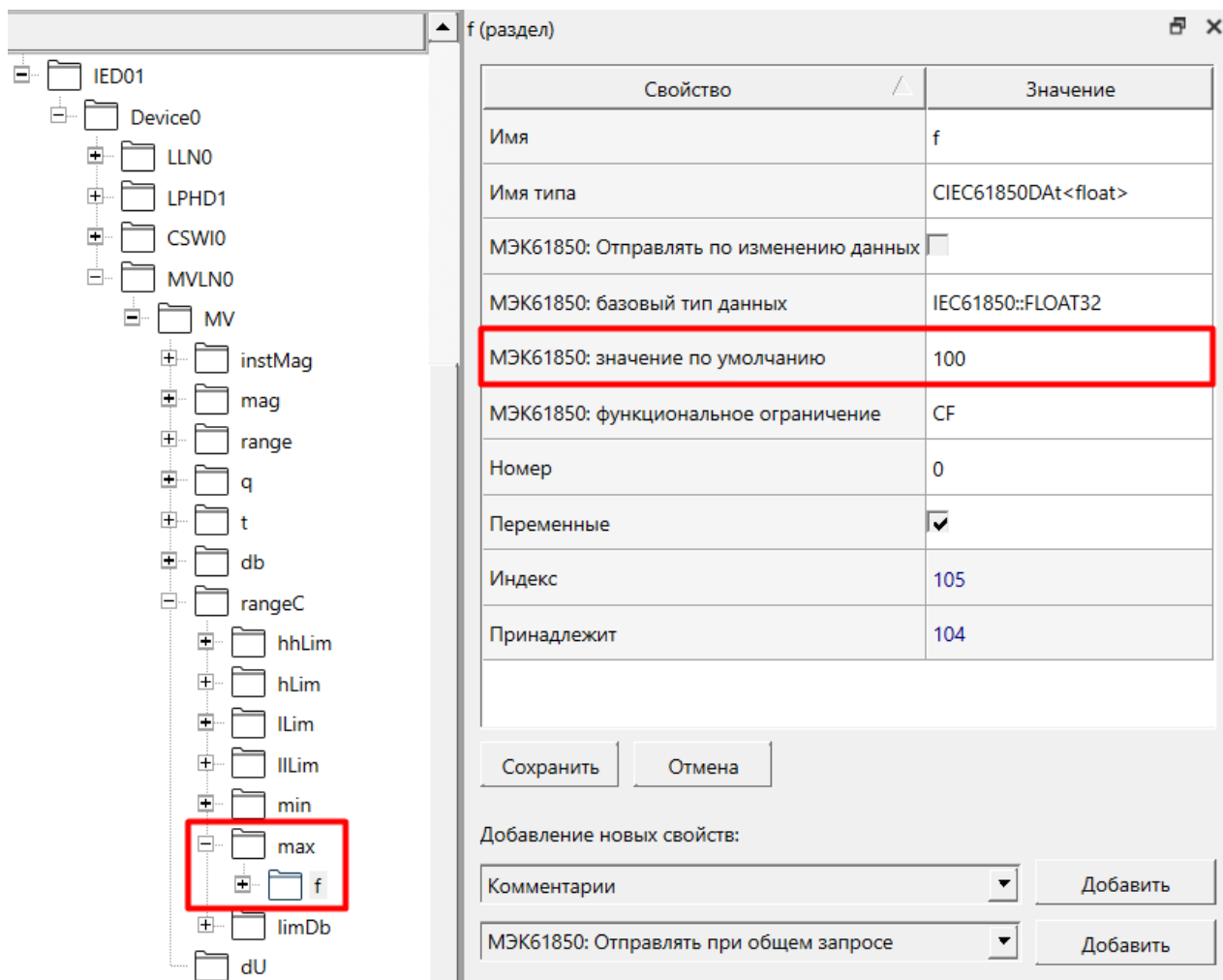


Рисунок 3.10 – Задание диапазона измеряемой величины: атрибут max

5. Задать величину зоны нечувствительности равной **2000** ($\pm 2\%$).

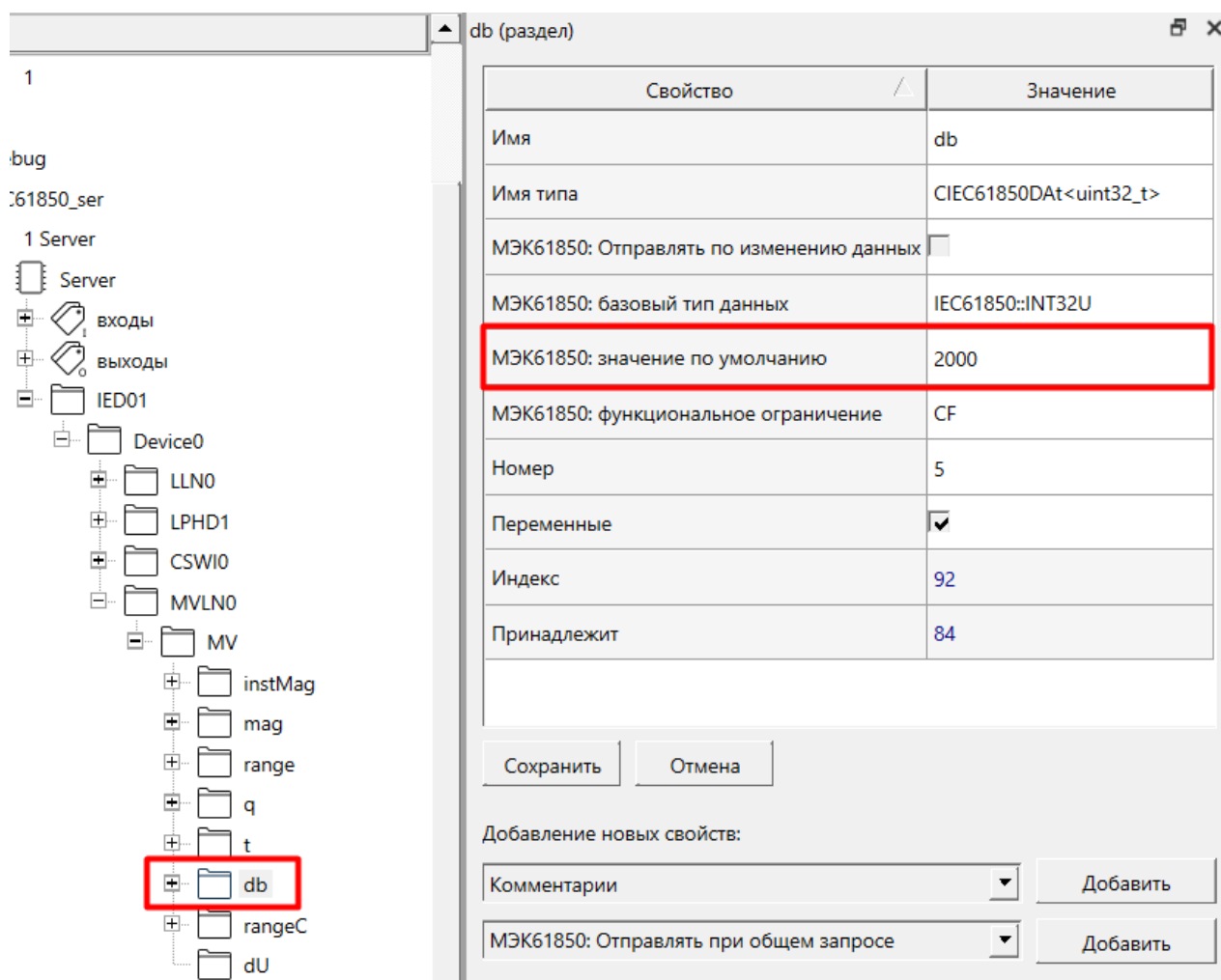


Рисунок 3.11 – Задание зоны нечувствительности: атрибут db

Для передачи измеряемой величины **MV** следует добавить ее в набор данных (отчет) **DataSet**. Пример настройки отчета см. в [разделе 3.2](#).

3.1.3 Настройка логического узла класса SPS – состояние

Для того, чтобы создать логический узел типа «телесигнализация **SPS**» внутри логического устройства, следует:

1. Добавить внутрь **Device0** логический узел, копируя шаблон библиотеки **LogicalNode**. Назначить имя **LN – SPSLN0**.
2. Скопировать в **SPSLN0** класс данных **SPS**: шаблон **SPS** из раздела **Классы общих данных**.

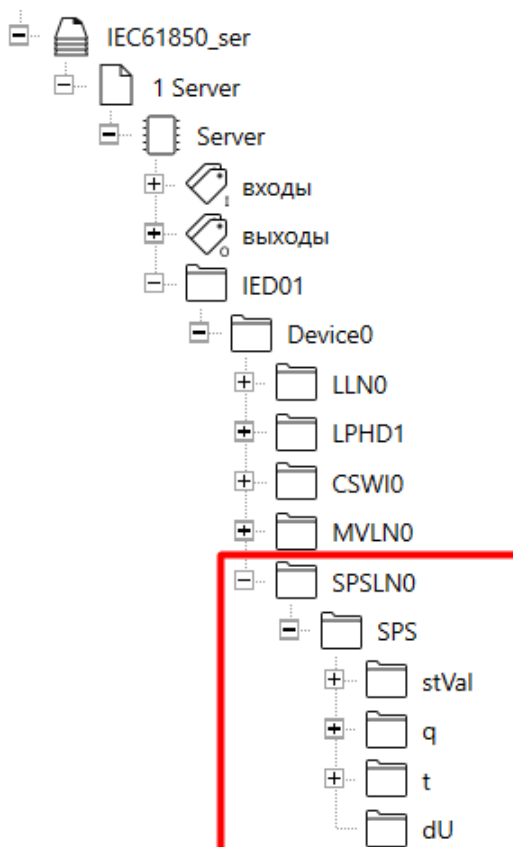


Рисунок 3.12 – Узел SPSLN0 в дереве проекта

- Для задания значения атрибута `stVal` из программы соединить его с выходом блока `TransBit` из библиотеки `paCore`.

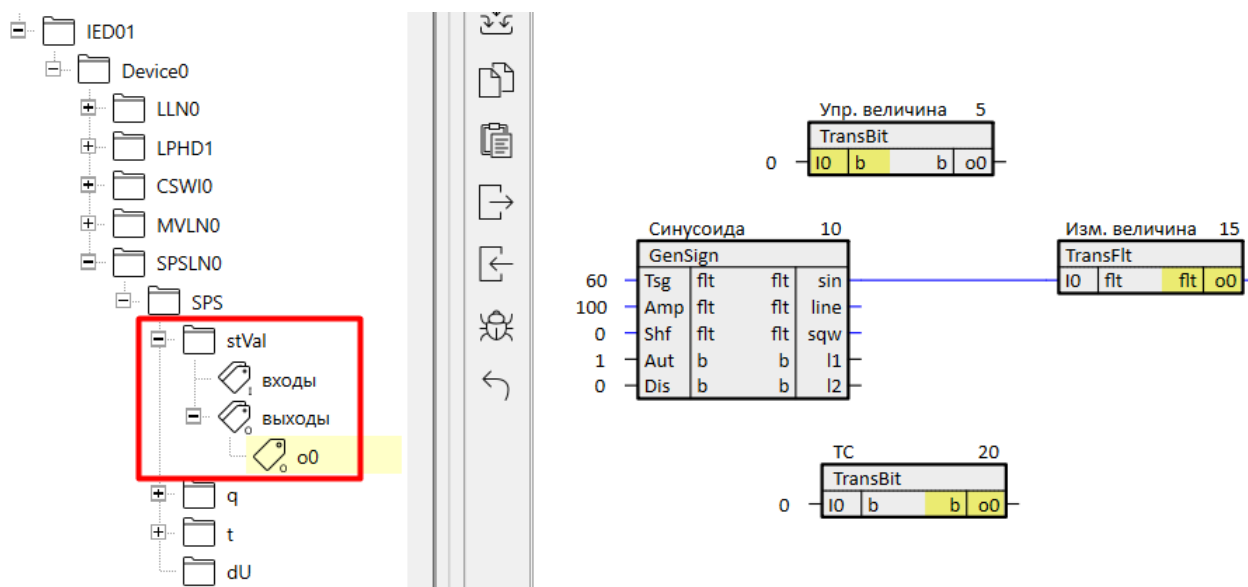


Рисунок 3.13 – Связь атрибута данных класса SPS с TransBit

- Для формирования метки качества использовать вспомогательный блок `IEC61850BitStrToQuality`.

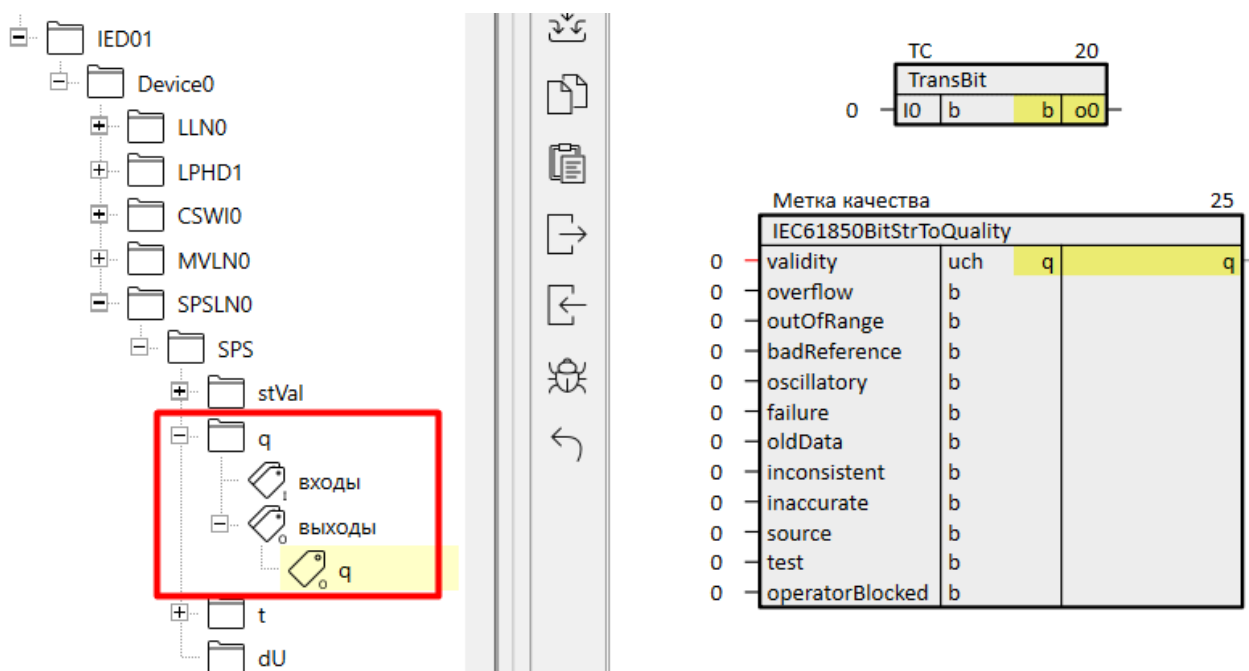


Рисунок 3.14 – Настройка метки качества

- Для передачи состояния **SPS** добавить его в набор данных (отчет) **DataSet**.

Пример настройки отчета см. в [разделе 3.2](#).

3.1.4 Настройка пользовательского логического узла

Для создания пользовательского логического узла внутри логического устройства следует:

- Добавить внутрь **Device0** логический узел, копируя шаблон библиотеки **LogicalNode**. Назначить имя **LN – MYLN**.
- Скопировать в **MYLN** объект данных: шаблон **DataObject** из раздела **Объекты данных**. Назначить имя **DO – MYDO**.
- Скопировать в **MYDO** целочисленный атрибут данных: шаблон **DataAttribute (INT32)** из раздела **Атрибуты данных**. Назначить имя **DA – MYDA**.

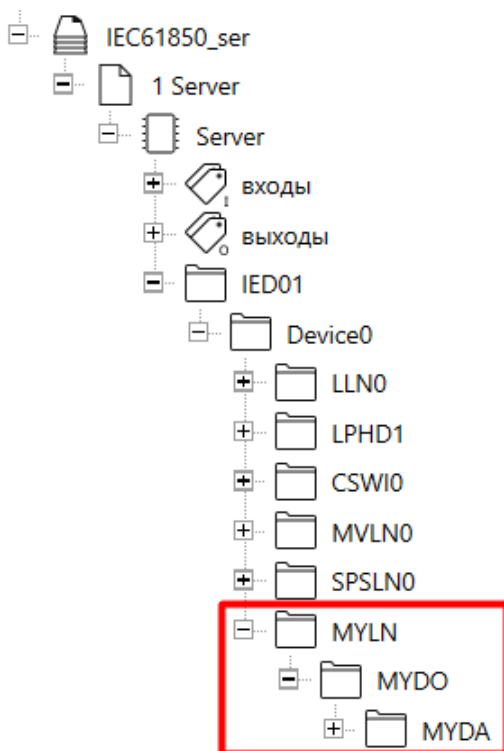


Рисунок 3.15 – Узел MYLN в дереве проекта

- Для созданного объекта данных необходимо применять сервисы протокола чтения и записи. Для этого связать атрибут **MYDA** с входом и выходом блока **TransInt32** из библиотеки **paCore**.



ПРИМЕЧАНИЕ

В данном случае связь с блоком **TransInt32** используется для демонстрации. Такую связь можно организовывать, например, с блоком сохранения уставок **SaverEx** из библиотеки **paCore** (см. документ [Архивирование и сохранение уставок](#)).



Рисунок 3.16 – Связь атрибута данных с TransInt32

- Задать функциональное ограничение для атрибута **MYDA** равным **CF**, чтобы клиент мог и читать, и писать значение.

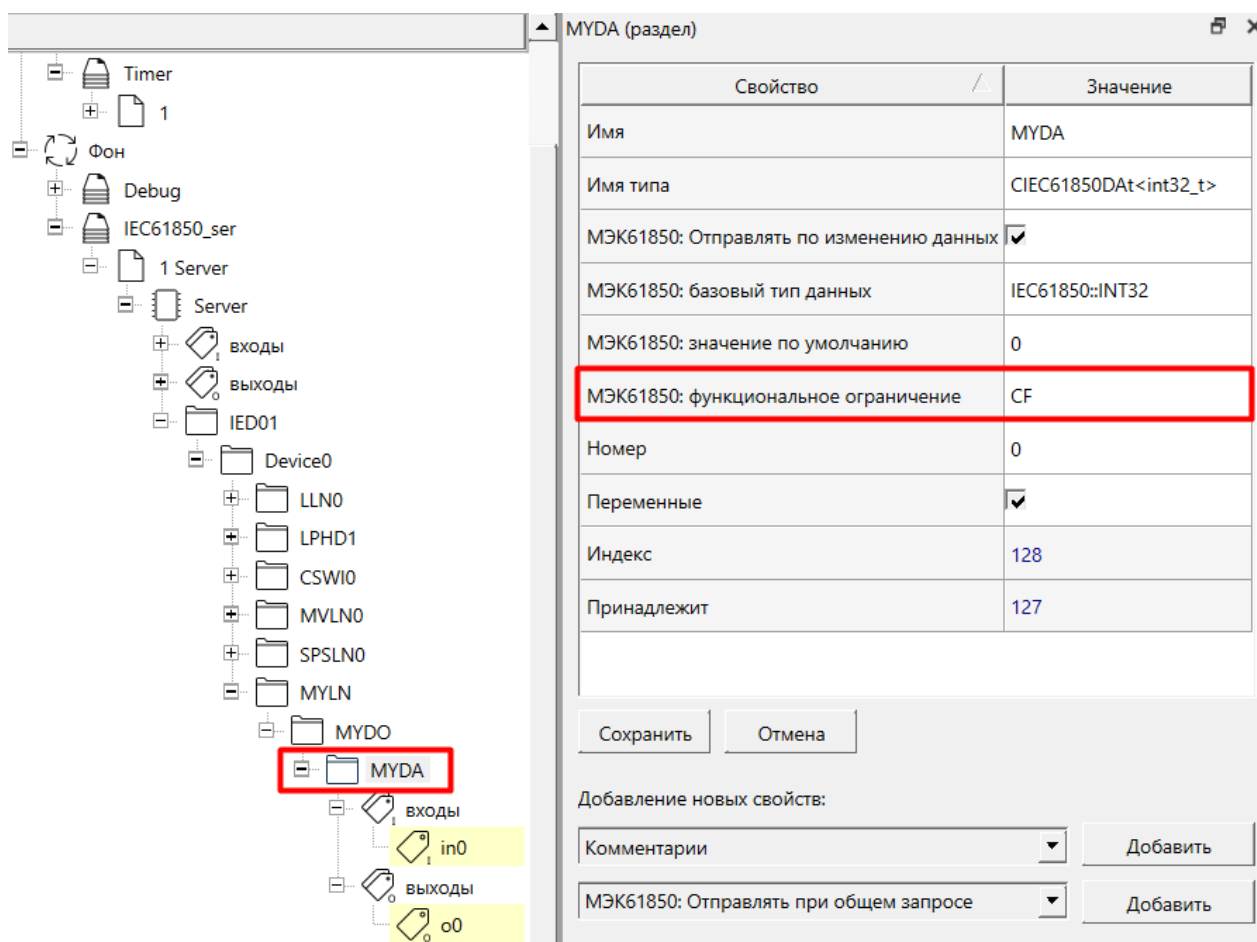


Рисунок 3.17 – Задание функционального ограничения для атрибута MYDA

3.2 Формирование наборов данных DS

Для настройки отправки наборов данных (отчетов) следует:

1. Добавить в узел логического устройства шаблоны **DataSet** и **ucrbDS01** из библиотеки **palEC850**.
2. Добавить в узел **Device0** шаблон **DataSet**. Назначить имя нового набора данных – **DS0**.
3. Для управления набором данных добавить в узел логического устройства **Device0** блок управления отчетом **ucrbDS01**. В свойстве блока управления **МЭК61850: название набора данных** прописать имя созданного набора данных – **DS0**.
4. Добавить причины передачи отчета: по изменению данных и при общем запросе.

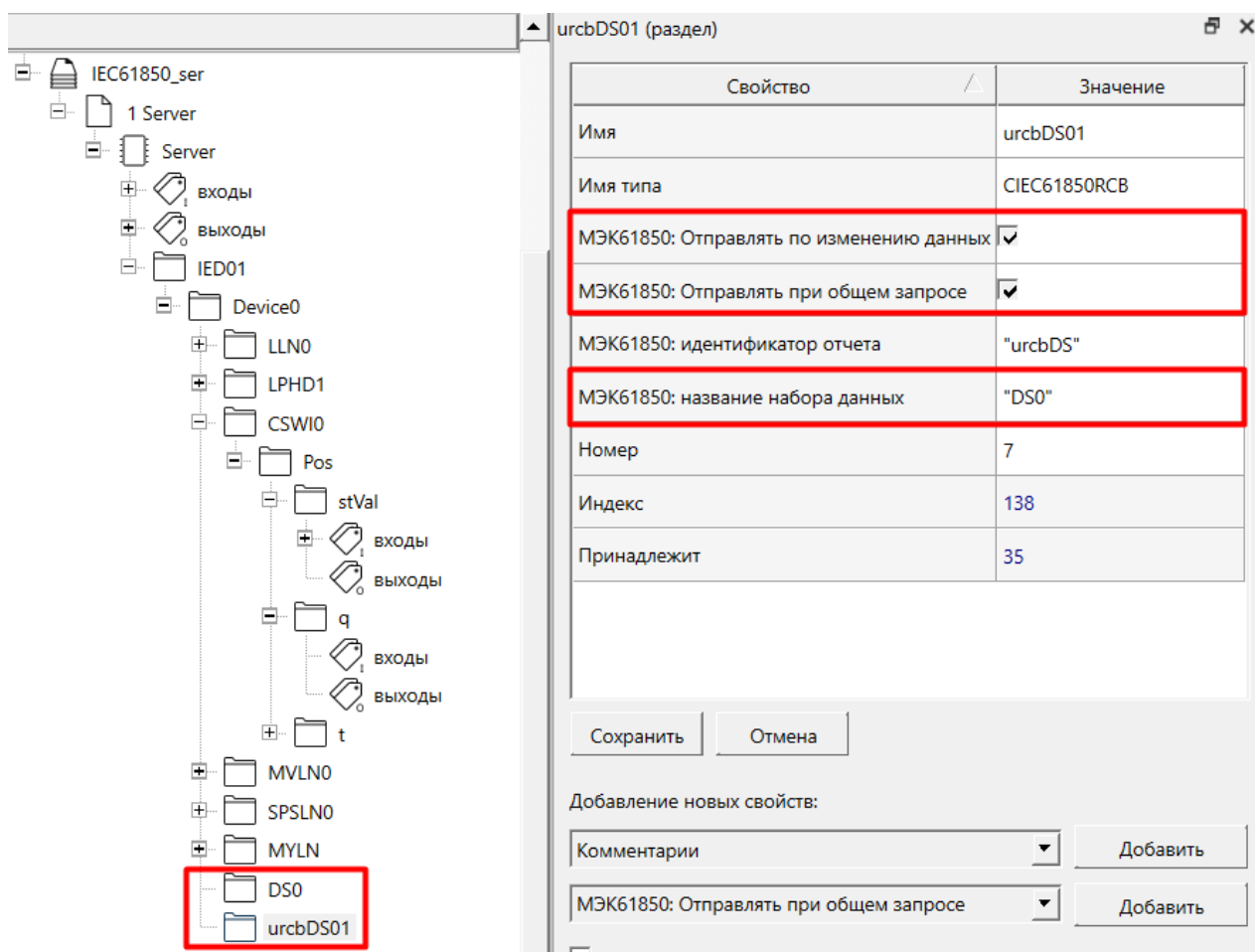


Рисунок 3.18 – Добавление набора данных и блока управления набором данных

5. Добавить в раздел набора данных данные, которые необходимо передавать в отчете.
Для этого следует перетащить необходимый объект данных в узел набора данных и в выпадающем меню выбрать **Добавить**.
6. Добавить в отчет **DSO** созданные в [разделе 3.1](#) объекты данных **MV** и **SPS**.

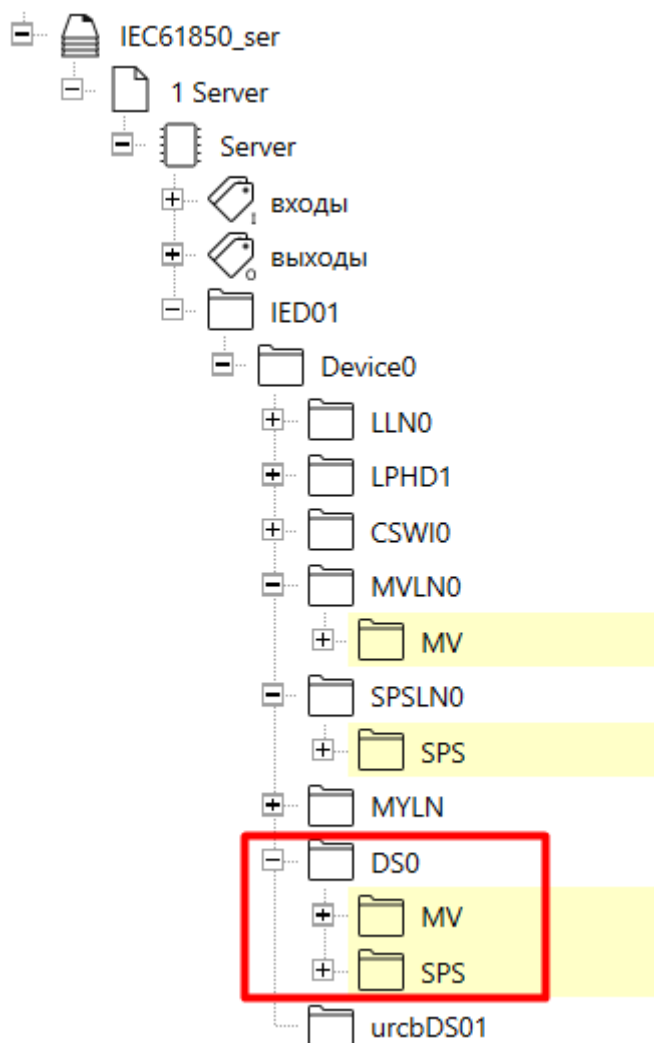


Рисунок 3.19 – Добавление объектов данных в отчет

3.3 Обмен данными с клиентом IEDScout

Здесь в качестве клиента для [настроенного сервера](#) используется программа **IEDScout**.

Для обмена данными с клиентом **IEDScout** следует:

1. Запустить программу с настроенным сервером на **ПЛК210**.
2. Запустить программу **IEDScout** и выбрать команду поиска **IED – Discover IED**.

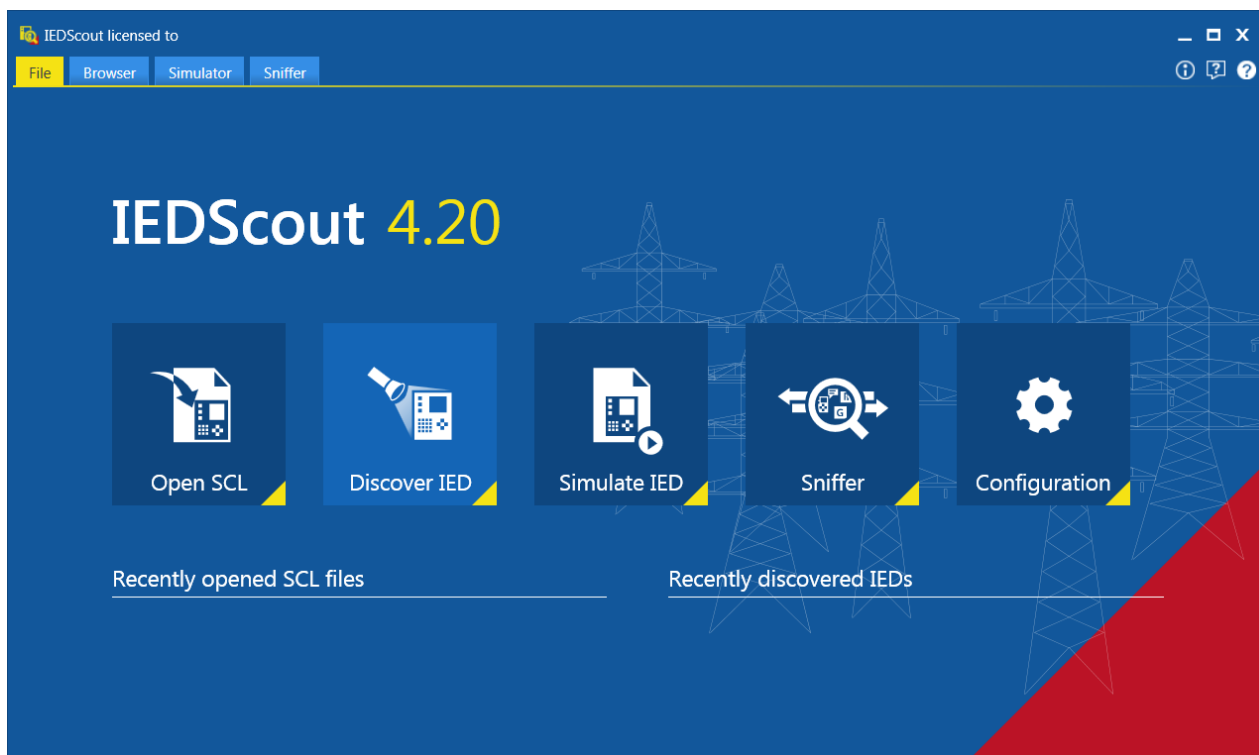


Рисунок 3.20 – Запуск клиента IEDScout

3. В открывшемся окне ввести IP-адрес контроллера и нажать **Discover**.

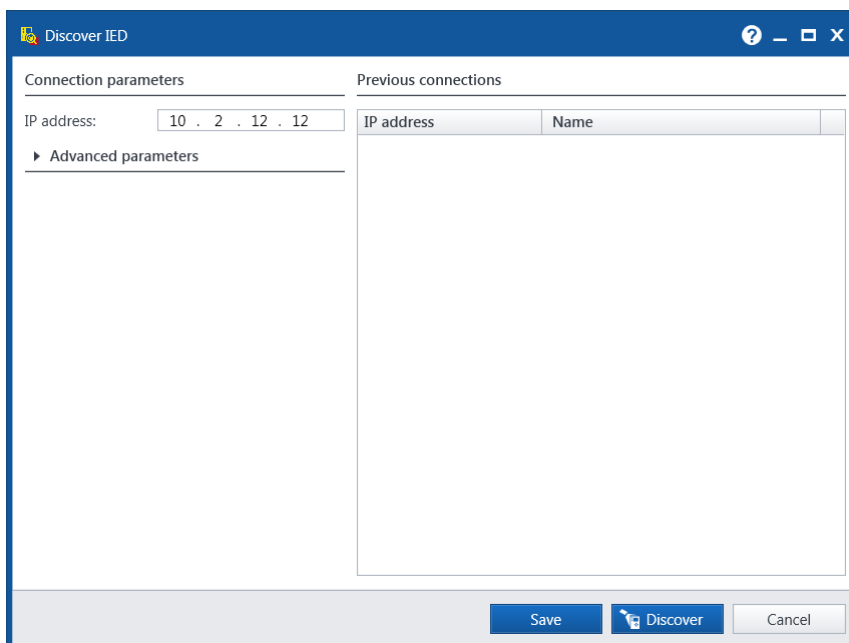


Рисунок 3.21 – Поиск устройства 61850

4. Модель данных сервера отобразится в дереве программы.

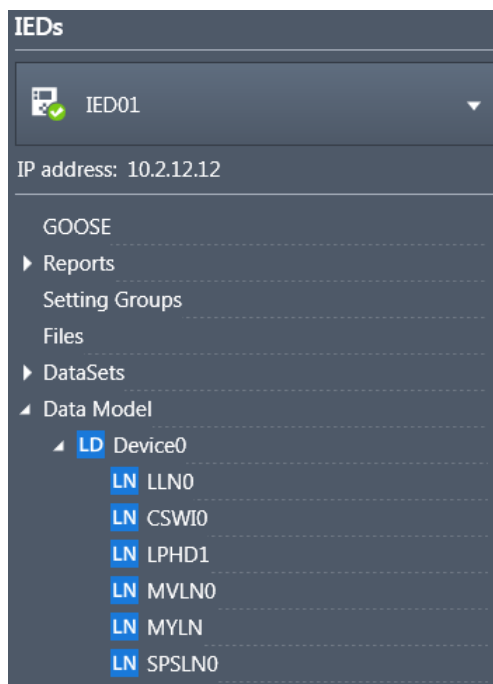


Рисунок 3.22 – Модель данных устройства 61850

5. Открыть настроенный отчет **DS0** в узле дерева программы **Reports**.

IED01 • Reports • Device0 • LLN0 • urcbDS01

R urcbDS01

Control Block attributes

Enabled	false
Reserved	false
Control Block reference	IED01Device0/LLN0\$RP\$urcbDS01
Report ID	urcbDS
DataSet reference	IED01Device0/LLN0\$DS0
Trigger options	DataChange, GeneralInterrogation
Buffer time (ms)	0
Configuration revision	1
Integrity period (ms)	5000
Owner	

Information received in last Report

Data

Name	Value
DO MVLN0.MV	
DA instMag [MX]	
DA mag [MX]	
DA range [MX]	
DA q [MX]	
DA t [MX]	
DO SPSLN0.SPS	
DA stVal [ST]	
DA q [ST]	
DA t [ST]	

Рисунок 3.23 – Отчет DS0

Запуск отчета осуществляется через кнопку **Enable**.

6. В открывшемся окне можно увидеть, что в клиенте автоматически установились те условия передачи, которые были заданы в блоке управления набором данных сервера – по изменению и при общем запросе.

Запустить отчет, нажав кнопку **Enable**.

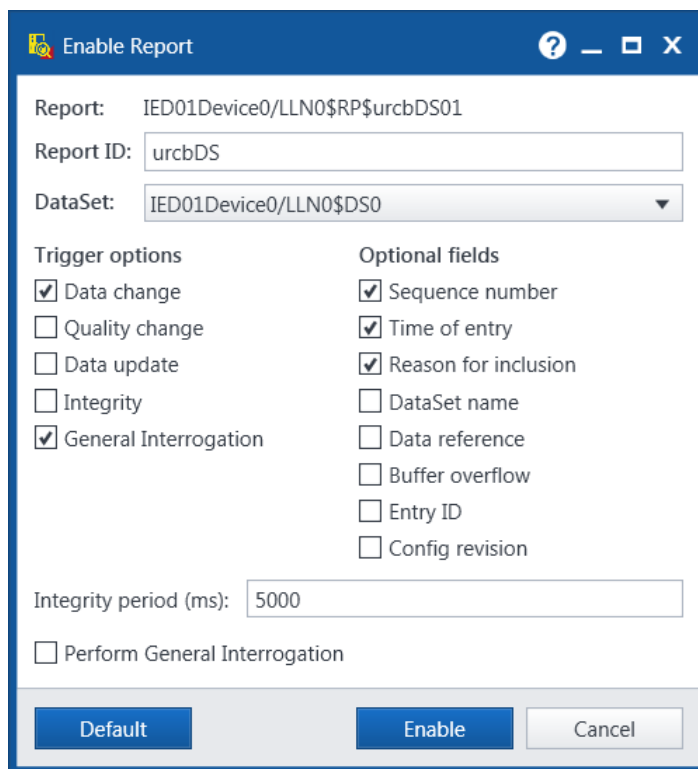


Рисунок 3.24 – Окно запуска отчета

Успешный обмен показан на рисунках ниже.

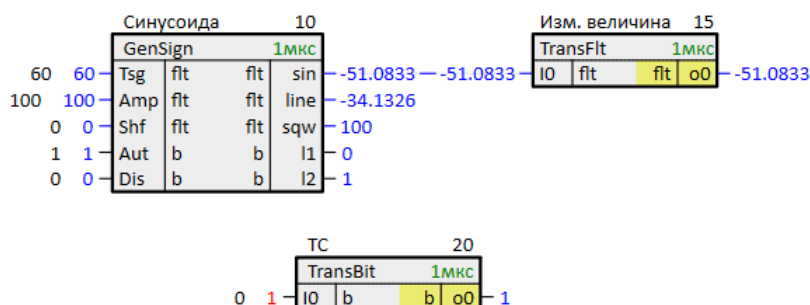


Рисунок 3.25 – Успешный обмен данными. Сервер

Data		Name	Value
▾	DO	MVLN0.MV	92,667
▸	DA	inst... [MX]	92,667
▸	DA	mag [MX]	92,667
	DA	range [MX]	normal
▸	DA	q [MX]	good
▸	DA	t [MX]	24.04.2024 19:07:29...
▾	DO	SPSLN0.SPS	true
	DA	stVal [ST]	true
▸	DA	q [ST]	good
▸	DA	t [ST]	24.04.2024 19:05:00...

Рисунок 3.26 – Успешный обмен данными. Клиент

При изменении метки качества данные не передадутся, так как не установлено условие передачи по изменению метки качества.

Для передачи метки качества следует:

1. Отключить отчет, установить в окне **Enable Report** условие передачи по изменению метки качества – **Quality change**, запустить отчет.

Метка качества		25	
IEC61850BitStrToQuality		1мкс	
0	2	validity	uch q
0	1	overflow	b
0	0	outOfRange	b
0	0	badReference	b
0	0	oscillatory	b
0	0	failure	b
0	1	oldData	b
0	0	inconsistent	b
0	0	inaccurate	b
0	0	source	b
0	0	test	b
0	0	operatorBlocked	b

Рисунок 3.27 – Передача метки качества. Сервер

DA q	[ST]	invalid	⚠
Validity		invalid	
Quality Details			⚠
Overflow		true	
OutOfRange		false	
BadReference		false	
Oscillatory		false	
Failure		false	
OldData		true	
Inconsistent		false	
Inaccurate		false	
Source		process	
Test		false	
OperatorBlocked		false	

Рисунок 3.28 – Передача метки качества. Клиент

- Для передачи ТУ открыть узел **CSWI0** в дереве **Data Model**.

The screenshot shows the configuration interface for IED01. On the left, the 'Data Model' tree is expanded to show the 'CSWI0' node. On the right, the details for the 'CSWI0' node are displayed, showing a table with the following data:

IED01 • Data Model • Device0 • CSWI0	
LN CSWI0	Switch controller
Name	Value
DO Pos	false

Рисунок 3.29 – Узел CSWI0

- Для записи значения выделить объект данных **Pos** и нажать кнопку **Control**.

В открывшемся окне следует ввести новое значение **Pos**. Так как выбрана модель управления с подтверждением перед записью, то следует сначала нажать **Select**, затем **Operate**.

Рисунок 3.30 – Отправка команды

4. Значение передается на сервер.

LN CSWI0 Switch controller		
Name		Value
DO	Pos	true
DA	stVal [ST]	true
DA	q [ST]	good
DA	t [ST]	24.04.2024 18:44:17.193
DA	Oper [CO]	
DA	Cancel [CO]	
DA	SBO [CO]	
DA	SBOw [CO]	
DA	ctlModel [CF]	sbo-with-enhanced-security

Рисунок 3.31 – Передача телеуправления. Клиент

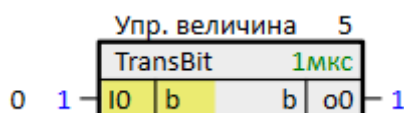


Рисунок 3.32 – Передача телеуправления. Сервер

Для логического узла **MYLN** можно использовать только сервисы чтения и записи. Узел **MYLN** находится в дереве **Data Model**.

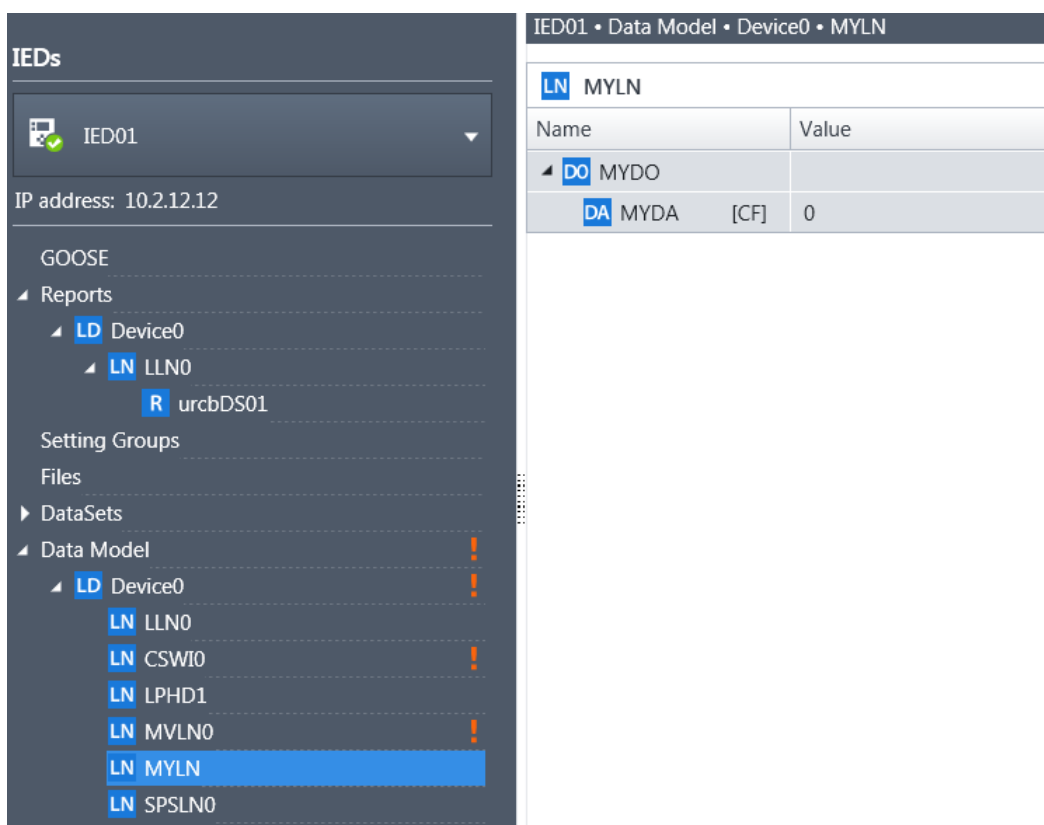


Рисунок 3.33 – Узел MYLN

Для записи значения следует выделить атрибут данных **MYDA** и нажать кнопку **Write**.
В открывшемся окне ввести новое значение и нажать **Write**.

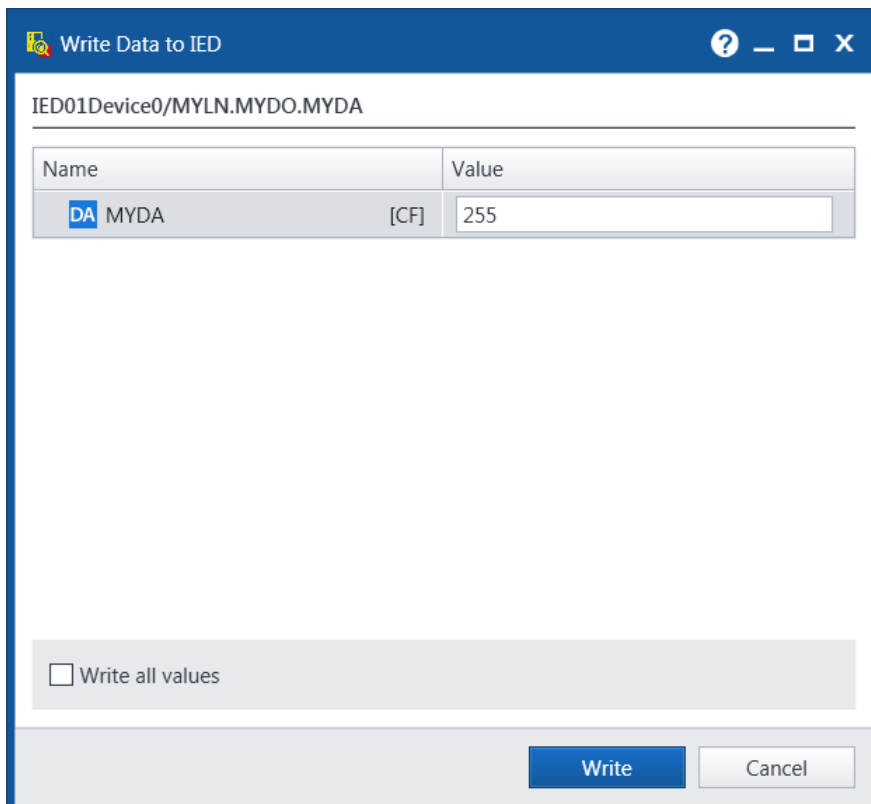


Рисунок 3.34 – Окно записи MYDA

Введенное значение будет записано.

LN MYLN		
Name	Value	
DO MYDO		
DA MYDA	[CF]	255

Рисунок 3.35 – Запись MYDA

Для того, чтобы прочитать значение, нужно выделить атрибут данных **MYDA** и нажать кнопку **Read**. Значение **MYDA** будет прочитано.

LN MYLN		
Name	Value	
DO MYDO		
DA MYDA	[CF]	255

Рисунок 3.36 – Чтение MYDA

Чтение/запись ОД 35
 TransInt32 1мкс
 0 255 — in0 i32 i32 o0 — 255

Рисунок 3.37 – Успешная запись MYDA в сервере

Программа **IEDScout** позволяет экспортировать конфигурацию сервера в файл с расширением **.cid** для последующего импортирования в программе клиента (см. [раздел 4](#)).

Для экспорта следует нажать кнопку **Save SCL**. При сохранении следует указать тип файла ***.cid**.

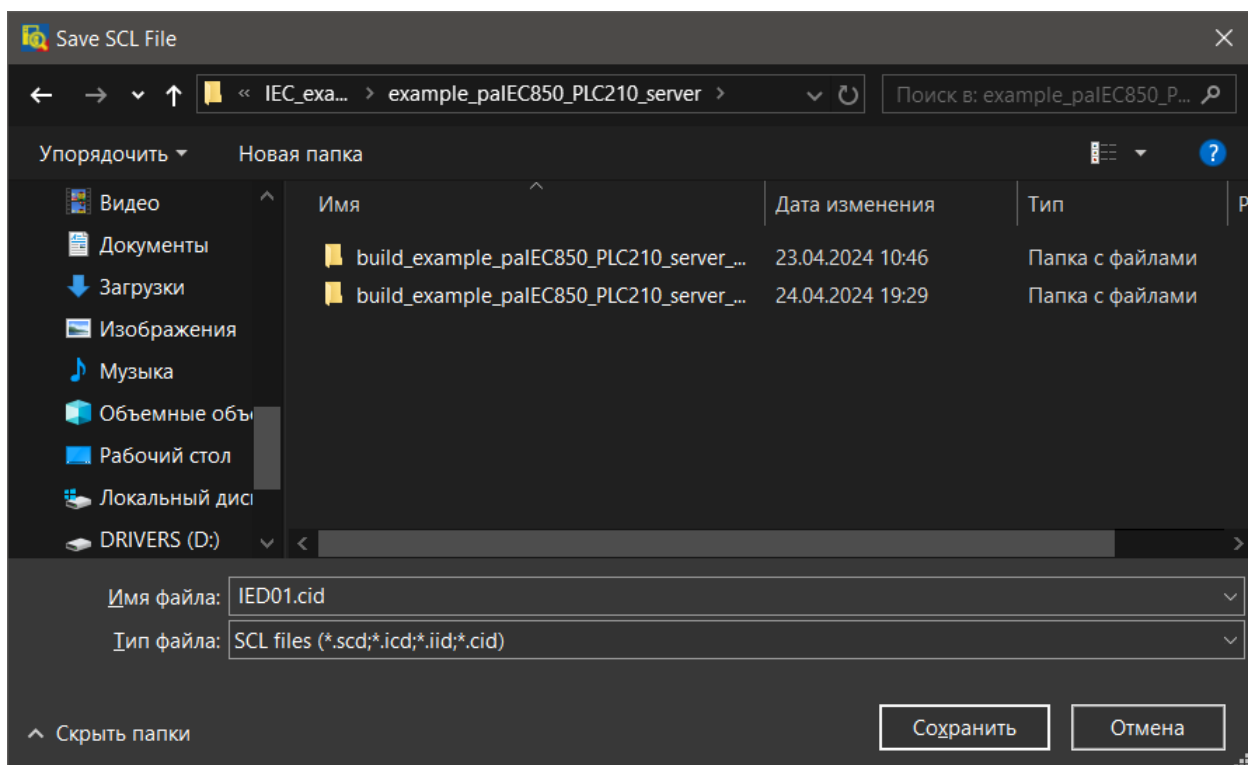


Рисунок 3.38 – Сохранение конфигурации сервера

3.4 Обмен данными с клиентом Multi-Protocol MasterOPC Server

Здесь в качестве клиента для настроенного сервера используется программа Multi-ProtocolMasterOPC Server.

Для обмена данными с Multi-ProtocolMasterOPC Server следует:

1. Запустить программу с настроенным сервером на ПЛК210.
2. Запустить Multi-ProtocolMasterOPC Server, добавить протокол **IEC61850CLIENT**, добавить в протокол устройство **IEC61850CLIENT**.
3. Задать IP-адрес и порт сервера в настройках узла **iec61850client**, включить использование атрибутов **q** и **t**.

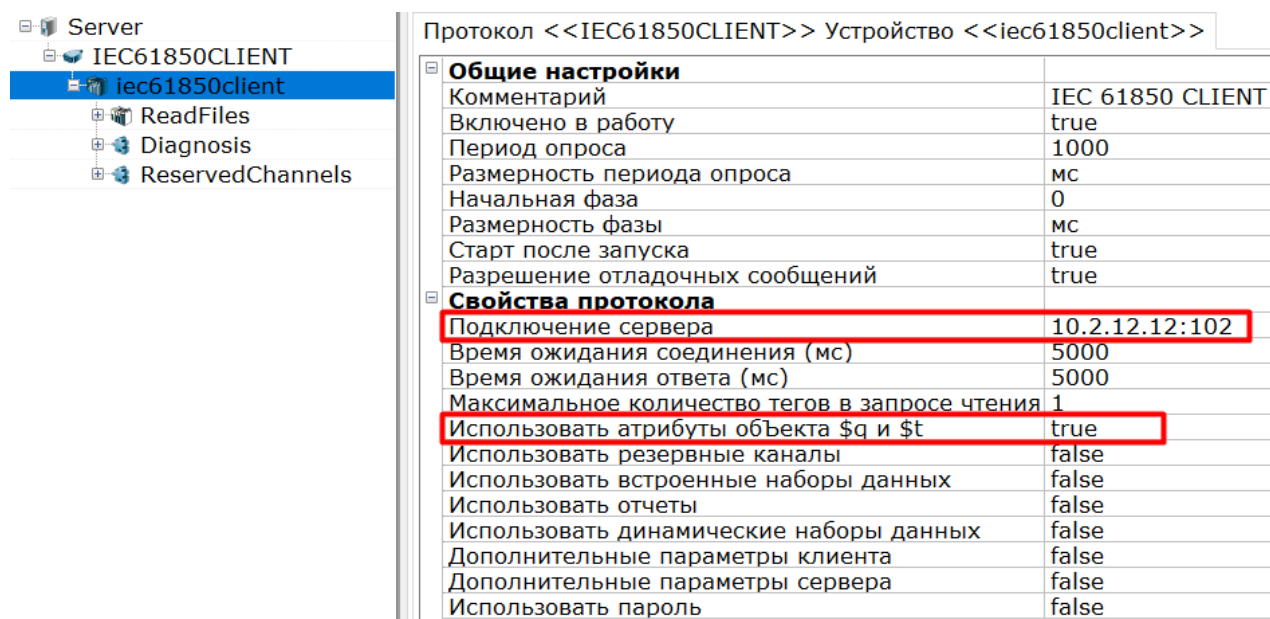


Рисунок 3.39 – Настройка iec61850client

4. В узле **iec61850client** выбрать команду **Добавить – Теги протокола (импорт)**.
5. В новом окне установить соединение с сервером и выбрать необходимые для импорта узлы.

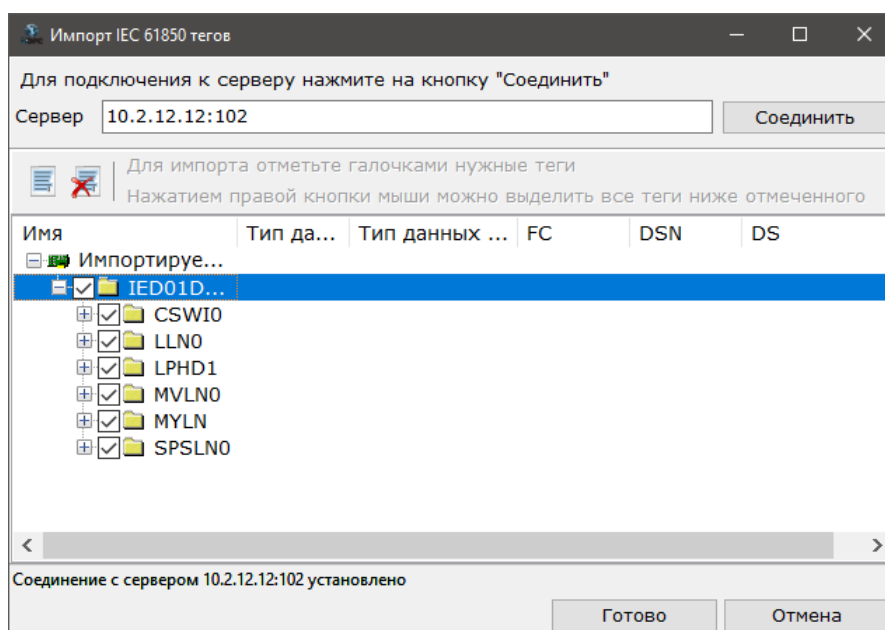


Рисунок 3.40 – Импорт тегов сервера

6. Нажать **Готово**. Модель данных сервера отобразится в дереве программы.

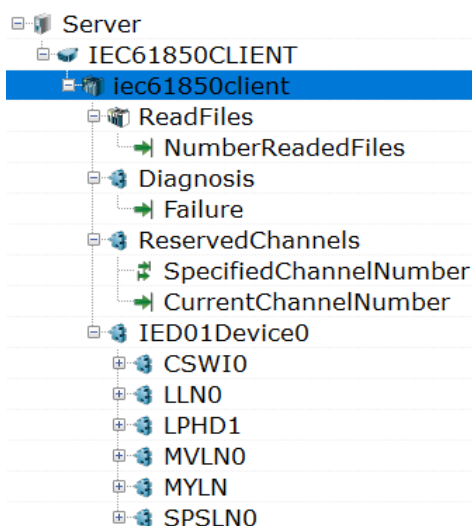


Рисунок 3.41 – Модель данных устройства 61850

7. Для того, чтобы посылать команду ТУ, задать для атрибута **CSWI\$Pos\$SBOw\$ctIVal** тип доступа **только запись**.

Устройство <<iec61850client>> Тер <<Pos\$SBOw\$ctIVal>>	
Общие настройки	
Включен в работу	true
Комментарий	
Тип данных в сервере	bool
Тип доступа	WriteOnly
HDA	
HDA доступ	false
Дополнительные настройки	
Чтение сразу после записи	false
Свойства протокола	
Функциональная принадлежность	CO
Тип данных MMS	boolean
Идентификатор	IED01Device0/CSWIO.Pos.SBOw.ctIVal
Имя набора данных	
Имя контрольного блока отчета	
Идентификатор признака качества	
Идентификатор метки времени	

Рисунок 3.42 – Настройка атрибута ТУ

8. Для того, чтобы читать и записывать атрибут **MYDA**, задать для него тип доступа **чтение/запись**.

Устройство <<iec61850client>> Тер <<MYDO\$MYDA>>	
Общие настройки	
Включен в работу	true
Комментарий	
Тип данных в сервере	int32
Тип доступа	ReadWrite
Сервис	
Пересчет (A*X + B)	false
Ограничение минимального значения	false
Ограничение максимального значения	false
HDA	
HDA доступ	false
Дополнительные настройки	
Чтение сразу после записи	false
Свойства протокола	
Функциональная принадлежность	CF
Тип данных MMS	integer
Идентификатор	IED01Device0/MYLN.MYDO.MYDA
Имя набора данных	
Имя контрольного блока отчета	

Рисунок 3.43 – Настройка атрибута MYDA

9. Запустить OPC-сервер.

10. Корректная передача измеряемой величины **MV** и состояния **SPS** показана на рисунках ниже.

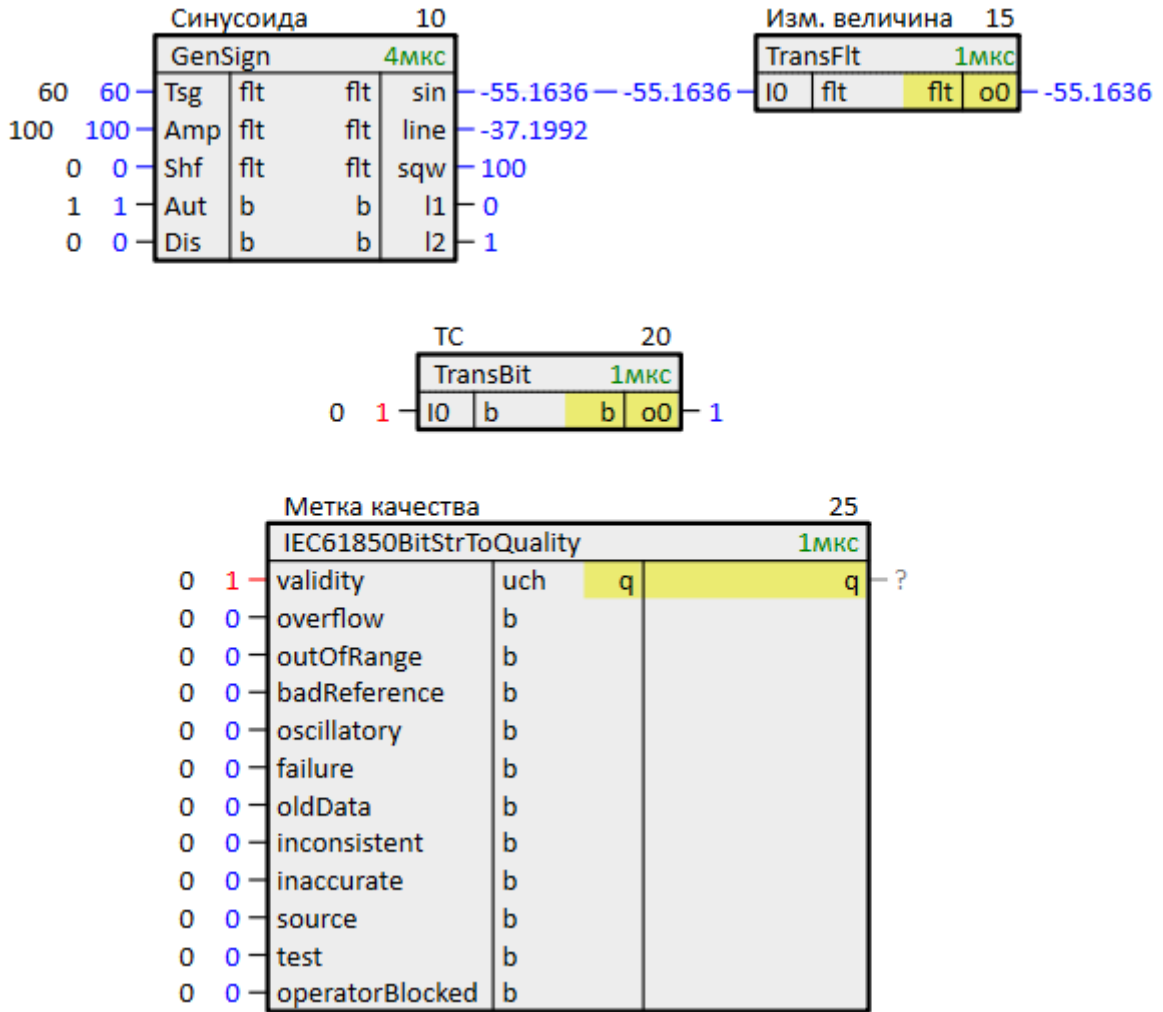


Рисунок 3.44 – Успешный обмен. Сервер

Group - MVLN0 : MVLN0

Идентификатор	Регион	Адрес в...	Значение	Качество
IEC61850CLIENT.iec...	IED01Device0/MVLN0.MV.db		2000	GOOD
IEC61850CLIENT.iec...	IED01Device0/MVLN0.MV.rangeC.hhLim.f		0	GOOD
IEC61850CLIENT.iec...	IED01Device0/MVLN0.MV.rangeC.hLim.f		0	GOOD
IEC61850CLIENT.iec...	IED01Device0/MVLN0.MV.rangeC.limDb		0	GOOD
IEC61850CLIENT.iec...	IED01Device0/MVLN0.MV.rangeC.lLim.f		0	GOOD
IEC61850CLIENT.iec...	IED01Device0/MVLN0.MV.rangeC.lLim.f		0	GOOD
IEC61850CLIENT.iec...	IED01Device0/MVLN0.MV.rangeC.max.f		100	GOOD
IEC61850CLIENT.iec...	IED01Device0/MVLN0.MV.rangeC.min.f		-100	GOOD
IEC61850CLIENT.iec...	IED01Device0/MVLN0.MV.dU			GOOD
IEC61850CLIENT.iec...	IED01Device0/MVLN0.MV.instMag.f		-63.74142	GOOD
IEC61850CLIENT.iec...	IED01Device0/MVLN0.MV.mag.f		-65.18240	GOOD
IEC61850CLIENT.iec...	IED01Device0/MVLN0.MV.q		000000000000	GOOD
IEC61850CLIENT.iec...	IED01Device0/MVLN0.MV.range		0	GOOD
IEC61850CLIENT.iec...	IED01Device0/MVLN0.MV.t		2024-04-25 08:00:47.839	GOOD

Рисунок 3.45 – Успешный обмен (MV). Клиент

Идентификатор	Регион	Адрес в...	Значение	Качество
IEC61850CLIENT.iec...	IED01Device0/SPSLN0.SPS.dU			GOOD
IEC61850CLIENT.iec...	IED01Device0/SPSLN0.SPS.q		100000000000	GOOD
IEC61850CLIENT.iec...	IED01Device0/SPSLN0.SPS.stVal		true	SUB_NORMAL
IEC61850CLIENT.iec...	IED01Device0/SPSLN0.SPS.t		2024-04-25 07:59:44.989	SUB_NORMAL

Рисунок 3.46 – Успешный обмен (SPS). Клиент

11. Чтобы передать команду управления, следует записать **true** в атрибут **CSWI\$Pos\$SBOw\$ctiVal**.

IEC61850CLIENT.iec...	IED01Device0/CSWI0.Pos.SBOW.ctlNum	0	GOOD
IEC61850CLIENT.iec...	IED01Device0/CSWI0.Pos.SBOW.ctlVal	true	GOOD
IEC61850CLIENT.iec...	IED01Device0/CSWI0.Pos.SBOW.origin.orCat	0	GOOD
IEC61850CLIENT.iec...	IED01Device0/CSWI0.Pos.SBOW.origin.orIdent	0	GOOD
IEC61850CLIENT.iec...	IED01Device0/CSWI0.Pos.SBOW.T	1970-01-01 03:00:00.000	GOOD
IEC61850CLIENT.iec...	IED01Device0/CSWI0.Pos.SBOW.Test	false	GOOD
IEC61850CLIENT.iec...	IED01Device0/CSWI0.Pos.q	00000000000000	GOOD
IEC61850CLIENT.iec...	IED01Device0/CSWI0.Pos.stVal	true	GOOD
IEC61850CLIENT.iec...	IED01Device0/CSWI0.Pos.t	2024-04-25 08:04:58.145	GOOD

Рисунок 3.47 – Команда ТУ. Клиент

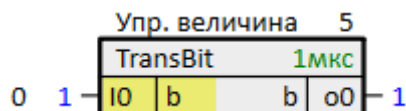


Рисунок 3.48 – Команда ТУ. Сервер

12. Задать новое значение для атрибута MYDA.

Идентификатор	Регион	Адрес в...	Значение
IEC61850CLIENT.iec6...	IED01Device0/MYLN.MYDO.MYDA		585

Рисунок 3.49 – Чтение/запись MYDA. Клиент

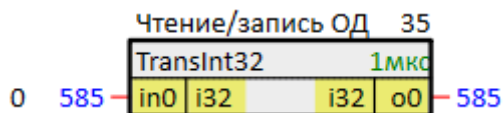


Рисунок 3.50 – Чтение/запись MYDA. Сервер

4 Пример настройки обмена по протоколу МЭК 61850 в режиме клиента

Данный пример доступен для скачивания на [сайте](#). Пароль для доступа к отладчику – 1.

Для настройки обмена по протоколу **МЭК 61850 MMS** в режиме **клиента** следует:

1. Добавить на любую страницу места работы **Фон** блок **IEC61850cli**.

выбор сетевой платы "/" или "/SOCK2" "/"

		Client		5
		IEC61850cli		
1	enb	b	ul32	sts
100	tmp	ul32	ul32	dsts0
	sdr	str		
0	cfg	ul32		
0	ctl	ul32		

Рисунок 4.1 – Блок IEC61850cli

2. Для функционирования протокола импортировать абстрактную модель данных сервера в узел блока **IEC61850cli** в дереве проекта.

Для этого нужно нажать правой кнопкой мыши на экземпляр блока **IEC61850cli** в дереве проекта и в выпадающем меню выбрать команду **Импорт**.

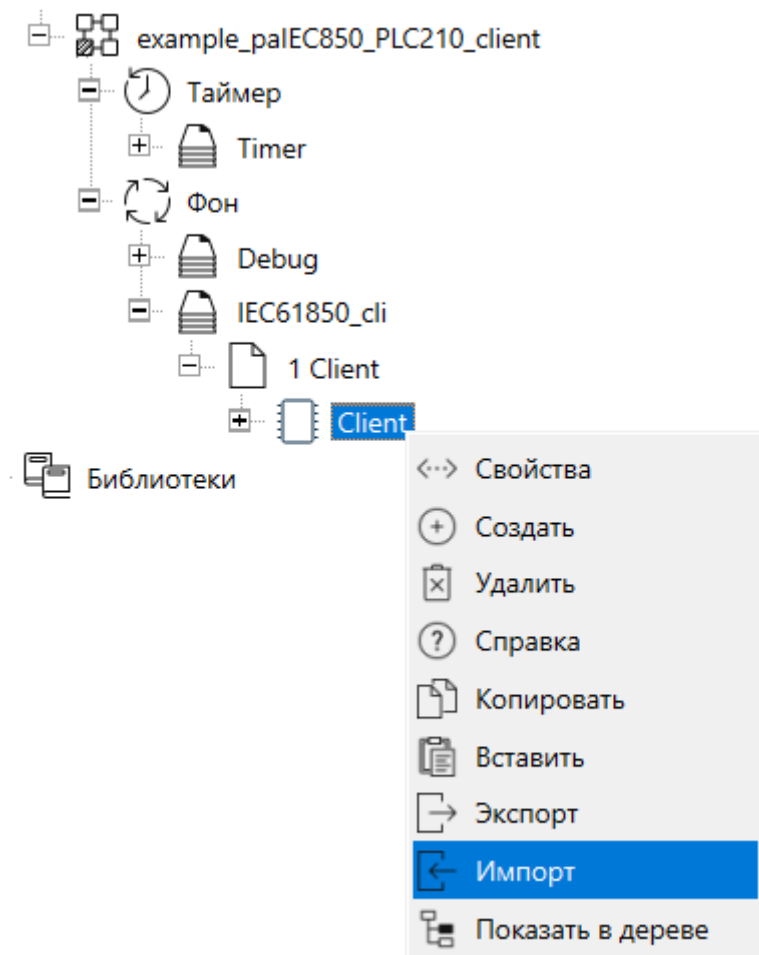


Рисунок 4.2 – Импорт конфигурации сервера

3. В появившемся окне выбрать необходимый файл с расширением ***.cid**.

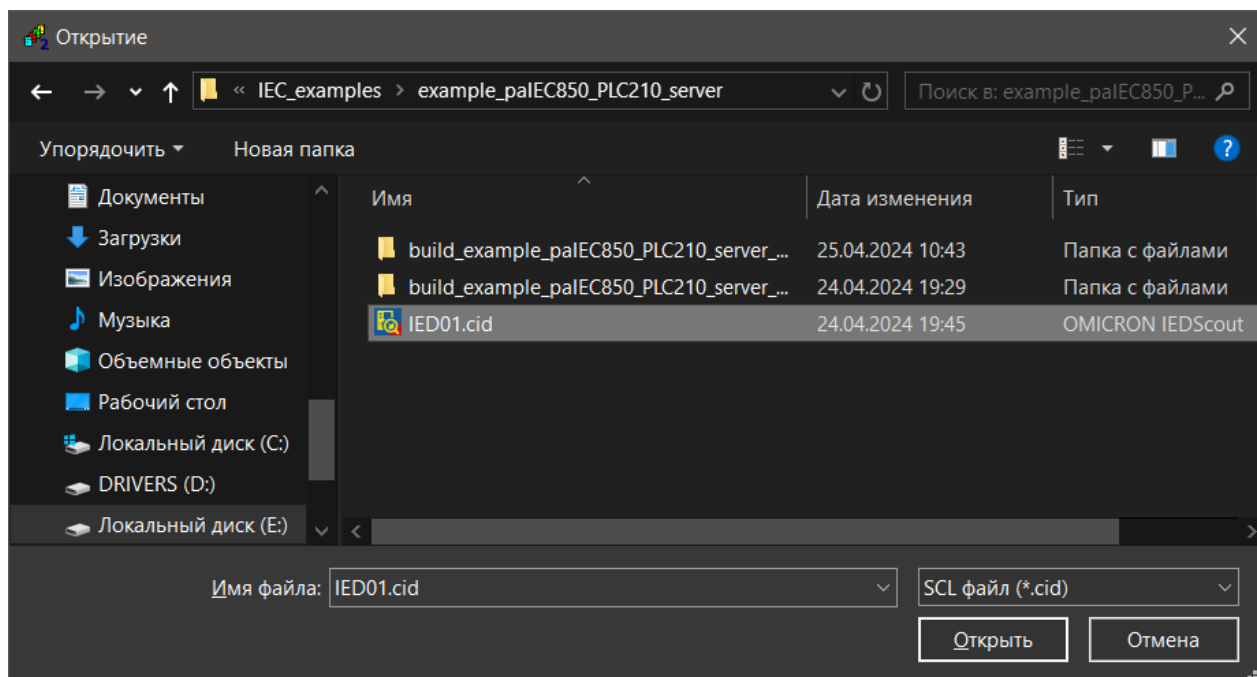


Рисунок 4.3 – Файл *.cid

4. В качестве сервера здесь используется ПЛК210, настроенный в [разделе 3](#). Необходимо импортировать его конфигурацию (пример сохранения конфигурации сервера описан в [разделе 3.3](#)). Модель данных сервера отобразится в дереве проекта.

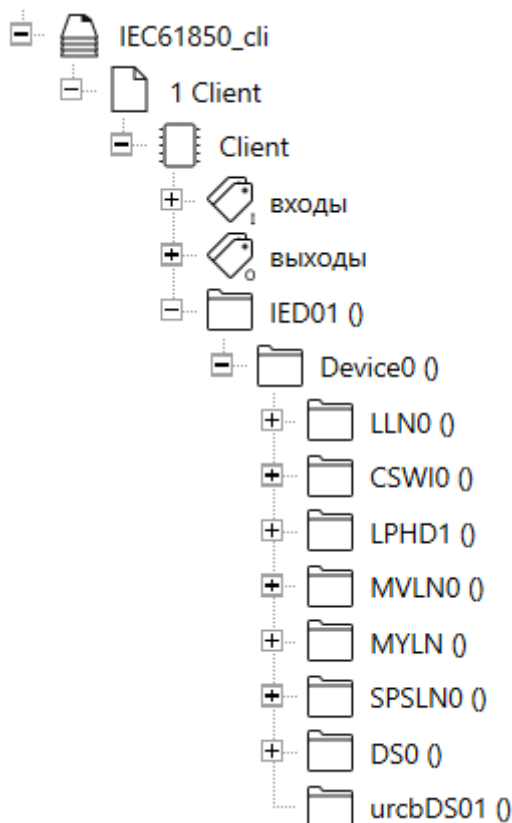


Рисунок 4.4 – Успешный импорт конфигурации сервера

5. В узле **IED01** задать IP-адрес и порт сервера.

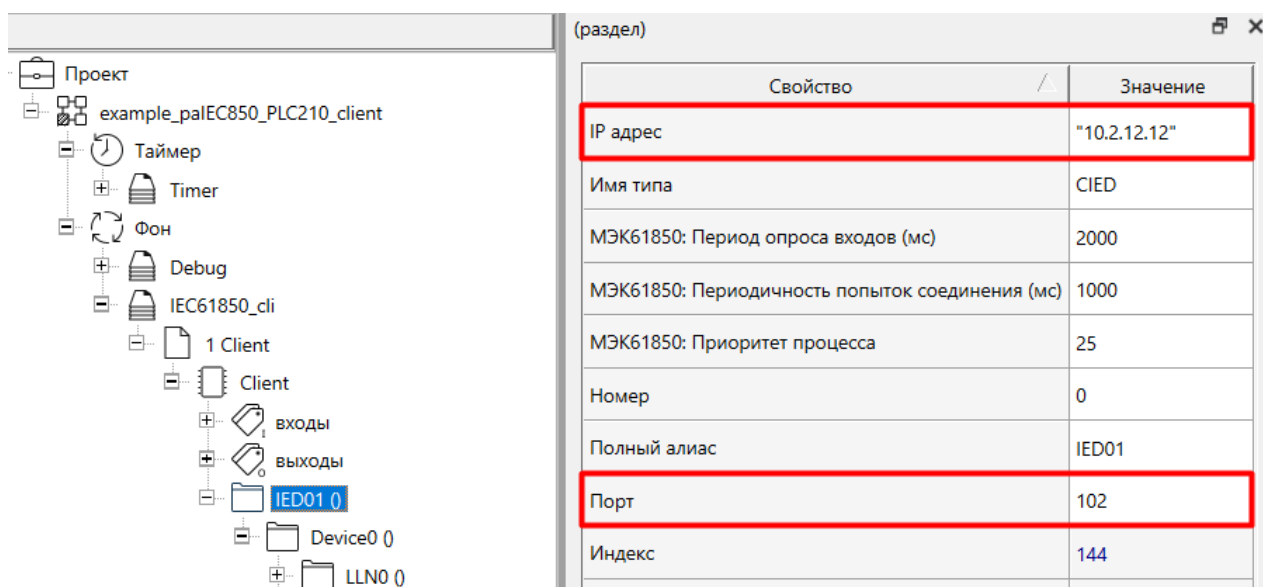


Рисунок 4.5 – Настройка IED01

6. В узле управления отчетом **urcbDS01** установить свойство **МЭК61850: включить отчет**.

Установленные ранее условия передачи отчета также активировались. Здесь можно добавить другие причины или убрать установленные.

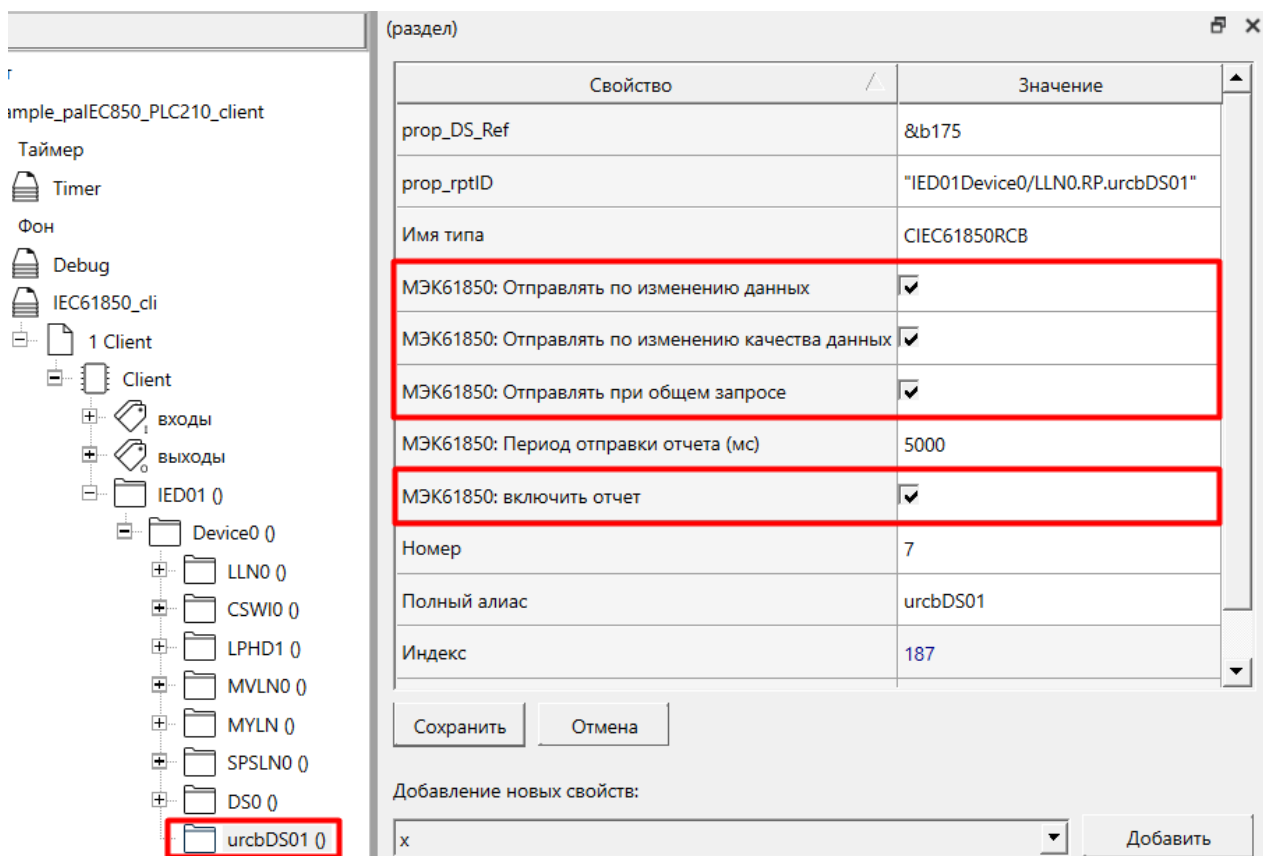


Рисунок 4.6 – Настройка IED01

7. Для получения данных протокола в проекте следует связать его атрибуты с входами/выходами блоков в проекте.

Для передачи контролируемых объектов используется вспомогательный блок **IEC61850CmdCSWI**. Для его подключения следует добавить блоку **IEC61850cli** коннектор **bo**.

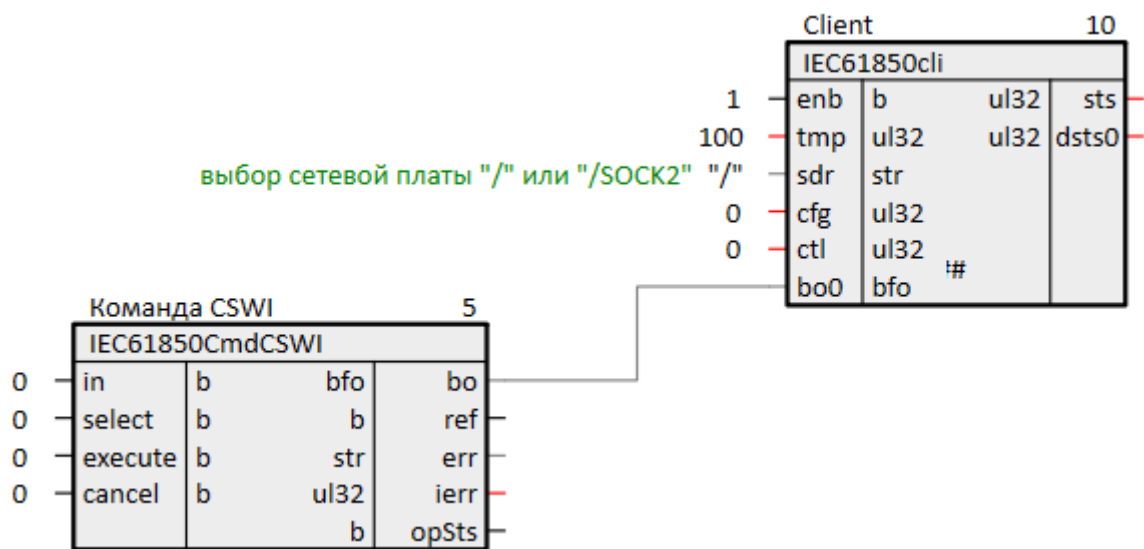


Рисунок 4.7 – Подключение блока IEC61850CmdCSWI

8. Для того, чтобы указать, к какому объекту данных относится данный блок выполнения команды, следует найти в дереве проекта в узле **CSWI0** атрибут **IED01Device0/CSWI0.Pos**. Зажать **Ctrl** и перетащить на атрибут выход блока **ref**. В появившемся меню выбрать команду **Назначить**.

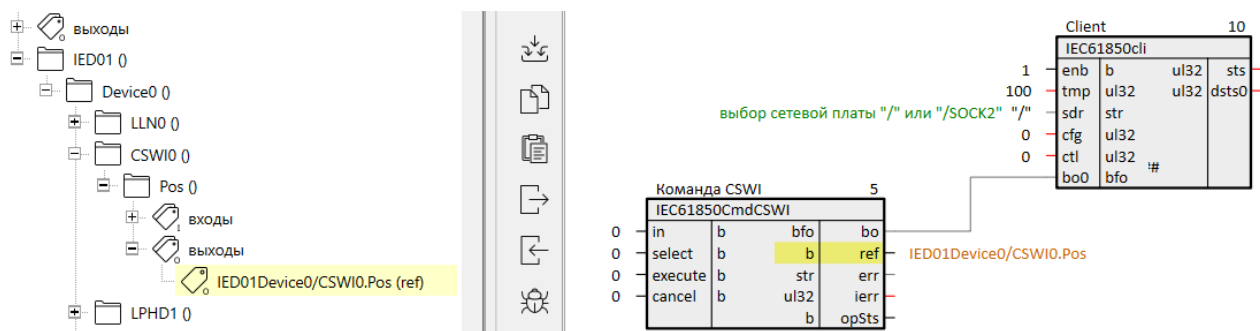


Рисунок 4.8 – Подключение DO к IEC61850CmdCSWI

9. Чтобы получать в программе значение измеряемой величины, нужно создать блок **TransFit** из библиотеки **paCore**.

**ПРИМЕЧАНИЕ**

Если необходимо получать объекты данных с отчетом, то связывать с блоками в программе следует именно объекты данных в отчете (в примере в узле **DS0**). Если необходимо периодически считывать данные и производить запись по изменению, то блоки программы следует соединять с объектами в дереве модели данных.

10. С помощью команды **Назначить** связать атрибут **IED01Device0/MVLN0.MV.mag.f** с входом блока **TransFit**. Аналогично связать вход блока **TransBit** из библиотеки **paCore** с атрибутом **IED01Device0/SPSLN0.SPS.stVal**.

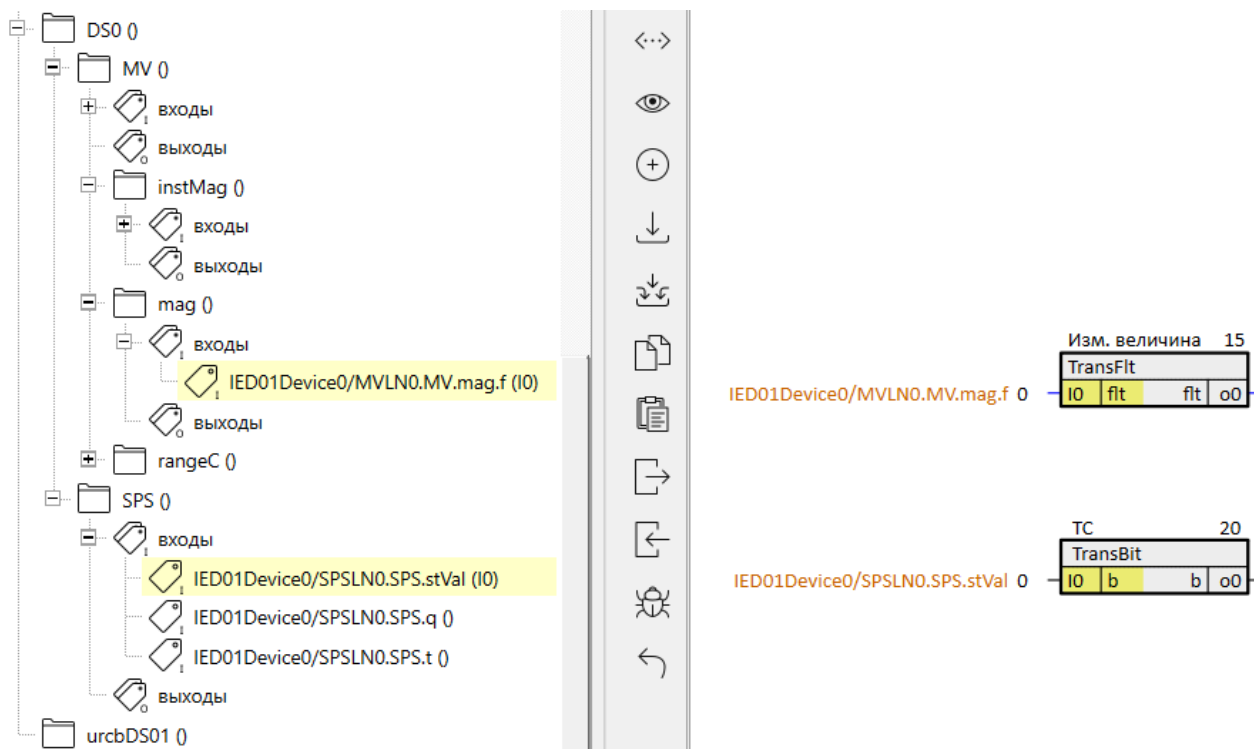


Рисунок 4.9 – Подключение данных отчета к блокам в проекте

11. Для расшифровки метки качества использовать вспомогательный блок **IEC61850TransQuality**.

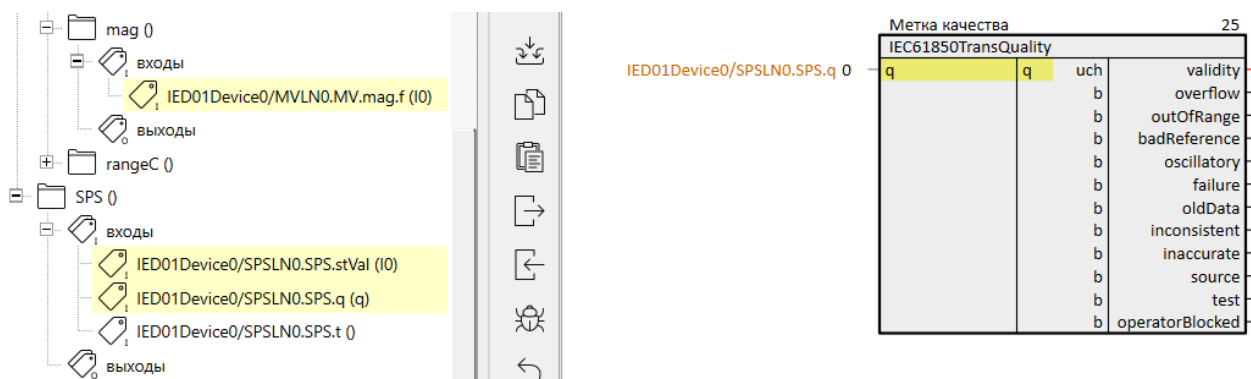


Рисунок 4.10 – Подключение метки качества к IEC61850TransQuality

12. Для чтения/записи атрибута данных **MYDA** нужно подключить его к блоку **TransInt32** из библиотеки **paCore**, как показано на рисунке ниже.

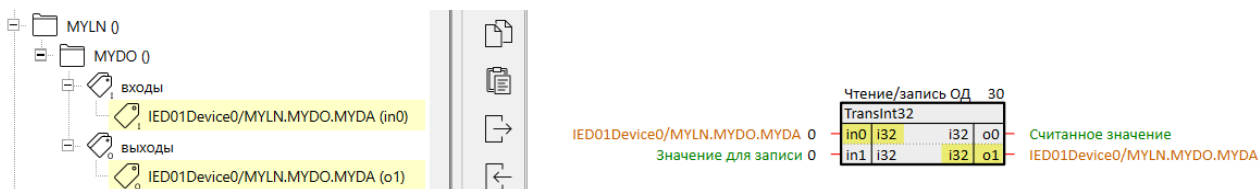


Рисунок 4.11 – Подключение клиента к серверу

13. Запустить проекты на обоих контроллерах.

Если подключение к серверу **IED01** прошло успешно, то на выходе **dsts** блока **IEC61850cli** появится **1**.

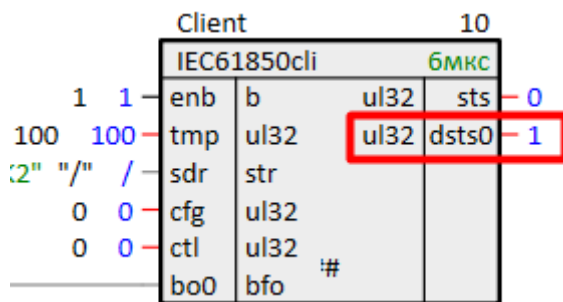


Рисунок 4.12 – Подключение MYDA к TransInt32

Успешная передача отчета показана на рисунках ниже.

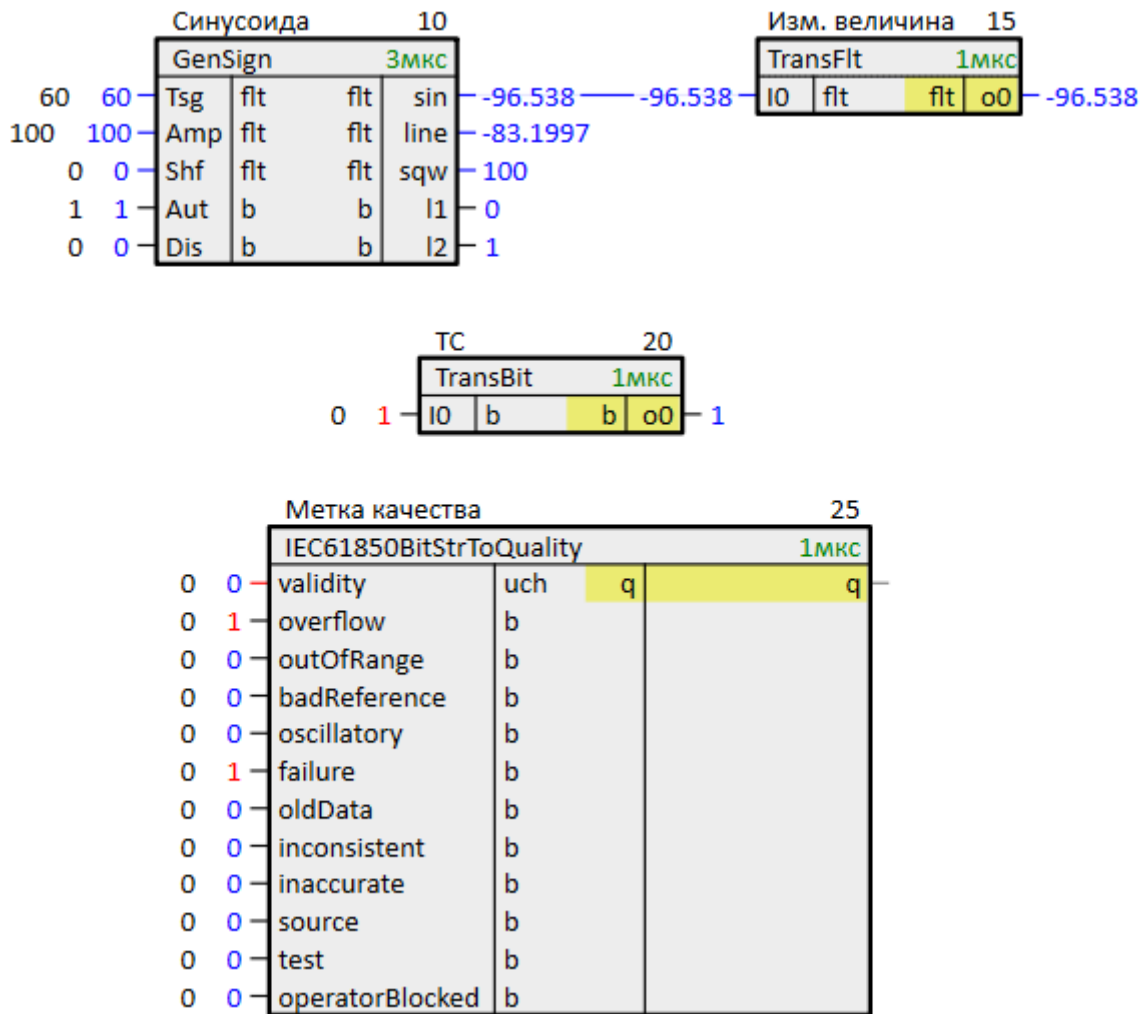


Рисунок 4.13 – Успешный обмен. Сервер

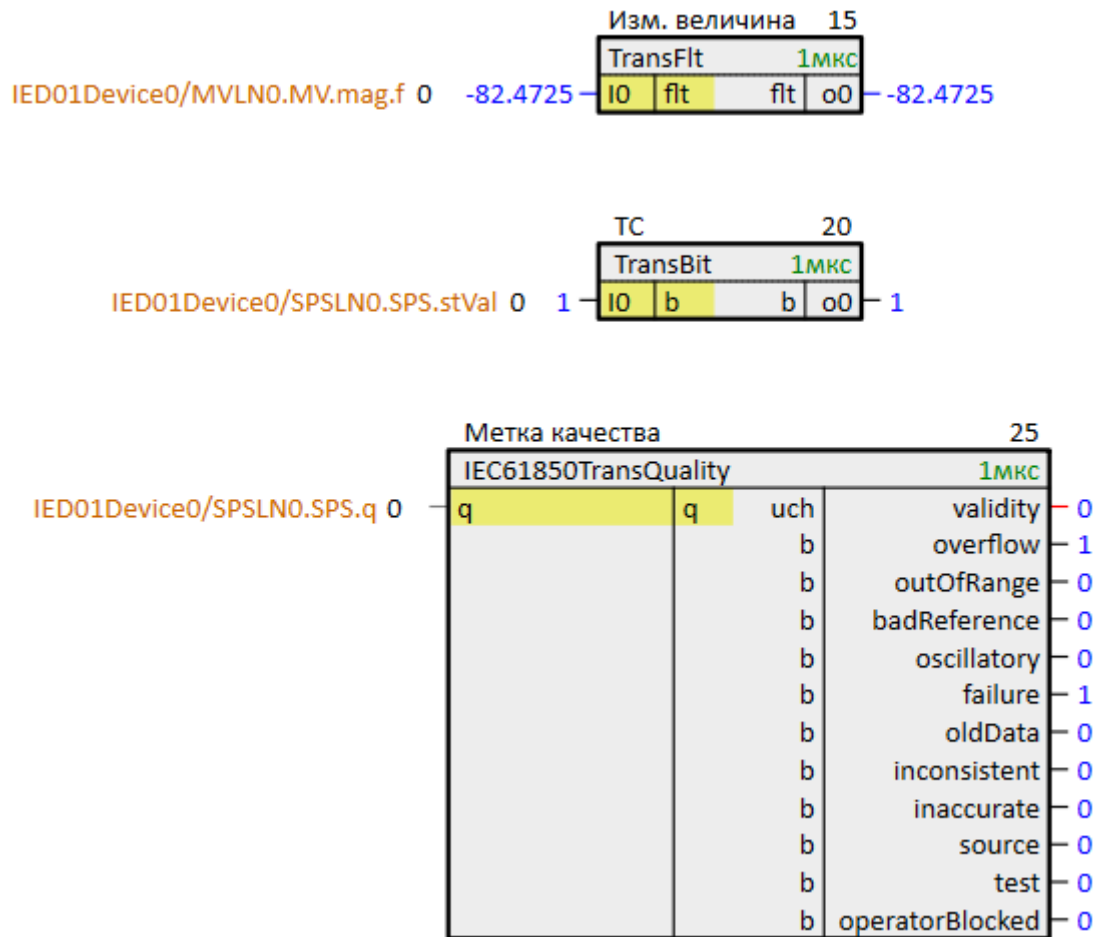


Рисунок 4.14 – Успешный обмен. Клиент

Для отправки команды **CSWI** следует подать 1 на вход **in** блока **IEC61850CmdCSWI**, затем подать 1 на вход **select**, так как в сервере настроена модель управления с предварительных выбором, и затем подать 1 на вход **execute**.

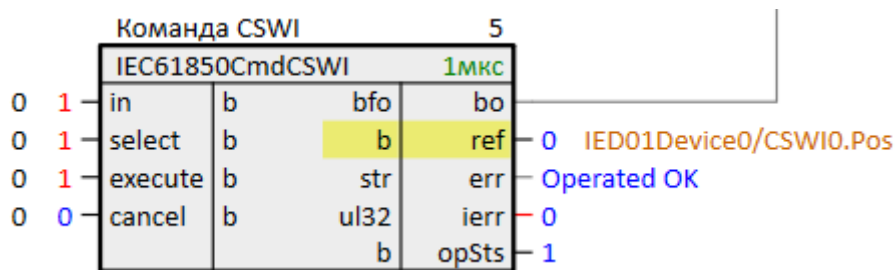


Рисунок 4.15 – Команда ТУ. Клиент

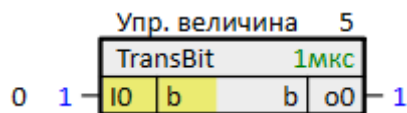


Рисунок 4.16 – Команда ТУ. Сервер

Затем можно установить новое значение **MYDA**.

Частота чтения объектов данных задается свойством **МЭК61850: Период опроса входов** узла **IED01**.

Запись происходит по изменению значения.

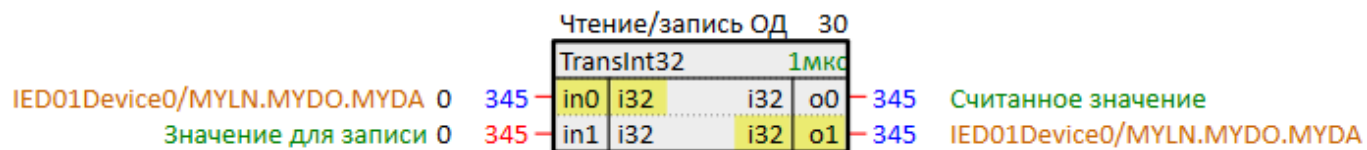


Рисунок 4.17 – Изменение атрибута MYDA. Клиент

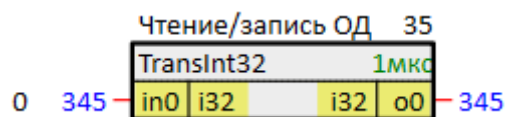


Рисунок 4.18 – Изменение атрибута MYDA. Сервер

5 Диагностика обмена

5.1 Диагностика обмена сервера IEC61850srv

Для получения диагностических сообщений сервера [IEC61850srv](#) следует запустить программу через консоль.

Типы выводимых диагностических сообщений задаются битами **8...15** на входе **ctl**.

Бит 8 отвечает за сообщения высокого приоритета (например, ошибка подключения к устройству), а **бит 15** за сообщения низкого приоритета (например, содержание буферов приема/передачи).

Для включения вывода всех сообщений следует подать на вход **0xFF00**.

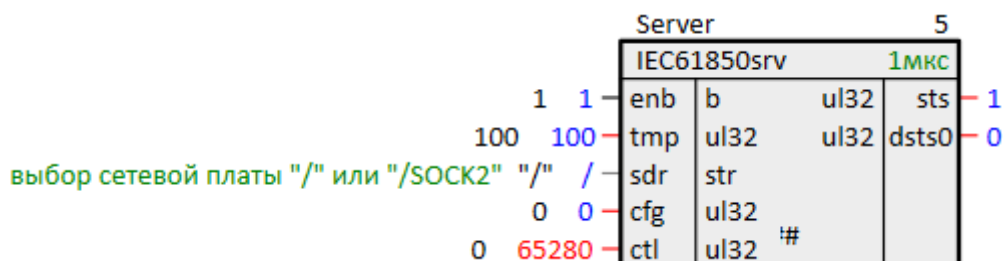


Рисунок 5.1 – Включение вывода всех диагностических сообщений сервера

```

root@plc210rk_12_polygon: ~
10:36:02.233 ==== IEC61850srv(0) -4- Server == Read Access For Object IED01Device
0/MYLN.MYDO (CIEC61850srv::readAccessHandler)
10:36:04.248 ==== IEC61850srv(0) -4- Server == Read Access For Object IED01Device
0/MYLN.MYDO (CIEC61850srv::readAccessHandler)
10:36:04.800 ==== IEC61850srv(0) -3- Server == Performing Check For IED01Device0/
CSWI0.Pos (CIEC61850srv::performCheckHandler)
10:36:06.255 ==== IEC61850srv(0) -4- Server == Read Access For Object IED01Device
0/MYLN.MYDO (CIEC61850srv::readAccessHandler)
10:36:06.446 ==== IEC61850srv(0) -3- Server == Performing Check For IED01Device0/
CSWI0.Pos (CIEC61850srv::performCheckHandler)
10:36:06.447 ==== IEC61850srv(0) -3- Server == Control Access For IED01Device0/CS
WI0.Pos (CIEC61850srv::controlHandler)
10:36:06.448 ==== IEC61850srv(0) -4- Server == Copying MmsValue(true) MmsType(MMS
_BOOLEAN (boolean value)) To Assigned Input OPCUA Type(1)... (CIEC61850srv::proce
ssInp)
10:36:08.261 ==== IEC61850srv(0) -4- Server == Read Access For Object IED01Device
0/MYLN.MYDO (CIEC61850srv::readAccessHandler)
10:36:10.276 ==== IEC61850srv(0) -4- Server == Read Access For Object IED01Device
0/MYLN.MYDO (CIEC61850srv::readAccessHandler)
10:36:12.291 ==== IEC61850srv(0) -4- Server == Read Access For Object IED01Device
0/MYLN.MYDO (CIEC61850srv::readAccessHandler)
10:36:14.307 ==== IEC61850srv(0) -4- Server == Read Access For Object IED01Device
0/MYLN.MYDO (CIEC61850srv::readAccessHandler)

```

Рисунок 5.2 – Вывод диагностических сообщений в консоль IEC61850srv



ПРИМЕЧАНИЕ

Если на входе **ctl** блока **IEC61850srv** задать **0**, то в консоль будут выводиться диагностические сообщения с типом **1**.

5.2 Диагностика обмена клиента IEC61850cli

Для того, чтобы получать диагностические сообщения [IEC61850cli](#), следует запустить программу через консоль.

Типы выводимых диагностических сообщений задаются битами **8...15** на входе **ctl**.

Бит 8 отвечает за сообщения высокого приоритета (например, ошибка подключения к устройству), а **бит 15** за сообщения низкого приоритета (например, содержание буферов приема/передачи).

Таким образом, включается вывод диагностических сообщений самого клиента *IEC61850cli*.

		Client		10
		IEC61850cli		1мкс
1	1	enb	b ul32	sts 0
100	100	tmp	ul32 ul32	dsts0 1
		sdr	str	
0	0	cfg	ul32	
0	65280	ctl	ul32	
		bo0	bfo #	

выбор сетевой платы "/" или "/SOCK2" "/" /

Рисунок 5.3 – Включение вывода всех диагностических сообщений клиента

Чтобы включить диагностику для каждого подключенного сервера *IED*, следует выполнить действия:

1. Добавить и установить в корневом узле *IED* свойство *Переменные*.

Свойство	Значение
IP адрес	"10.2.12.12"
Имя типа	CIED
МЭК61850: Период опроса входов (мс)	2000
МЭК61850: Периодичность попыток соединения (мс)	1000
МЭК61850: Приоритет процесса	25
Номер	0
Переменные	<input checked="" type="checkbox"/>
Полный алиас	IED01
Порт	102
Индекс	144
Принадлежит	143

Рисунок 5.4 – Узел *IED*. Свойство переменные

2. В появившийся раздел **выходы** добавить выход блока *TransUInt32* из библиотеки *paCore*. Для этого следует перетащить выход блока с зажатым **Ctrl** на узел *IED* и в выпадающем меню выбрать **Добавить**.

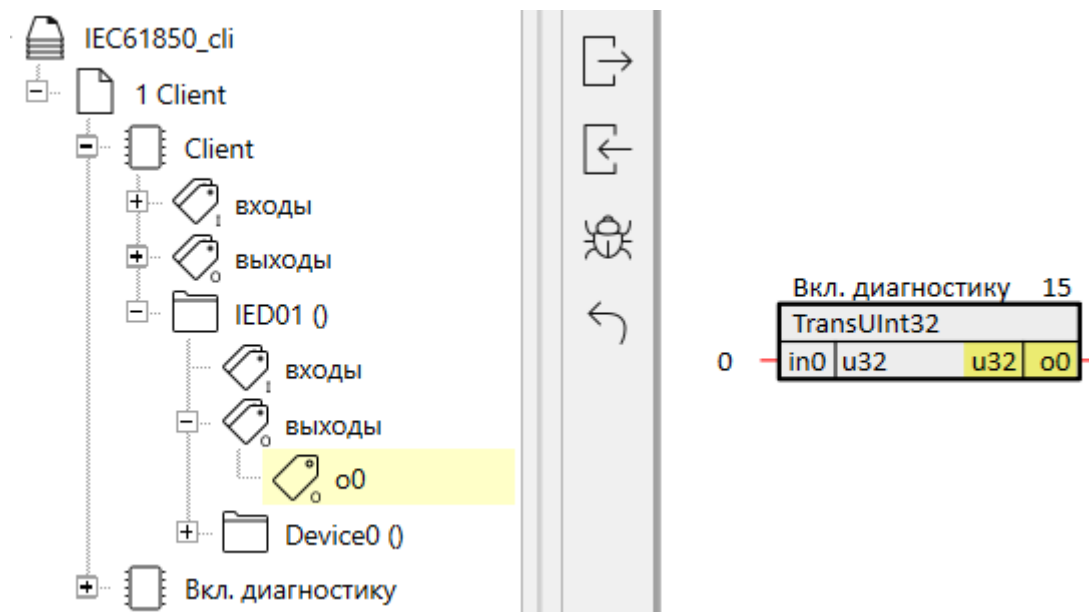


Рисунок 5.5 – Узел IED. Добавление выхода TransUInt32

3. Добавить выходу *TransUInt32* свойство *Полный алиас* со значением *Control*.

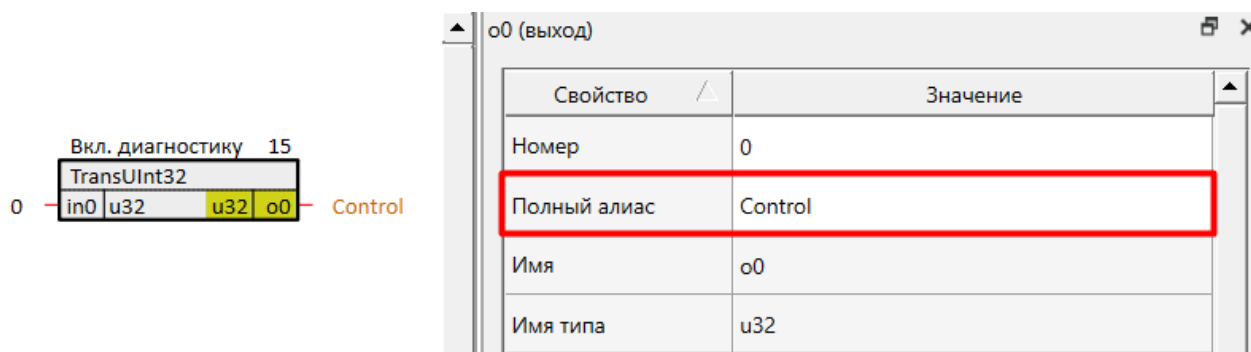


Рисунок 5.6 – Свойство Полный алиас выхода TransUInt32

Если задать 1 на входе *TransUInt32*, в консоль будут выводиться сообщения для данного устройства 61850.

```

root@plc210rk_14_pl: ~
ut OPCUA Type(10) CID Type(FLOAT32)... (CIEC61850LD::copyMmsValue)
14:05:30.613 ==== IEC61850cli(0) -2- IED01Device0 == curVal == NULL! (CIEC61850LD::processReportTree)
14:05:31.259 ==== IEC61850cli(0) -3- IED01Device0 == Received report IED01Device0/LLN0.RP.urcbDS01 (CIEC61850LD::reportCallbackFunction)
14:05:31.259 ==== IEC61850cli(0) -4- IED01Device0 == Found Input IED01Device0/MV.LLN0.MV.mag.f, m_indx(0), data(-81.875710), Reason(the element is included due to a change of the data value) (CIEC61850LD::processReportObject)
14:05:31.259 ==== IEC61850cli(0) -4- IED01Device0 == Copying MmsValue(-81.875710) MmsType(MMS_FLOAT (represents all float type (32 and 64 bit))) To Assigned Input OPCUA Type(10) CID Type(FLOAT32)... (CIEC61850LD::copyMmsValue)
14:05:31.259 ==== IEC61850cli(0) -2- IED01Device0 == curVal == NULL! (CIEC61850LD::processReportTree)
14:05:31.986 ==== IEC61850cli(0) -3- IED01Device0 == Received report IED01Device0/LLN0.RP.urcbDS01 (CIEC61850LD::reportCallbackFunction)
14:05:31.987 ==== IEC61850cli(0) -4- IED01Device0 == Found Input IED01Device0/MV.LLN0.MV.mag.f, m_indx(0), data(-85.967972), Reason(the element is included due to a change of the data value) (CIEC61850LD::processReportObject)
14:05:31.987 ==== IEC61850cli(0) -4- IED01Device0 == Copying MmsValue(-85.967972) MmsType(MMS_FLOAT (represents all float type (32 and 64 bit))) To Assigned Input OPCUA Type(10) CID Type(FLOAT32)... (CIEC61850LD::copyMmsValue)
14:05:31.987 ==== IEC61850cli(0) -2- IED01Device0 == curVal == NULL! (CIEC61850LD::processReportTree)

```

Рисунок 5.7 – Вывод диагностических сообщений в консоль IEC61850cli

**ПРИМЕЧАНИЕ**

Если задать **0** на входе **ctl** блока **IEC61850cli**, то в консоль будут выводиться диагностические сообщения с типом **1**.

Приложение А. Поддерживаемые сервисы ACSI

Таблица А.1 – Поддерживаемые сервисы ACSI

Сервис	Поддержка в раIEC850
Модель SERVER (сервер)	
GetServerDirectory	Да
Модель ASSOCIATION (ассоциация)	
Associate	Да
Abort	Да
Release	Да
Модель LOGICAL-DEVICE (логическое устройство)	
GetLogicalDeviceDirectory	Да
Модель LOGICAL-NODE (логический узел)	
GetLogicalNodeDirectory	Да
GetAllDataValues	Да
Модель DATA (данные)	
GetDataValues	Да
SetDataValues	Да
GetDataDefinition	Да
GetDataDirectory	Да
Модель DATA-SET (набор данных)	
GetDataSetValues	Да
DataSetValues	Да
CreateDataSet	Нет
DeleteDataSet	Нет
GetDataSetDirectory	Да
Модель подстановки	
SetDataValues	Да
GetDataValues	Да
Модель SETTING-GROUP-CONTROL-BLOCK (блок управления группой настроек)	
SelectActiveSG	Нет
SelectEditSG	Нет
SetSGValues	Нет
ConfirmEditSGValues	Нет
GetSGValues	Нет
GetSGCBValues	Нет
Модель REPORT-CONTROL-BLOCK (блок управления генерацией отчетов) и модель LOG-CONTROL-BLOCK (блок управления журналом)	
BUFFERED-REPORT-CONTROL-BLOCK (блок управления буферизованным отчетом)	
Report	Да
GetBRCBValues	Да
SetBRCBValues	Да
UNBUFFERED-REPORT-CONTROL-BLOCK (блок управления небуферизованным отчетом)	
Report	Да
GetURCBValues	Да
SetURCBValues	Да

Продолжение таблицы А.1

Сервис	Поддержка в раIEC850
LOG-CONTROL-BLOCK (блок управления журналом)	
GetLCBValues	Нет
SetLCBValues	Нет
QueryLogByTime	Нет
QueryLogAfter	Нет
GetLogStatusValues	Нет
Модель общих событий подстанции – GSE	
GOOSE	
SendGOOSEMessage	Нет
GetGoReference	Нет
GetGOOSEElementNumber	Нет
GetGoCBValues	Нет
SetGoCBValues	Нет
GSSE	
SendGSSEMessage	Нет
GetGsReference	Нет
GetGSSEDataOffset	Нет
GetGsCBValues	Нет
SetGsCBValues	Нет
Модель передачи выборочных значений	
MULTICAST-SAMPLE-VALUE-CONTROL-BLOCK (блок управления многоадресным выборочным значением)	
SendMSVMessage	Нет
GetMSVCBValues	Нет
SetMSVCBValues	Нет
UNICAST-SAMPLED-VALUE-CONTROL-BLOCK (блок управления одноадресным выборочным значением)	
SendUSVMessage	Нет
GetUSVCBValues	Нет
SetUSVCBValues	Нет
Модель управления	
Select	Да
SelectWithValue	Да
Cancel	Да
Operate	Да
CommandTermination	Да
TimeActivatedOperate	Нет
Время и временная синхронизация	
TimeSynchronization	Да
Модель передачи FILE (файла)	
GetFile	Нет
SetFile	Нет
DeleteFile	Нет
GetFileAttributeValues	Нет

Приложение Б. Функциональные ограничения (Functional Constraints)

Таблица Б.1 – Функциональные ограничения (Functional Constraints)

Обозначение	Название	Разрешенные сервисы протокола
ST	Информация о состоянии	Атрибут данных DA должен предоставлять информацию о состоянии в значениях, которые могут быть считаны, включены в отчет, но не могут быть записаны
MX	Измеряемая величина	Атрибут данных DA должен предоставлять информацию об измеряемой величине, значение которой может быть считано, включено в отчет, но не может быть записано
CO	Управление	Атрибут данных DA должен предоставлять информацию управления, значением которой можно оперировать (модель управления) и которое можно считывать
SP	Уставка	Атрибут данных DA должен предоставлять информацию уставки, значением которой можно управлять (модель управления) и которое можно считывать
SV	Подстановка	Атрибут данных DA должен предоставлять информацию подстановки, значение которой можно записывать и считывать
CF	Конфигурация	Атрибут данных DA должен предоставлять информацию конфигурации, значение которой можно записывать и считывать
DC	Описание	Атрибут данных DA должен предоставлять информацию описания, значение которой можно записывать и считывать
SG	Группа настроек	Логические устройства, реализующие класс SGCB , поддерживают сгруппированное множество значений атрибутов DA с FC = SG , и только одно текущее значение в каждой группе для каждого атрибута должно быть активным
SE	Редактируемая группа настроек	Атрибут данных DA , который может быть изменен сервисами SGCB
EX	Расширенное определение	Атрибут данных DA должен предоставлять информацию по расширению, обеспечивающую ссылку на пространство имен



Россия, 111024, Москва, 2-я ул. Энтузиастов, д. 5, корп. 5
тел.: +7 (495) 641-11-56, факс: (495) 728-41-45
тех. поддержка 24/7: 8-800-775-63-83, support@owen.ru
отдел продаж: sales@owen.ru
Веб-сайт ООО "ПромАвтоматика-Софт": www.pa.ru
рег.:1-RU-135061-1.2