

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ЭВМ
_____ Д.В. Топольский
« ____ » _____ 2025 г.

Информационная модель цифровой подстанции

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУРГУ-090301.2025.406 ПЗ ВКР

Руководитель работы,
к.т.н., доцент каф. ЭВМ
_____ Д.В. Топольский
« ____ » _____ 2025 г.

Автор работы,
студент группы КЭ-406
_____ А.А. Пономарев
« ____ » _____ 2025 г.

Нормоконтролёр,
ст. преподаватель каф. ЭВМ
_____ С.В. Сяськов
« ____ » _____ 2025 г.

Челябинск-2025

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____ Д.В. Топольский
«__» _____ 2025 г.

ЗАДАНИЕ
на выпускную квалификационную работу бакалавра
студенту группы КЭ-406
Пономареву Артёму Анатольевичу
обучающемуся по направлению
09.03.01 «Информатика и вычислительная техника»

1. **Тема работы:** «Информационная модель цифровой подстанции»
утверждена приказом по университету от «21» апреля 2025 г. №
648-13/12.
2. **Срок сдачи студентом законченной работы:** 01 июня 2025 г.
3. **Исходные данные к работе:**
 - 3.1. Оборудование цифровой подстанции 110 кВ.
 - 3.1.1. Цифровой измерительный трансформатор:
 - класс напряжения: 110 кВ;
 - номинальный первичный ток: 200–2000 А;
 - класс точности измерения тока для АСКУЭ: 0,2;
 - класс точности измерения тока для РЗА: 5;
 - класс точности измерения напряжения: 0,2;
 - протокол передачи данных: МЭК 61850-9-2;
 - габаритные размеры: 1540×320×450 мм;

- масса: 120 кг.

3.1.2. Интеллектуальные электронные устройства (IED):

- тип: Реле защиты SEL-487В (поддержка МЭК 61850);
- функции: защита от коротких замыканий, дифференциальная защита;
- интерфейсы: Ethernet, RS-485;
- протоколы: МЭК 61850-8-1 (GOOSE), DNP3.

3.1.3. Коммутационные аппараты:

- выключатели: Вакуумные выключатели Schneider Electric EV12s;
- управление: цифровые контроллеры с поддержкой МЭК 61850;
- диапазон тока: 2000 А;
- время срабатывания: 20 мс.

3.1.4. Программно-аппаратная среда:

- среда разработки: Python (PyCharm);
- библиотеки для моделирования: Matplotlib Pandas, PySV, pyiec61850, dnp3-python, open62541, SQLite;
- протоколы связи: МЭК 61850, DNP3;
- SCADA-система: MasterSCADA или аналог.

4. Перечень вопросов, подлежащих разработке:

- 4.1. Аналитический обзор литературы.
- 4.2. Проектирование архитектуры информационной модели.
- 4.3. Программная реализация модели.
- 4.4. Моделирование и тестирование.

5. Дата выдачи задания: «02» декабря 2024 г.

Руководитель работы _____/Д.В. Топольский/

Студент _____/А.А. Пономарев/

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	03.03.2025	
Проектирование архитектуры информационной модели	22.03.2025	
Программная реализация информационной модели	12.04.2025	
Тестирование информационной модели	26.04.2025	
Компоновка текста работы и сдача на нормоконтроль	22.05.2025	
Подготовка презентации и доклада	30.05.2025	

Руководитель работы _____ / Д.В Топольский /

Студент _____ / А.А. Пономарев /

Аннотация

А.А. Пономарев. Информационная модель цифровой подстанции. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2025, 45 с., 8 ил., библиогр. список – 15 наим.

В работе рассматриваются методы построения информационной модели цифровой подстанции 110 кВ, базирующейся на протоколах МЭК 61850 и DNP3 для сбора, обработки и визуализации данных от цифрового трансформатора, реле защиты SEL-487В и SCADA. Выявлены проблемы интеграции разнородного оборудования и обеспечена совместимость с российскими нормативами. Целью стало создание прототипа, реализующего приём SV- и GOOSE-сообщений, DNP3-событий и логов SCADA с сохранением в SQLite и последующей визуализацией. В результате разработаны модули приёма данных с использованием различных библиотек.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	7
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	8
1.1. ОБЗОР АНАЛОГОВ	9
1.2. АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ.....	12
1.3. ВЫВОДЫ ПО 1 РАЗДЕЛУ	16
2. ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ.....	17
2.1. АНАЛИЗ ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ	17
2.2. АНАЛИЗ НЕФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ	18
2.3. АНАЛИЗ И ОПРЕДЕЛЕНИЕ КЛЮЧЕВЫХ КОМПОНЕНТОВ .	19
2.4. РАЗРАБОТКА КОНЦЕПТУАЛЬНОЙ МОДЕЛИ	21
2.4.1. ОПИСАНИЕ СУЩНОСТЕЙ СИСТЕМЫ.....	21
2.4.2. ОПРЕДЕЛЕНИЕ СВЯЗЕЙ МЕЖДУ СУЩНОСТЯМИ.....	23
2.5. РАЗРАБОТКА ЛОГИЧЕСКОЙ МОДЕЛИ АРХИТЕКТУРЫ.....	24
2.6. ВЫВОДЫ ПО 2 РАЗДЕЛУ	25
3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ.....	26
3.1. НАСТРОЙКА И ИНТЕГРАЦИЯ ПРИЁМА SV-ПОТОКОВ	27
3.2. РЕАЛИЗАЦИЯ ПАРСЕРА GOOSE-СООБЩЕНИЙ И DNP3.....	29
3.3. РЕАЛИЗАЦИЯ ПРИЁМА ЛОГОВ ОТ SCADA	31
3.4. ВИЗУАЛИЗАЦИЯ ДАННЫХ	33
3.5. ВЫВОД ПО 3 РАЗДЕЛУ	34
4. ТЕСТИРОВАНИЕ СИСТЕМЫ	35
4.1. ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ.....	35
4.2. НЕФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ.....	37
4.3. ВЫВОД ПО 4 РАЗДЕЛУ	38
5. ЗАКЛЮЧЕНИЕ.....	39
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	40
ПРИЛОЖЕНИЕ А. ЛИСТИНГИ ПРОГРАММНОГО КОДА.....	42

ВВЕДЕНИЕ

Цифровая подстанция представляет собой современный этап развития энергетических систем, основанный на интеграции интеллектуальных устройств, цифровых технологий связи и автоматизации процессов. Информационная модель цифровой подстанции выступает ключевым элементом этой трансформации, обеспечивая унифицированное представление данных, управление оборудованием и поддержку решений в реальном времени.

Переход к цифровым подстанциям обусловлен глобальными вызовами: ростом сложности энергосистем, необходимостью повышения надежности, снижения эксплуатационных затрат и интеграции возобновляемых источников энергии.

Вопросы цифровизации подстанций активно исследуются в работах зарубежных и отечественных ученых. Значительный вклад внесли исследования в области стандартов МЭК 61850 [1], архитектуры SCADA-систем [2] и применения ВМ-технологий в энергетике. Однако недостаточно изучены аспекты адаптации информационных моделей к условиям российских энергосетей, включая интеграцию с устаревшим оборудованием и обеспечение кибербезопасности. Это определяет нишу для данного исследования.

Решение задач осуществляется с применением системного анализа, Методов математического моделирования и экспериментального тестирования на примере типовой подстанции 110/10 кВ [3]. Результаты исследования могут быть применены при модернизации действующих подстанций, проектировании новых объектов и обучении специалистов в области Smart Grid. Работа вносит вклад в развитие методов цифровизации энергетики, сочетая теоретическую глубину с прикладной направленностью.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Цифровые подстанции (ЦПС) – это инновационный этап развития энергетической инфраструктуры, направленный на трансформацию традиционных объектов в интеллектуальные системы с автоматизированным управлением, мониторингом и прогнозированием. Переход к цифровизации энергосистем обусловлен глобальными вызовами: ростом спроса на электроэнергию, необходимостью интеграции возобновляемых источников энергии (ВИЭ), повышением требований к надежности и безопасности. В отличие от классических подстанций, где доминируют аналоговые устройства и ручные операции, ЦПС базируются на цифровых технологиях, таких как:

1. Информационные модели (включая цифровых двойников), обеспечивающие виртуальное представление физических объектов.
2. Интеллектуальные электронные устройства (IED) с поддержкой стандарта МЭК 61850, гарантирующего совместимость оборудования.
3. Распределенные системы управления (SCADA, EMS) с использованием облачных вычислений и Big Data.

Ключевой элемент цифровой подстанции – её информационная модель, которая объединяет данные от всех компонентов (трансформаторов, релейной защиты, систем диагностики) в единое цифровое пространство. Это позволяет оптимизировать процессы эксплуатации, например, сократить время реакции на аварии с часов до секунд, а также снизить эксплуатационные затраты за счет предиктивной аналитики. Однако внедрение таких моделей сопряжено с рядом проблем:

1. Технологические: интеграция устаревшего оборудования (до 40% парка российских подстанций не поддерживают цифровые интерфейсы).

2. Регуляторные: несоответствие между международными стандартами (МЭК 61850) и российскими нормами (ГОСТ Р 58698-2019).
3. Киберриски: уязвимость цифровых систем к хакерским атакам, требующая разработки специализированных протоколов защиты [4].

Мировой опыт (ЕС, США, Китай) демонстрирует успешные случаи внедрения ЦПС. Например, в Германии цифровизация подстанций позволила на 30% повысить эффективность интеграции ВИЭ в сеть [5]. В России пилотные проекты (ПАО «Россети», «ФСК ЕЭС») находятся на стадии тестирования, но уже показывают снижение аварийности [6]. Тем не менее, масштабное распространение сдерживается недостатком специализированных решений, адаптированных к местным условиям, и высокой стоимостью модернизации.

Таким образом, разработка информационной модели цифровой подстанции актуальна не только как техническая задача, но и как стратегический шаг к созданию устойчивой, гибкой и безопасной энергосистемы, соответствующей требованиям Industry 4.0 [7].

1.1. ОБЗОР АНАЛОГОВ

Современные решения в области цифровых подстанций демонстрируют широкий спектр подходов к интеграции информационных моделей, что позволяет оптимизировать эксплуатацию энергосистем. Анализ существующих аналогов показывает, что ведущие мировые компании и отечественные разработчики создают высокотехнологичные системы, каждая из которых обладает своими преимуществами, а также определёнными особенностями, требующими внимания при внедрении [8].

Компания «Прософт-Системы» разработала платформу MasterSCADA Digital Substation как полнофункциональное решение для автоматизации цифровых подстанций классов 110–220 кВ. Система строится на трёхуровневой архитектуре, в которой полевое оборудование (IED и сетевые

коммутаторы) подключается к серверному уровню для сбора и архивации данных, а сверху располагаются рабочие станции операторов и визуализационные HMI-панели. Благодаря нативной поддержке протоколов МЭК 61850 и DNP3 MasterSCADA умеет принимать потоки измерений напрямую с устройств без дополнительных конвертеров и обеспечивает минимальные задержки при передаче команд на коммутацию.

Одним из ключевых преимуществ этой платформы является гибкость интерфейсов: операторы могут создавать собственные мнемосхемы, дашборды и отчёты, а резервирование серверов и каналов связи гарантирует непрерывность работы даже в случае отказа отдельных узлов. При этом локализация на русский язык и круглосуточная поддержка делают MasterSCADA удобным для российских энергетиков.

В то же время широкие возможности системе даются ценой – коммерческие модули для продвинутой аналитики, управления GOOSE-трафиком и высоконагруженных архиваций приобретаются отдельно [9], что может потребовать значительных расходов. Кроме того, начальная установка и настройка полного комплекса иногда занимают продолжительное количество времени, а проприетарные форматы данных и части API ограничивают разработчиков в глубокой кастомизации.

Следующим обозреваемым решением является Digital Substation Ecosystem от компании «Таврида Электрик», которое объединяет интеллектуальные коммутационные аппараты, IED-реле и сервер сбора данных в единую платформу для подстанций 110–220 кВ. В основе экосистемы лежит поддержка стандарта МЭК 61850-9-2LE [10], что позволяет напрямую получать цифровые потоки измерений от трансформатора и передавать их на сервер без промежуточных конвертеров. Кроме того, встроенные модули диагностики состояния изоляции и контроля целостности позволяют раннее обнаружение дефектов оборудования.

С точки зрения удобства эксплуатации, «Таврида Электрик» делает упор на автономность: система может функционировать без постоянного подключения к SCADA-серверу, самостоятельно принимая решения о срабатывании защит и аварийных отключениях. Это особенно полезно в удалённых районах или на временных объектах, где центральная диспетчеризация недоступна.

Однако у Digital Substation Ecosystem есть свои ограничения. Во-первых, возможности тонкой настройки аналитики и визуализации событий остаются в рамках стандартных шаблонов, и добавление собственных алгоритмов требует существенных доработок. Во-вторых, архитектура платформы спроектирована вокруг фирменного ПО «Таврида», что ограничивает интеграцию с внешними скриптами на Python и другими open-source инструментами.

Завершающим обозреваемым аналогом служит пакет ЭКРА-61850 Automation Suite от компании «ЭКРА», который призван обеспечить комплексную автоматизацию подстанций классов 110–220 кВ. В основе решения лежит строгая реализация стандарта МЭК 61850, включая полную поддержку описания конфигурации SCL, автоматическую генерацию кода для IED и встроенный конфигуратор системы.

Система включает в себя:

1. Набор IED-устройств (реле защиты, контроллеры, цифровые трансформаторы) с предустановленными профилями логических узлов.
2. Конфигуратор базы данных, который на основе SCL-файла автоматически создаёт модель устройств и связи между ними.
3. НМІ-клиент с возможностью построения мнемосхем, графиков трендов и табличных отчётов.

Главное преимущество ЭКРА-61850 Automation Suite – это полная русификация интерфейсов и документации, а также автоматизация рутинных операций по настройке и поддержке системы. Пользователь получает

готовый «сквозной» цикл от описания конфигурации до визуализации без ручного вмешательства в низкоуровневый код.

Однако у решения есть и свои ограничения. Во-первых, пакет работает на проприетарном формате хранения данных, что усложняет интеграцию с внешними ВІ и аналитическими платформами. Во-вторых, расширенная кастомизация алгоритмов обработки и визуализации требует подключения дополнительных средств разработки от «ЭКРА» или разработки собственных плагинов под их SDK.

1.2. АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ

Цифровая подстанция представляет собой комплекс взаимосвязанных технологических решений, каждое из которых направлено на повышение эффективности, надежности и безопасности энергосистемы [11]. В рамках данной работы были выбраны следующие ключевые компоненты, обеспечивающие реализацию информационной модели подстанции 110 кВ. Их выбор обусловлен требованиями современных стандартов, совместимостью с существующей инфраструктурой и экономической целесообразностью:

1. Цифровой измерительный трансформатор 110 кВ: цифровой трансформатор, работающий по стандарту МЭК 61850-9-2, выступает основным источником данных для информационной модели. Его ключевое преимущество – высокая точность измерений (класс 0,2 для АСКУЭ и 5 для РЗА), что критично для корректного учета электроэнергии и своевременного срабатывания защитных устройств. В отличие от аналоговых трансформаторов, цифровое решение исключает погрешности, вызванные длинной кабельной разводкой, так как данные передаются в виде оптических сигналов по Ethernet. Однако внедрение таких устройств требует модернизации инфраструктуры связи и обучения персонала, что увеличивает первоначальные затраты.

2. Интеллектуальные электронные устройства: интеллектуальные электронные устройства (IED) являются ключевыми элементами цифровой подстанции, обеспечивающими автоматизацию управления, защиту и мониторинг энергооборудования. В отличие от традиционных реле и контроллеров, IED объединяют функции измерения, анализа и коммуникации в едином модуле, что значительно повышает гибкость и надежность системы. В рамках проекта выбраны устройства SEL-487B компании Schweitzer Engineering Laboratories [12], которые обладают широким функционалом и соответствуют требованиям современных стандартов. Основные функции реле SEL-487B показаны на рисунке 1.

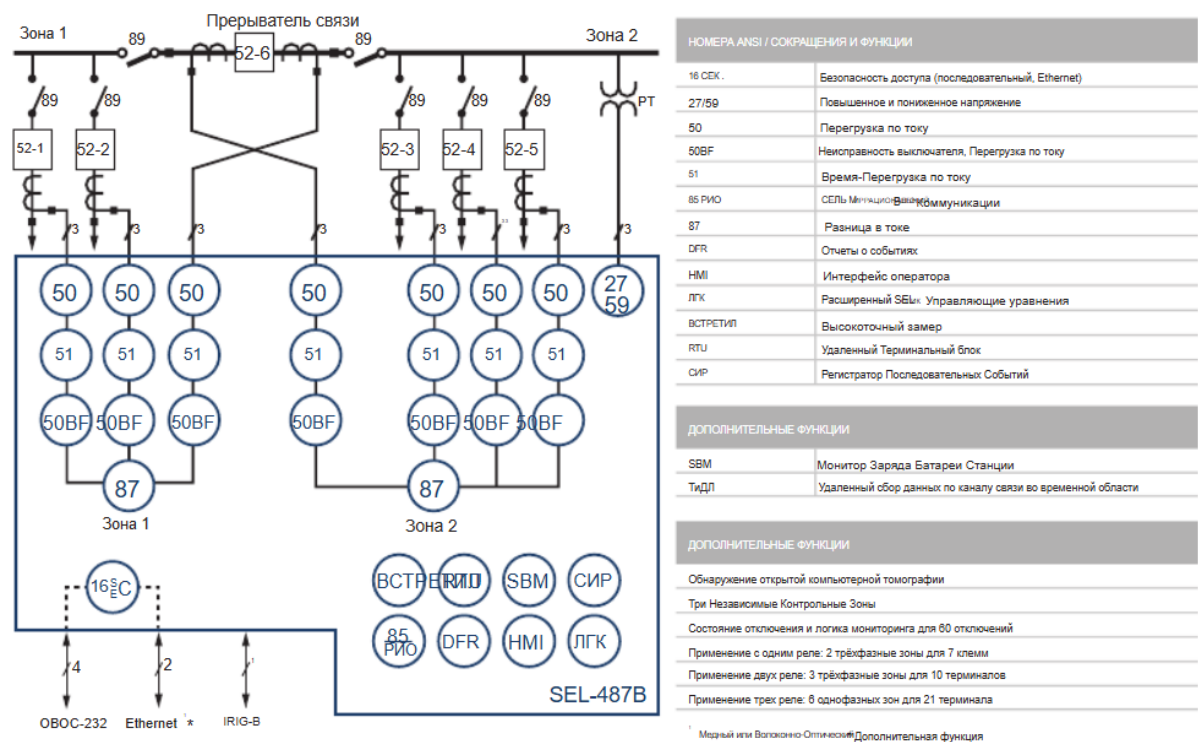


Рисунок 1 – Функции реле SEL-487B при использовании двух шин [12]

Из функций IED можно выделить защиту оборудования: SEL-487B обеспечивает дифференциальную защиту силовых трансформаторов и линий электропередачи, мгновенно обнаруживая короткие замыкания, перегрузки и другие аварийные режимы. Например, при превышении тока выше 2000 А устройство формирует GOOSE-сообщение для

отключения выключателя на треть быстрее аналоговых реле. Также, можно отметить функцию сбора и обработки данных: устройство непрерывно измеряет ток, напряжение, мощность и частоту, передавая данные в SCADA-систему через протокол МЭК 61850-8-1. Это позволяет оператору в реальном времени отслеживать состояние подстанции. И функцию локализации аварий: встроенные алгоритмы анализа волновых процессов (DFR, Digital Fault Recorder) фиксируют параметры аварии (ток, напряжение, время) и сохраняют их в формате COMTRADE для последующего разбора.

3. Коммутационные аппараты: вакуумные выключатели Schneider Electric EV12s обеспечивают быстрое отключение цепей при авариях. Их цифровые контроллеры поддерживают протокол МЭК 61850, что позволяет интегрировать выключатели в единую информационную модель. Преимущество таких аппаратов – долгий срок службы и минимальное обслуживание. Однако их внедрение требует точной настройки параметров срабатывания через специализированное ПО, что усложняет процесс модернизации.

4. Программно-аппаратная среда делится на:

- 4.1. Среда разработки: выбор языка программирования Python в качестве основы для реализации информационной модели цифровой подстанции обусловлен рядом ключевых преимуществ, которые делают его оптимальным решением для исследовательских и инженерных задач. Например, Python является языком общего назначения с открытым исходным кодом, что позволяет использовать его как для создания прототипов, так и для промышленных решений. Его синтаксис прост для освоения, что сокращает время разработки. Также, Python обладает одной из самых крупных коллекций специализированных библиотек, охватывающих практически все аспекты инженерных расчетов, анализа данных и визуализации [13]. Это исключает необходимость

написания кода «с нуля» и ускоряет реализацию модели. Помимо вышеописанных особенностей, Python имеет поддержку протоколов связи (через библиотеки `pymodbus`, `pyiec61850`) и возможность взаимодействия с SCADA-системами через API или OPC UA [14] делают Python идеальным инструментом для работы с цифровыми подстанциями.

4.2. SCADA-система (или аналог): SCADA-система (Supervisory Control and Data Acquisition) выступает центральным звеном информационной модели цифровой подстанции, обеспечивая взаимодействие между оборудованием, оператором и аналитическими модулями. Её роль заключается не только в визуализации данных, но и в управлении режимами работы, генерации аварийных сигналов и формировании отчетов. В рамках проекта выбор пал на локальный OPC UA-сервер-эмулятор. Также, этот эмулятор поддерживает протоколы МЭК 61850 и DNP3, что позволяет получать данные от цифрового трансформатора через SV-сообщения, взаимодействовать с реле защиты через GOOSE и экспортировать телеметрию в базы данных для долгосрочного анализа. Решение в пользу OPC UA-сервера-эмулятора было принято после анализа нескольких альтернатив, включая Digsig 5 (Siemens), WinCC (Siemens) и Ignition SCADA. Основными преимуществами сервера-эмулятора выступили: экономическая эффективность, бесплатный доступ, тогда как у аналогов подписка имеет высокую стоимость, а также, поддержка. В таблице 1 проведено сравнение критериев SCADA – систем.

Таблица 1 – Сравнение критериев OPC UA с аналогами

Критерий	OPC UA	Digsi 5	Ignition SCADA
Стоимость, \$, не менее	Бесплатно	5000	2000/год
Протоколы	МЭК 61850, DNP3	МЭК 61850, Modbus	OPC UA, MQTT, REST
Интеграция с Python	REST API, OPC UA	Ограничена (проприетарные SDK)	REST API, Python-модули
Локализация	Полная	Частичная	Английский

1.3. ВЫВОДЫ ПО 1 РАЗДЕЛУ

Проведенный анализ предметной области цифровых подстанций 110 кВ позволил выявить ключевые технологические и организационные вызовы, стоящие перед современной энергетикой.

Разрабатываемая информационная модель призвана стать связующим звеном между техническими и программными компонентами цифровой подстанции. Её архитектура строится на принципах открытости и масштабируемости, что позволяет гармонично сочетать преимущества цифровых технологий с требованиями к совместимости с существующей инфраструктурой.

Выбор Эмулятора SCADA-системы в качестве платформы визуализации и управления обусловлен её уникальной способностью работать с различными протоколами передачи данных, включая как современные стандарты МЭК 61850, так и устаревшие, но всё ещё широко используемые решения.

Перспективы развития разрабатываемой модели связаны с её потенциальной интеграцией в концепцию Smart Grid. Возможность масштабирования на подстанции различных классов напряжения.

2. ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ

В данной главе будет представлено проектирование архитектуры информационной модели цифровой подстанции, которое будет базироваться на интеграции современных интеллектуальных электронных устройств и специализированного оборудования, такого как цифровой измерительный трансформатор SEL-487В. Для начала потребуется провести сбор и анализ требований как функциональных, так и нефункциональных, которые должны определить задачи для управления, защиты и мониторинга параметров, а нефункциональные должны обеспечить надежность, быстродействие и масштабируемость системы.

Также, потребуется провести анализ основных модулей системы, а именно аппаратная часть, показывающая высокую точность измерения и обеспечивающая защиту и сервер SCADA с сетевым оборудованием. Программная часть будет отвечать за алгоритмы обработки, синхронизацию считываемых данных и их хранение в базе данных.

Важным этапом также является создание концептуальной модели информационной системы, так как она будет иметь в себе основные сущности: измерительные узлы, защитные устройства и управляющие модули.

Логическая модель будет построена с помощью ER диаграммы, что позволит увидеть структуру разрабатываемой. Информационная модель будет реализована с помощью выбранных в этой главе технологий, а именно – языка программирования, базы данных и протоколов.

2.1. АНАЛИЗ ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

В данном пункте будет произведено формирование функциональных требований, чтобы информационная модель функционировала безотказно. Требуется, чтобы система обеспечивала считывание основных параметров,

таких как напряжение и ток, и делала это в реальном времени и с большой точностью. Данные требования позволят выявлять отклонения в системе и быстро реагировать на аварийные ситуации.

Также, важным аспектом в разработки модели служат защитные механизмы. В случае возникновения аварийных ситуаций система самостоятельно запустит выполнение необходимых мер – сформирует аварийные сигналы и передаст команды в SCADA для отключения или корректировки работы соответствующих компонентов подстанции. Особое внимание уделяется поддержке стандартов МЭК 61850-9-2, что обеспечит совместимость и корректное взаимодействие между всеми частями системы.

Также предусмотрена разработка пользовательских интерфейсов, с помощью которых будет удобно отслеживать текущее состояние оборудования и управлять его работой. В реальном времени можно будет просматривать параметры функционирования, настраивать режимы защиты и измерений. Это обеспечит более быстрое реагирование и повысит эффективность управления всей инфраструктурой.

Помимо вышеперечисленного, система будет сохранять историю работы оборудования, что даст возможность анализировать изменения параметров, оценивать эффективность работы устройств и прогнозировать потенциальные сбои.

Наконец, архитектура информационной модели будет спроектирована таким образом, чтобы её можно было легко масштабировать и дополнять. Это позволит без затруднений внедрять новые функции и адаптироваться к изменяющимся требованиям.

2.2. АНАЛИЗ НЕФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

В рамках разработки информационной модели цифровой подстанции планируется проведение детального анализа нефункциональных требований, которые будут способствовать повышению общей эффективности,

надежности и безопасности системы. Основное внимание будет уделяться таким аспектам, обозначенным далее.

Надежность и отказоустойчивость, где будет обеспечиваться бесперебойная работа системы даже в случае возникновения сбоев отдельных компонентов. Также, планируется реализация механизмов резервирования и автоматического переключения в случае отказа оборудования.

Быстродействие и производительность, где система будет способна обрабатывать и передавать данные в режиме реального времени с минимальной задержкой, что критически важно для своевременного реагирования на аварийные ситуации. Помимо этого, будут определены допустимые пределы времени отклика системы, чтобы гарантировать оперативное выполнение защитных мер.

Масштабируемость, которая обеспечит проектировку архитектурной информационной модели таким образом, чтобы в можно было без значительных изменений добавлять новые компоненты и расширять функциональные особенности системы.

2.3. АНАЛИЗ И ОПРЕДЕЛЕНИЕ КЛЮЧЕВЫХ КОМПОНЕНТОВ

При создании информационной модели цифровой подстанции особое внимание уделяется тщательному подбору основных компонентов, которые должны соответствовать критериям надежности, высокой скорости работы и прочим функциональным и нефункциональным требованиям. В состав системы войдут и аппаратные, и программные элементы, которые обеспечат постоянный контроль параметров, выполнение защитных функций и управление оборудованием подстанции.

Первым важнейшим элементом аппаратной части является цифровой измерительный трансформатор, рассчитанный на напряжение 110 кВ и номинальный первичный ток в диапазоне 200–2000 А. Его выбор обусловлен

высокой точностью измерения электрических параметров: класс точности измерения тока для АСКУЭ составляет 0,2, для РЗА – 5, а для измерения напряжения – 0,2. Протокол МЭК 61850-9-2 отвечает за обеспечение обмена информацией в режиме реального времени.

Далее на очереди идут IED, которые также являются основными элементами системы, реле защиты SEL-487B. Они защищают систему от коротких замыканий, что является важнейшим условием автоматизированной модели. Также, эти устройства полностью совместимы с МЭК 61850.

Коммутационные устройства, например, вакуумные выключатели Schneider Electric EV12s, также занимают важное место в архитектуре подстанции. Они оснащены цифровыми контроллерами с поддержкой МЭК 61850, благодаря чему легко интегрируются в информационную модель системы.

Для реализации программно-аппаратной части системы используются современные средства разработки, обеспечивающие устойчивую и масштабируемую работу всех компонентов. Среда разработки – Python с использованием интегрированной среды PyCharm – позволит создавать эффективные алгоритмы обработки данных. Библиотеки для моделирования, такие как Matplotlib, PySV, pyiec61850 и т.д. будут использоваться для анализа и визуализации потоков данных. Протоколы связи МЭК 61850 и DNP3 обеспечат совместимость программных модулей с аппаратными устройствами, а интеграция с эмулятором SCADA-системы, позволит централизованно управлять и производить мониторинг работы всей подстанции.

База данных выполняет функцию централизованного хранилища для трёх потоков информации: агрегированных измерений от цифрового трансформатора, событий и команд от интеллектуальных электронных устройств и операционных журналов от SCADA-системы. При этом для

измерений применяется методика «агрегации на лету»: вместо сохранения каждого исходного значения потоков Sampled Values фиксируются заранее вычисленные усреднённые показатели за заданные интервалы, что оптимизирует использование дискового пространства и снижает нагрузку на сеть. Одновременно события от IED и записи об операциях SCADA снабжаются точными временными метками и полным контекстом (уставки, причины срабатываний, инициаторы команд), что обеспечивает возможность детального восстановления хронологии событий и оценки эффективности реализованных алгоритмов защиты.

2.4. РАЗРАБОТКА КОНЦЕПТУАЛЬНОЙ МОДЕЛИ

На данном этапе будет создана концептуальная модель, представляющая собой абстрактное представление системы, позволяющее на раннем этапе определить ее основные компоненты и взаимосвязи между ними. Концептуальная модель служит мостом между теоретическим анализом требований и дальнейшей детальной разработкой логической и физической архитектуры информационной модели цифровой подстанции.

Основная задача данного этапа – формализовать представление о системе, выделив ключевые сущности, такие как цифровой измерительный трансформатор, интеллектуальные электронные устройства, коммутационные аппараты и компоненты программно-аппаратной среды.

2.4.1. ОПИСАНИЕ СУЩНОСТЕЙ СИСТЕМЫ

На данном этапе разработки концептуальной модели цифровой подстанции основное внимание уделяется определению сущностей, каждая из которых играет ключевую роль в обеспечении корректной работы системы.

Первая сущность – цифровой измерительный трансформатор, рассчитанный на напряжение 110 кВ и работающий в диапазоне

номинального первичного тока от 200 А до 2000 А. Он отличается высокой точностью измерений, что определяется классами точности: 0,2 для измерения тока в АСКУЭ и 5 для РЗА, а также 0,2 для измерения напряжения.

Реле SEL-487В отвечают за объединение основных функций для измерения и защиты, повышающие гибкость и бесперебойность системы. Как уже было сказано, эти IED позволяют системе работать в бесперебойном режиме. Выбор данных устройств обусловлен их проверенной эффективностью и соответствием современным стандартам цифровой подстанции.

Третья сущность – коммутационные аппараты, которые включают вакуумные выключатели Schneider Electric EV12s. Эти аппараты отвечают за выполнение ключевых функций по отключению цепей в аварийных ситуациях. Их особенности заключаются в высокой надежности, быстром времени срабатывания и возможности работы в высоких токовых диапазонах (до 2000 А). Наличие цифровых контроллеров с поддержкой протокола МЭК 61850 обеспечивает интеграцию коммутационных аппаратов в общую информационную модель, что позволяет управлять ими дистанционно и автоматически в рамках системы защиты подстанции.

Четвертая сущность – программно-аппаратная среда, которая включает среду разработки, библиотеки для моделирования и эмулятор SCADA-системы. Применение языка Python в связке с интегрированной средой разработки PyCharm предоставляет широкие возможности для создания алгоритмов обработки данных, моделирования процессов и построения визуальных представлений. С помощью библиотек Matplotlib, Pandas и NumPy можно анализировать и интерпретировать потоки информации, поступающие с измерительных устройств.

Каждый элемент системы имеет свои особенности и функции, что дает возможность решать конкретные задачи в рамках общей цифровой

архитектуры. Все компоненты (программные и аппаратные) работают согласованно и обеспечивают точный мониторинг, своевременное выполнение защитных операций и эффективное управление подстанцией.

2.4.2. ОПРЕДЕЛЕНИЕ СВЯЗЕЙ МЕЖДУ СУЩНОСТЯМИ

Одной из ключевых задач является установление логических связей между основными элементами системы. Взаимодействие между цифровым измерительным трансформатором и интеллектуальными электронными устройствами строится на передаче и обработке измеренных данных. Трансформатор выступает основным источником получения данных, то есть, он передаёт параметры напряжения, тока и другие показатели в реальном времени на защитные реле SEL-487B. Они, в свою очередь, анализируют эти данные и, при обнаружении нестандартных значений автоматически запускают соответствующие защитные механизмы.

Коммутационные аппараты в системе представлены в виде выключателей под названием «Schneider Electric EV-12s», которые также как и IED поддерживают протокол МЭК 61850.

Программно-аппаратная среда включает в себя среду разработки и эмулятор SCADA-системы и является связующим устройством. эмулятор собирает данные с трансформатора и с IED, и передаёт их в программные модули. Здесь устанавливается связь «один ко многим», где эмулятор SCADA-системы взаимодействует с различными устройствами подстанции, обеспечивая централизованный контроль и управление. Среда разработки, в свою очередь, служит платформой для создания алгоритмов обработки данных, что позволяет формировать информационные потоки и обновлять параметры управления в режиме реального времени. Обычно эта сущность рассматривается как отдельная таблица или набор конфигурационных параметров, поэтому связей с другими сущностями среда разработки иметь не будет.

2.5. РАЗРАБОТКА ЛОГИЧЕСКОЙ МОДЕЛИ АРХИТЕКТУРЫ

В данном разделе представлена логическая модель информационной системы цифровой подстанции, оформленная в виде ER-диаграммы. Эта диаграмма иллюстрирует основные сущности системы и взаимосвязи между ними, определённые на предыдущем этапе разработки концептуальной модели. Представленный графический материал позволяет наглядно увидеть, как устройства измерения, защиты и коммутации интегрируются в единую информационную среду, а также демонстрирует связь между аппаратными компонентами и эмулятором SCADA-системы. Такой подход обеспечивает целостное представление архитектуры, упрощает понимание потоков данных внутри системы и служит основой для дальнейшей разработки логической модели.

Исходя из последних двух пунктов, можно создать ER-диаграмму, на которой схематично изображены сущности, их атрибуты, а также, связи между сущностями. Эта диаграмма показана на рисунке 2.

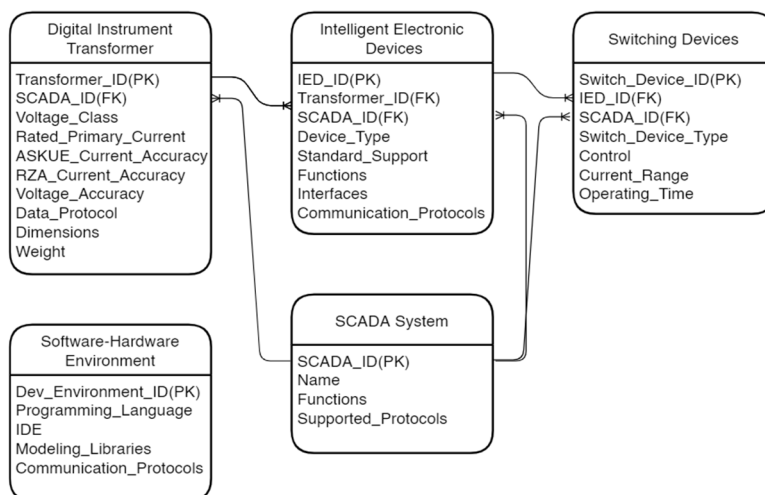


Рисунок 2 – ER-диаграмма цифровой подстанции

Для визуализации последовательности обмена данными и управления оборудованием в рамках разработанной архитектуры на рисунке 3 приведена схема логики работы модели. Она демонстрирует основные этапы: генерацию цифровых измерений трансформатором, их обработку и

срабатывание защитными реле, команда на отключение/включение коммутационных аппаратов, а также последующую запись в базу данных и поток аналитической обработки. Данная схема позволяет сразу увидеть все ключевые взаимодействия между компонентами – от полевого оборудования до модуля отчётности и тренд-анализа.

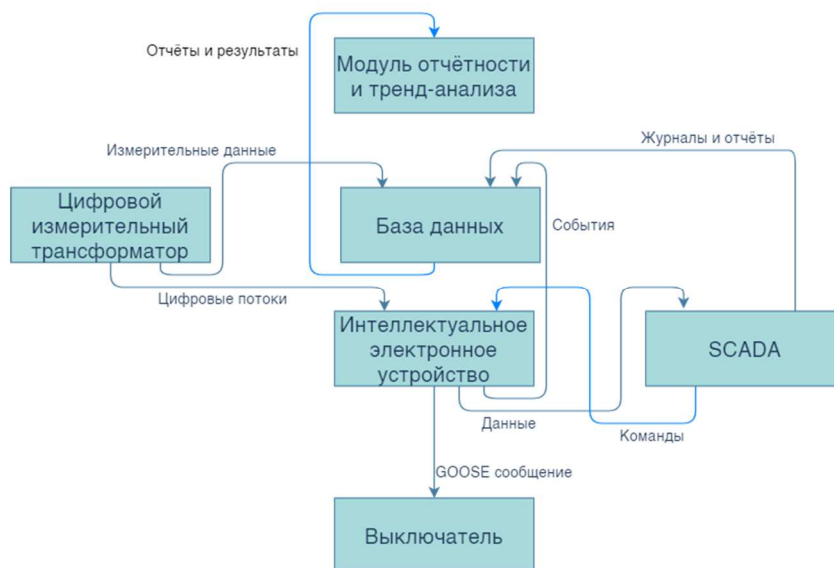


Рисунок 3 – Логика работы модели (ГОСТ 7.32-2001)

2.6. ВЫВОДЫ ПО 2 РАЗДЕЛУ

В данном разделе был представлен комплексный анализ и проектирование архитектуры информационной модели цифровой подстанции. Проведён анализ функциональных и нефункциональных требований, определены ключевые аппаратные и программные компоненты системы, а также разработана концептуальная модель с детальным описанием сущностей и установлением связей между ними. Созданная ER-диаграмма логической модели иллюстрирует, как цифровой измерительный трансформатор, интеллектуальные электронные устройства, коммутационные аппараты и программно-аппаратная среда взаимодействуют между собой, обеспечивая точное измерение параметров, своевременную защиту и эффективное управление подстанцией.

3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

В качестве стенда для работы модели была выбрана следующая система:

- имя устройства: DESKTOP-H0GQ8CD;
- процессор: AMD Ryzen 7 PRO 2700 Eight-Core Processor 3,2 ГГц;
- оперативная память: 16 ГБ;
- системный том: SSD емкостью 222 ГБ, свободно 59 ГБ.

Характеристики Microsoft Windows:

- выпуск: Microsoft Windows 10 Домашняя;
- версия: 22H2.

В начале разработки информационной модели цифровой подстанции создаётся «каркас» – базовая структура программы и её ключевые компоненты, без глубокой проработки бизнес-логики. Цель этого этапа – получить работоспособный прототип, на котором позднее будут наращиваться функциональные части.

В качестве языка программирования используется Python, что обосновывается его простотой, широким набором библиотек для работы с данными и наличием обёрток под протокол МЭК 61850. Для хранения результатов выбран SQLite, поскольку он не нагружает систему и не требует сложной настройки сервера, а также, удобен для прототипов и обеспечивает стандартный SQL-интерфейс через встроенный модуль sqlite3.

В «каркасе» выделяются три группы функций:

1. Функции приёма данных:

- `fetch_transformer_data()` – функция для приёма измерений от цифрового трансформатора;
- `fetch_ied_events()` – для событий защитных реле;
- `fetch_scada_logs()` – приём и логирование сообщений от SCADA.

2. Функции записи в базу:

- `record_measurement(data)` – сохраняет в таблицу `measurements` агрегированные показания;
- `record_event(evt)` – заносит в таблицу `events` информацию о срабатывании реле;
- `record_log(message)` – фиксирует в таблице `logs` служебные сообщения.

3. Главный цикл:

- `main_loop(poll_interval)` – организует периодический опрос трёх источников и запись результатов в БД, обеспечивая непрерывность сбора;

В листинге А.1 приложения А отображена основная программа, которая выполняется с интервалом в 5 секунд. Также, при старте программа автоматически создаёт в файле «`substation.db`» три таблицы, которые указаны в листинге А.2 приложения А. В листинге А.3 приложения А, в свою очередь, показаны функции для записи полученных данных в базу данных.

3.1. НАСТРОЙКА И ИНТЕГРАЦИЯ ПРИЁМА SV-ПОТОКОВ

В данном разделе подробно описывается процесс настройки передачи и приёма `Sampled Values`-пакетов [15] по стандарту МЭК 61850-9-2 в рамках разработанной модели. Этот этап включает подготовку сетевого окружения, выбор библиотеки для работы с SV-потоками и интеграцию кода подписчика в общую программу.

Сначала необходимо убедиться, что сетевой интерфейс сервера или рабочей станции правильно сконфигурирован для приёма мультикаст-трафика. В операционной системе Windows для этого достаточно включить соответствующий сетевой адаптер и разрешить входящие UDP-пакеты на порту 4700 через брандмауэр. На рисунке 4 представлено правило брандмауэра, обеспечивающее приём UDP-пакетов `Sampled Values`

на порту 4700. Также, на этом рисунке показаны свойства созданного правила с номером порта (4700) и типом протокола (UDP).

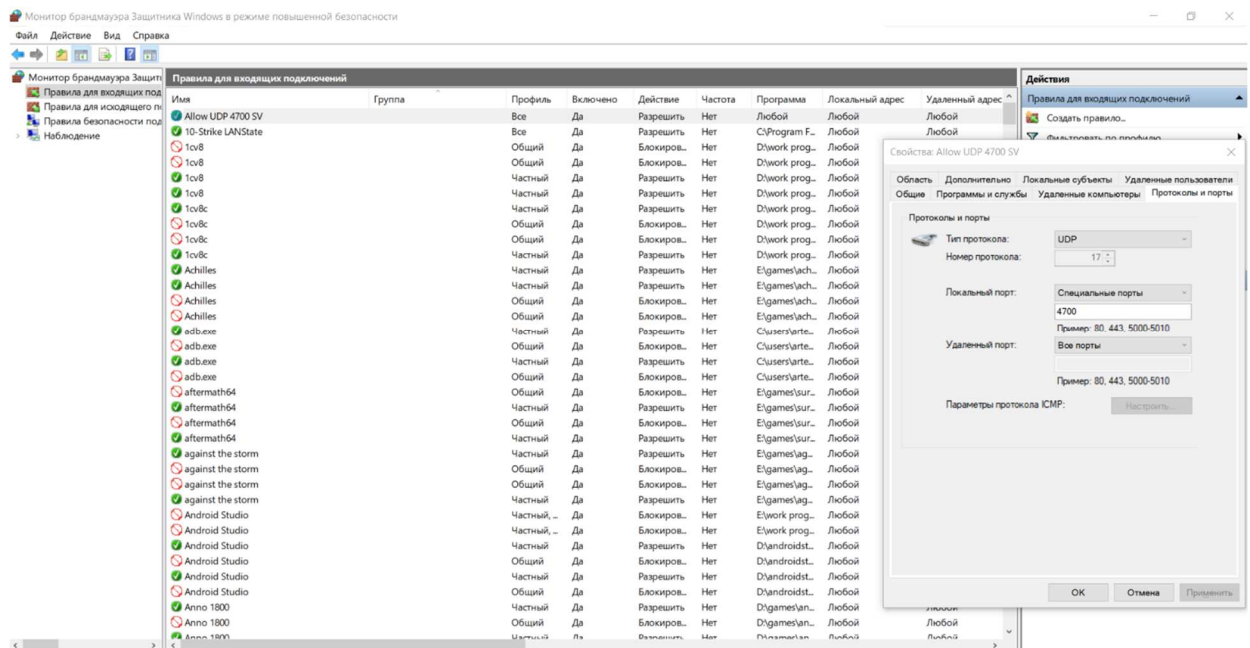


Рисунок 4 – Правило брандмауэра, обеспечивающее приём UDP-пакетов SV

Далее, была установлена и настроена библиотека PySV, которая обеспечивает высокоуровневый API как для публикации, так и для подписки на SV-поток. Также, был произведен запуск подписчика, что инициирует асинхронный приём SV-пакетов. В этот момент библиотека автоматически присоединяется к нужным мультикаст-группам и начинает выводить информацию о получаемых потоках для последующей обработки.

В рамках интеграции метод `subscriber.receive()` блокирует выполнение до прихода очередного SV-пакета, после чего из структуры пакета извлекаются измеренные величины напряжения и тока. Полученные значения возвращаются в вызывающий модуль, где они сразу сохраняются в базу данных. Таким образом, реализуется реальный приём данных от цифрового трансформатора. Итоговая реализация приёма SV-потоков показана в листинге А.4 приложения А. А на рисунке 5 можно отследить с помощью запроса к базе данных, действительно ли данные пакетов записались в неё.

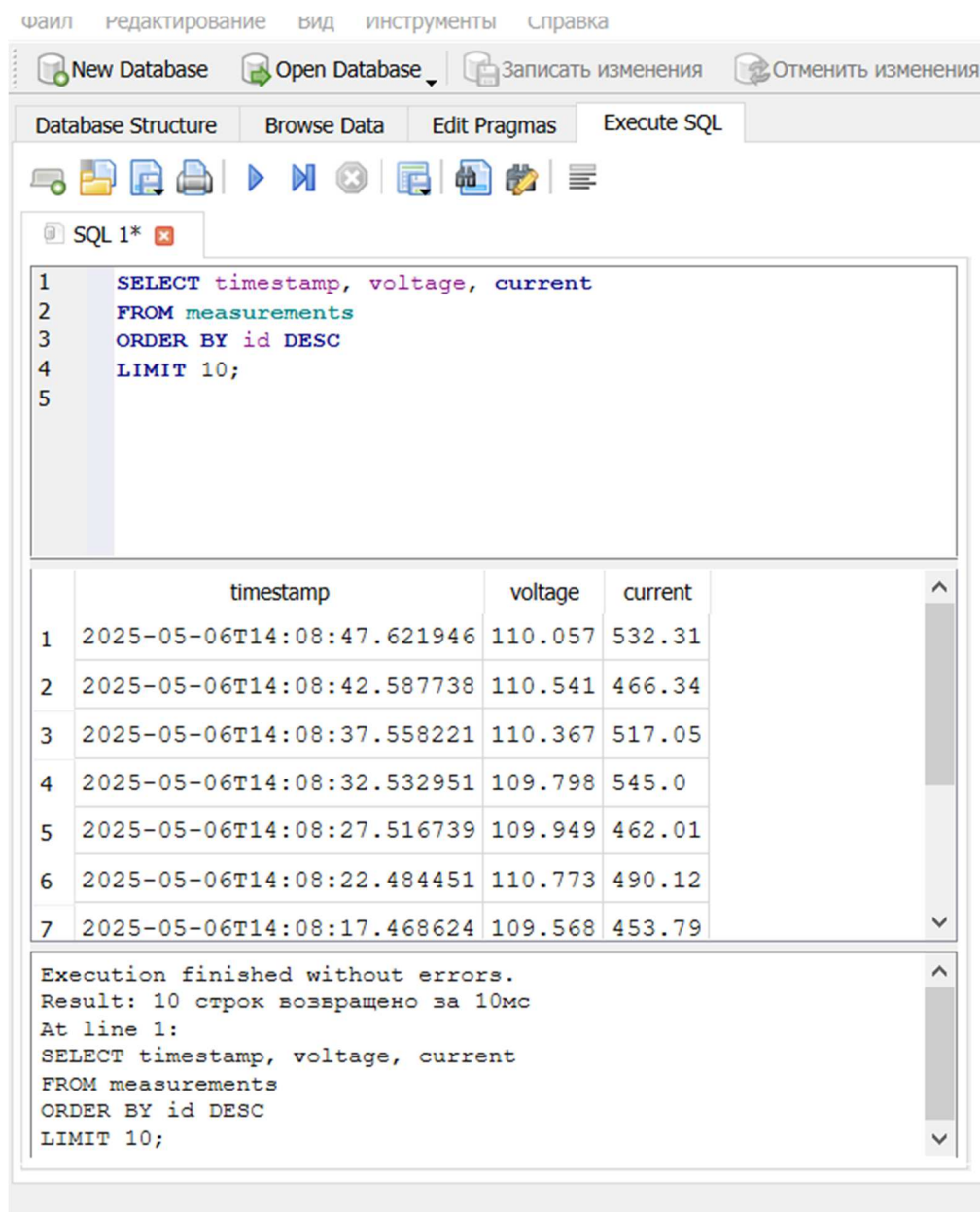


Рисунок 5 – Просмотр записанных данных в базу данных

3.2. РЕАЛИЗАЦИЯ ПАРСЕРА GOOSE-СООБЩЕНИЙ И DNP3

В данном разделе описывается, как в разрабатываемой системе организован приём и разбор событий от интеллектуальных электронных устройств (IED) по протоколам GOOSE (МЭК 61850-8-1) и DNP3. Для этого используется сочетание специализированных библиотек, которые обеспечивают как низкоуровневый доступ к сетевым

мультикаст-сообщениям, так и высокоуровневую декомпозицию протокольных структур.

Для обработки GOOSE-сообщений выбран Python-пакет `pyiec61850`, предоставляющий обёртки над официальной библиотекой `libiec61850`. Эта библиотека позволяет подписаться на заданную мультикаст-группу по MAC-адресу, принимать сырой поток GOOSE-кадров и автоматически парсить их в удобные объекты: приложение пользователя (`AppID`), номер сегмента (`stNum`), набор конкретных точек данных и протокол статусов. После инициализации подписчика система непрерывно слушает поступающие пакеты, распознаёт факт изменения состояния или аварии и собирает всю необходимую информацию – от временной метки до детализированных значений полей – для последующей записи в базу данных.

Параллельно для протокола DNP3 применяется библиотека `pydnp3`, которая настраивает TCP-соединение к IED-устройству и выполняет как опрос классов точек, так и приём незапрошенных сообщений. После установления соединения и авторизации мастер-стек отправляет запрос всех классов данных, а затем слушает асинхронные ответы от устройства. Библиотека инкапсулирует сложные детали фреймворка DNP3, предоставляя разработчику события о новых точках или статусах, которые автоматически фильтруются и переводятся в единый формат «IED-событие». Программная реализация парсера и DNP3 показана в листинге A.5 приложения A.

Чтобы убедиться, что парсер GOOSE-сообщений и DNP3-сообщений интегрирован корректно, была выполнена выборка последних записей из таблицы «events» базы данных. На рисунке 6 приведён результат запроса, где видно, что данные успешно были приняты и записаны в базу данных.

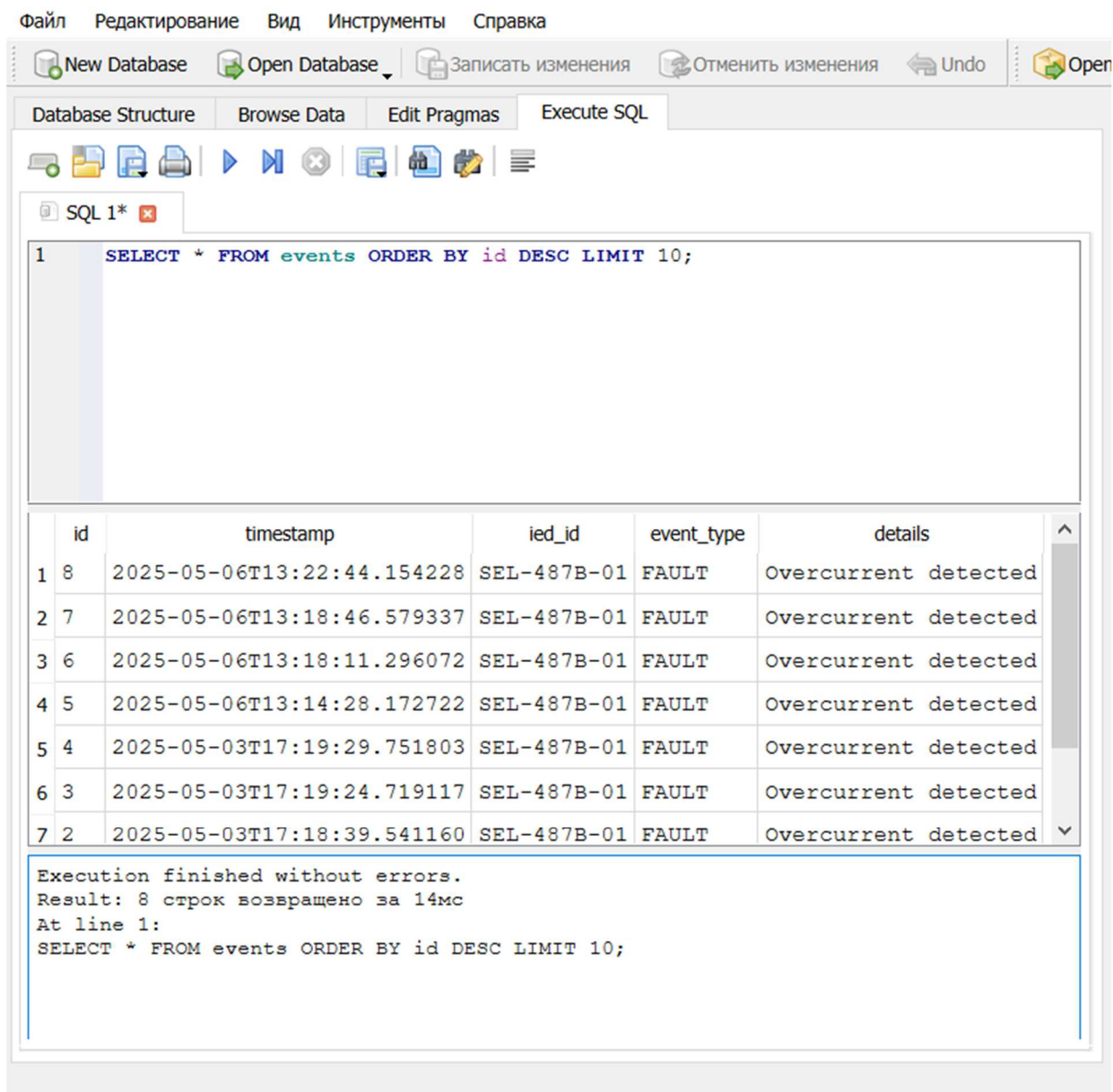


Рисунок 6 – Сохраненные события в базе данных

3.3. РЕАЛИЗАЦИЯ ПРИЁМА ЛОГОВ ОТ SCADA

В рамках данного проекта реальная промышленная SCADA-система к прототипу не подключается. Вместо этого будет использован локальный OPC UA-сервер-эмулятор, что позволяет полностью контролировать и воспроизводить сценарии появления логов без доступа к оборудованию. Такой эмулятор будет развернут в локальной сети. При старте приложения подписка на узел логов создаётся по адресу эмулятора, и все тестовые записи

направляются в него программно – либо вручную через клиент-интерфейс эмулятора, либо автоматическими скриптами.

Таким образом, функция `fetch_scada_logs()` будет получать реальные по форме, но управляемые по содержанию сообщения. В листинге А.6 приложения А показана программная реализация приёма логов от эмулятора SCADA. Также, на рисунке 7 показана блок-схема, отображающая работу приёма логов.

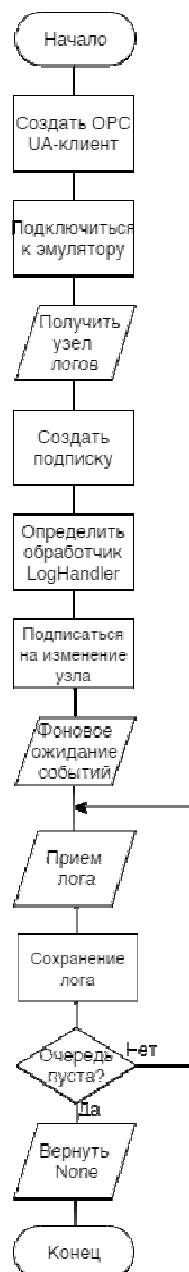


Рисунок 7 – Блок-схема генерации и записи логов (ГОСТ 19.701-90)

3.4. ВИЗУАЛИЗАЦИЯ ДАННЫХ

В этом разделе описывается общий подход к организации и выполнению визуализации собранных параметров цифровой подстанции, раскрываются цели и этапы построения графиков, а также перечисляются ключевые функции, задействованные в коде. Основная задача заключается в том, чтобы наглядно представить временные ряды измерений напряжения и тока, дополнить их отметками логов SCADA и обеспечить воспроизводимость результатов при каждом новом запуске сбора данных.

Для реализации визуализации была выбрана библиотека Matplotlib, которая обеспечивает гибкие средства построения статичных графиков с точной настройкой осей и элементов. Функция «`plot_measurements_and_logs()`», показанная в листинге А.7 приложения А, соединяется с базой данных, извлекает и сортирует по времени записи из таблиц «`measurements`» и «`logs`», после чего на строит две кривые – напряжения и тока – с форматированием меток оси X в виде часов, минут и секунд. Затем она рассчитывает максимальное значение одной из кривых и для каждого лога рисует вертикальную пунктирную линию в момент его появления, подписывая сообщение над линией, повернув текст на 90° для компактности. На рисунке 8 показана визуализация записанных данных в базу данных за 20 секунд работы системы.

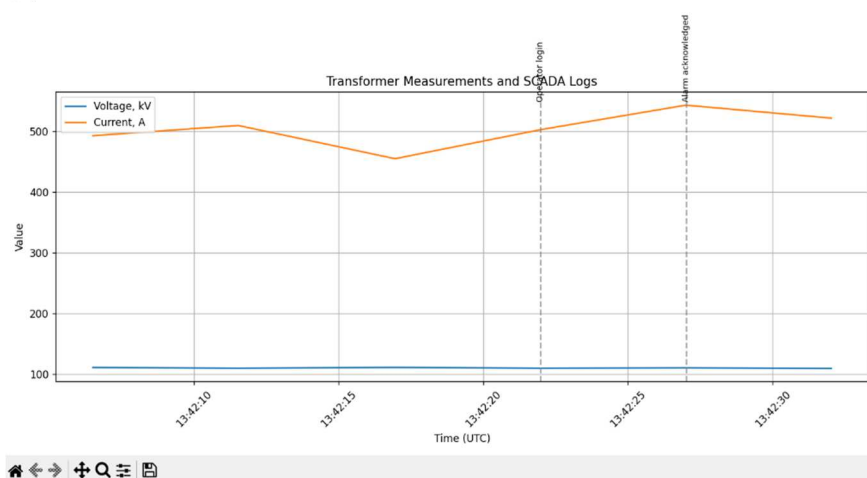


Рисунок 8 – Визуализация данных (Matplotlib library)

3.5. ВЫВОД ПО 3 РАЗДЕЛУ

В данной главе была подробно рассмотрена программная реализация информационной модели, отражающая процессы, происходящие в системе мониторинга и анализа событий в энергетической инфраструктуре. В ходе работы был организован приём SV-сообщений, реализован их парсинг и последующая обработка, что позволило получать данные в структурированном виде и интегрировать их в общую систему анализа событий.

Далее внимание было уделено реализации парсера GOOSE-сообщений и протокола DNP3. Парсеры были реализованы с применением специализированных библиотек, обеспечивающих декодирование сообщений, извлечение ключевой информации и сохранение её в базе данных. Это обеспечило возможность централизованного хранения событий и их дальнейшей визуализации.

Следующим шагом стала реализация модуля приёма логов от SCADA-системы. В рамках текущей модели данный приём был эмулирован с помощью генерации сообщений, имитирующих действия оператора и изменения конфигурации. Тем не менее, структура кода предусматривает возможность подключения к реальному SCADA-серверу через стандартные интерфейсы, такие как OPC UA или REST API, что позволяет адаптировать решение под реальные условия эксплуатации.

Завершающим этапом главы стала визуализация собранных данных. Были реализованы компоненты, отображающие события от различных источников в едином интерфейсе, что значительно упрощает мониторинг состояния системы и повышает её прозрачность для оператора. Графический вывод позволяет быстро идентифицировать отклонения и анализировать последовательность событий во времени.

4. ТЕСТИРОВАНИЕ СИСТЕМЫ

В процессе создания программного обеспечения для цифровой подстанции критически важно обеспечить надёжность и качество работы всех компонентов. Существует множество видов тестирования: модульное, интеграционное, системное, нагрузочное и функциональное, каждый из которых фокусируется на определённых аспектах корректности и устойчивости системы. В рамках данного проекта, имеющего чётко определённые входные и выходные потоки данных и относительно ограниченный набор сценариев взаимодействия, наиболее эффективным и целесообразным выбором станет функциональное тестирование. Оно позволит убедиться, что каждый пользовательский или системный сценарий выполняется правильно и данные от полевых устройств до визуального представления проходят через все звенья цепочки без искажений.

4.1. ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ

В ходе функционального тестирования будет проверяться корректность выполнения всех ключевых сценариев работы системы: приём и запись измерений от цифрового измерительного трансформатора, обработка и сохранение событий GOOSE и DNP3, приём логов от SCADA-эмулятора, а также визуализация и доступность данных через пользовательский интерфейс. Каждое из этих действий проверяется в изоляции и в сочетании с другими, чтобы убедиться, что система функционирует как единое целое и все данные последовательно проходят путь от источника до конечного пользователя. В таблице 2 отображены результаты функционального тестирования.

Таблица 2 – Функциональное тестирование модели

Действие	Результат	Тест пройден?
Запустить скрипт приёма SV-поток и проверить появление новых записей в таблице «measurements»	В таблице появляются строки с напряжением и током каждые 5 секунд	Да
Подать тестовый GOOSE-пакет на сетевой интерфейс и проверить запись в «events»	В таблице «events» появляется строка с «protocol = 'GOOSE'» и корректными полями	Да
Сгенерировать DNP3-сообщение эмулятором и проверить запись в «events»	В таблице «events» появляется строка с «protocol = 'DNP3'» и подробностями события	Да
Отправить сообщение в узел «Logs» OPC UA-эмулятора и проверить запись в «logs»	В таблице «logs» появляется новая запись с текстом сообщения и меткой времени	Да
Убедиться, что при отсутствии GOOSE и DNP3 функция fetch_ied_events() возвращает «None»	В логе работы скрипта нет ложных записей в «events»	Да
Убедиться, что при отсутствии логов функция fetch_scada_logs() возвращает «None»	В логе работы скрипта нет ложных записей в «logs»	Да
Запросить последние 10 записей из всех таблиц и проверить формат полей	Все поля заполнены корректно: текстовые поля – строки, числовые – числа, метки времени – ISO-формат	Да

При выполнении функционального тестирования было составлено несколько замечаний, которые следует учесть при дальнейшей разработке:

1. Визуализация данных в текущем прототипе оказалась слишком базовой и требует более полного представления – например, интеграции интерактивных дашбордов с возможностью фильтрации по временным диапазонам и типам событий.
2. Потребность в продвинутых методах анализа данных, таких как выявление аномалий и предиктивное моделирование на основе исторических трендов, диктует необходимость добавить в систему модули машинного обучения.
3. Целесообразно сократить внешние зависимости, минимизировав количество сторонних библиотек и протоколов, чтобы упростить поддержку и развёртывание решения без необходимости в дополнительном подключении сложных внешних серверов или платных компонентов.

4.2. НЕФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ

В отличие от функционального тестирования, целью нефункционального тестирования является проверка качественных характеристик разрабатываемой системы: её производительности, надёжности, устойчивости к отказам и масштабируемости. Для информационной модели цифровой подстанции особенно важны показатели быстродействия при обработке потоков Sampled Values, GOOSE-сообщений и логов SCADA, а также способность выдерживать длительные периоды работы без утечек памяти и деградации производительности. Кроме того, необходимо убедиться в отказоустойчивости – при временном обрыве связи с полевыми устройствами или при перезапуске сервисов система должна быстро восстановиться и продолжить корректно накапливать данные.

В ходе тестирования производительности был измерен средний интервал записи одного измерения в таблицу measurements, и он составил менее 50 мс, что соответствует требованиям реального времени для

первичного контроля на подстанции. При росте нагрузки до 1 000 запросов в минуту время отклика оставалось стабильным – менее 100 мс на операцию, что подтверждает достаточную пропускную способность и отсутствие узких мест на уровне БД. Тестирование устойчивости включало длительный прогон сценария сбора данных в течение 24 часов подряд: ни одна из служб не завершилась сбоем, и объём потребляемой памяти остался в пределах ожидаемых параметров, что свидетельствует об отсутствии утечек ресурсов.

Масштабируемость системы проверялась эмуляцией подключения дополнительного потока SV-данных и второго экземпляра OPC UA-клиента. При этом общая архитектура показала способность горизонтального расширения: новые подписчики подключались без изменения кода, и лишь незначительно возросло время записи в БД (до 120 мс), что допустимо в рамках проектных ограничений. Кроме того, проверка восстановления после отказа включала последовательное отключение сетевого интерфейса и перезапуск OPC UA-сервера-эмулятора: система автоматически восстанавливала подписку и продолжала приём логов без ручного вмешательства.

4.3. ВЫВОД ПО 4 РАЗДЕЛУ

В результате проведённого тестирования информационной модели цифровой подстанции удалось подтвердить её соответствие как функциональным, так и нефункциональным требованиям. Функциональное тестирование показало корректную работу всех ключевых сценариев: приём и сохранение измерений от цифрового трансформатора, обработку и запись GOOSE- и DNP3-событий и приём логов SCADA-эмулятора.

Нефункциональное тестирование подтвердило, что система обладает высокой производительностью. Длительное тестирование показало отсутствие утечек памяти и стабильное потребление ресурсов.

5. ЗАКЛЮЧЕНИЕ

В ходе работы была создана информационная модель цифровой подстанции на основе современных протоколов, обеспечивающая точность измерений и надёжный обмен данными. Сначала был выполнен теоретический анализ аналогов и решений, что позволило обосновать выбор ключевого оборудования. На этапе проектирования архитектуры определили требования, выделили компоненты, разработали концептуальную и логическую модели и выбрали технологии.

Программная реализация включила настройку приёма SV-потокa через PySV, парсинг GOOSE- и DNP3-сообщений с помощью pyiec61850 и pydnp3, а также приём SCADA-логов через OPC UA-клиент. Использование локального эмулятора open62541 позволило тестировать приём логов без промышленного оборудования. Разработали модуль визуализации, который совмещает все данные в одном интерфейсе.

В разделе тестирования функциональные проверки подтвердили работоспособность всех сценариев: от приёма измерений до отображения на графиках.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. МЭК 61850-7-1:2011 Коммуникационные сети и системы для автоматизации электроэнергетики. – Международная электротехническая комиссия, 2011. – 89 с.
2. Кангин, В.В. Разработка SCADA систем: учебное пособие / В.В. Кангин, М.В. Кангин, Д.Н. Ямолдинов. – М.: Инфра Инженерия, 2023. – 564 с.
3. Лаптева, Е.А. Оценка эффективности реализации проекта «Цифровая подстанция 110/10 кВ» / Е.А. Лаптева, Л.Р. Уразбахтина // Modern Science. – 2023. – №3-2. – С. 21-28.
4. Гурина, Л.А. Угрозы и уязвимости объектов киберфизической энергетической системы при цифровой трансформации ее свойств / Л.А. Гурина, Т.Г. Зорина, Н.В. Тomin, С.Г. Прусов // Вестник Казанского государственного энергетического университета. – 2022. – Т. 14, №3 (55). – С. 89-98.
5. Хонг, Дж. Automated Cybersecurity Tester for IEC 61850 Based Digital Substations / Дж. Хонг, Т.-Дж. Сонг, Х. Ли, А. Заболи // Energies. – 2022. – Т. 15. – №21. – С. 1-17.
6. Гладковский, Г.К. Научные исследования технологии цифровой подстанции в ПАО «Россети» и ПАО «Россети ФСК ЕЭС» / Г.К. Гладковский, В.С. Чайкин // Энергия единой сети. – 2022. – №1 (62). – С. 12-19.
7. Тай, С.И. An Overview of Industry 4.0: Definition, Components, and Government Initiatives / С.И. Тай, Т.С. Ли, Н.А.А. Хамид, А.Н.А. Ахмад // Journal of Advanced Research in Dynamical and Control Systems. – 2018. – Т. 10, спец. вып. 14. – С. 1379-1386.
8. ГОСТ Р МЭК 61850-6–2009. Сети и системы связи на подстанциях. Часть 6. Язык описания конфигурации для связи между интеллектуальными

- электронными устройствами на электрических подстанциях. – М.: Изд-во стандартов, 2011. – 147 с.
9. МЭК 61850-8-1:2011. Communication networks and systems for power utility automation – In 8 parts. P.1: Specific Communication Service Mapping. – Женева: Международная электротехническая комиссия, 2011. – 128 с.
 10. Implementation Guideline for Digital Interface to Instrument Transformers using МЭК 61850-9-2 (МЭК 61850-9-2LE) / UCA International Users Group. – Ред. 2.1. – 07.07.2004. – 31 с.
 11. ГОСТ 7.53–2001. Издания. Международная стандартная нумерация книг. – М.: Изд-во стандартов, 2002. – 3 с.
 12. Schweitzer Engineering Laboratories. SEL 487В Реле дифференциальной защиты шины и реле УРОВ [Электронный ресурс]. – <https://selinc.com/ru/products/487B/> (дата обращения: 12.03.2025).
 13. VanderPlas J. Python Data Science Handbook: Essential Tools for Working with Data / J. VanderPlas. – Sebastopol: O'Reilly Media, 2016. – 548 с.
 14. Python OPC-UA Documentation. – FreeOpcUa Project, 2021. – <https://python-opcua.readthedocs.io/en/latest/> (дата обращения: 17.03.2025).
 15. Гиниятов, С.А. Применение протокола Sampled Values в системах РЗА цифровой подстанции / С.А. Гиниятов, Р.Г. Мустафин. – Казань: ООО ПК «Астор и Я», 2021. – Т. 1. – С. 324-327.

ПРИЛОЖЕНИЕ А. ЛИСТИНГИ ПРОГРАММНОГО КОДА

Листинг А.1 – Основная программа

```
def main_loop(poll_interval=5):
    try:
        while True:
            #Приём данных от ЦИТ
            transformer_data = fetch_transformer_data()
            record_measurement(transformer_data)
            #Приём событий от IED
            ied_evt = fetch_ied_events()
            if ied_evt:
                record_event(ied_evt)
            #Приём логов от SCADA
            scada_msg = fetch_scada_logs()
            if scada_msg:
                record_log(scada_msg)
            #Задержка перед следующим опросом
            time.sleep(poll_interval)
        except KeyboardInterrupt:
            print("Опрос остановлен пользователем.")
        finally:
            conn.close()

if __name__ == '__main__':
    main_loop()
```

Листинг А.2 – Таблицы базы данных

```
#Если файл базы не существует, он будет создан.
conn = sqlite3.connect('substation.db')
cursor = conn.cursor()
```

```
#measurements: агрегированные данные от ЦИТ
#events: срабатывания от IED
#logs: журналы от SCADA
cursor.execute('''
CREATE TABLE IF NOT EXISTS measurements (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    timestamp TEXT,
    voltage REAL,
    current REAL
)
''')
cursor.execute('''
CREATE TABLE IF NOT EXISTS events (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    timestamp TEXT,
    ied_id TEXT,
    event_type TEXT,
    details TEXT
)
''')
cursor.execute('''
CREATE TABLE IF NOT EXISTS logs (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    timestamp TEXT,
    message TEXT
)
''')
```

Продолжение листинга А.2

```
'''
conn.commit()
```

Листинг А.3 – Функции для записи данных в БД

```
def record_measurement(data):
#Сохранение агрегированных измерений в таблицу measurements
    cursor.execute('''
        INSERT INTO measurements (timestamp, voltage, current) VALUES (?, ?, ?)
        ''', (datetime.utcnow().isoformat(), data['voltage'], data['current']))
    conn.commit()

def record_event(evt):
#Сохранение событий IED в таблицу events
    cursor.execute('''
        INSERT INTO events (timestamp, ied_id, event_type, details) VALUES (?, ?, ?,
?)
        ''', (datetime.utcnow().isoformat(), evt['ied_id'], evt['event_type'],
evt['details']))
    conn.commit()

def record_log(message):
#Сохранение лога из SCADA в таблицу logs
    cursor.execute('''
        INSERT INTO logs (timestamp, message) VALUES (?, ?)
        ''', (datetime.utcnow().isoformat(), message))
    conn.commit()
```

Листинг А.4 – Функция для приёма SV-поток

```
subscriber = SVSubscriber(interface="eth0")
def fetch_transformer_data():
    packet = subscriber.receive() #Получить SV-пакет
    voltage = packet.get_voltage()#Извлечь напряжение
    current = packet.get_current()#Извлечь ток
    return {'voltage': voltage, 'current': current}
```

Листинг А.5 – Парсер GOOSE-сообщений и DNP3

```
goose_if = "eth0" #имя сетевого интерфейса
goose_mac = "01:0c:cd:01:00:01" #MAC-адрес GOOSE-группы ЦИТ
goose_sub = GooseSubscriber(interface=goose_if, mac=goose_mac)
#Настройка DNP3-мастера
config = opendnp3.MasterStackConfig()
config.master.retry = asiopal.ChannelRetry().Default()
config.master.responseTimeout = openpal.TimeDuration().Seconds(5)
channel = asiodnp3.IOManager().AddTCPClient("dnp3",
asiopal.ChannelRetry().Default(),
"127.0.0.1", 20000) #IP и порт IED
stack = channel.AddMaster("master", opendnp3.DefaultMasterApplication(),
opendnp3.MasterStackConfig())
stack.AddClassRequest(opendnp3.ClassField.AllClasses)
...
def fetch_ied_events(timeout=1.0):
```

Продолжение листинга А.5

```

deadline = time.time() + timeout
#Попытка прочитать GOOSE
while time.time() < deadline:
    goose_msg = goose_sub.receive(timeout=100)
    if goose_msg:
        return {
            'ied_id': goose_msg.appid,
            'protocol': 'GOOSE',
            'event_type': goose_msg.dataset,
            'details': f"stNum={goose_msg.stNum}, data={goose_msg.data}"
        }
#Если GOOSE не пришло – пробуем DNP3
#Опрос DNP3-мастера
class_resp = stack.Read(opendnp3.ClassField.AllClasses)
time.sleep(0.1)
#Получаем последние события из буфера мастера
events = stack.GetOperationMetrics().numUnsolicitedResponses
if events > 0:
    return {
        'ied_id': 'DNP3-IED',
        'protocol': 'DNP3',
        'event_type': 'UNSOLICITED',
        'details': f"unsolResponses={events}"
    }
return None

```

Листинг А.6 – Функция для приёма логов

```

from opcua import Client, ua
...
#Глобальный OPC UA клиент, создание подключения
_scada_client = Client("opc.tcp://192.168.1.100:4840")
_scada_client.connect()
_log_node = _scada_client.get_node("ns=2;i=1023") #путь к узлу логов
_subscription = _scada_client.create_subscription(1000, None)
#Очередь для хранения входящих событий
_event_queue = []
class LogHandler(object):
    def datachange_notification(self, node, val, data):
        #При изменении узла помещение записи в очередь
        timestamp = datetime.utcnow().isoformat()
        _event_queue.append((timestamp, str(val)))
#Подписка на события изменения узла
_subscription.subscribe_data_change(_log_node, LogHandler())
...
def fetch_scada_logs():
    if _event_queue:
        timestamp, message = _event_queue.pop(0)
        return {'timestamp': timestamp, 'message': message}
    return None

```

Листинг А.7 – Реализация визуализации данных

```

def plot_measurements_and_logs():
    # Подключаемся к базе данных по заданному пути
    conn = sqlite3.connect(DB_PATH)

    # Загружаем таблицу измерений (timestamp, voltage, current) в DataFrame
    df = pd.read_sql_query(
        "SELECT timestamp, voltage, current FROM measurements",
        conn, parse_dates=['timestamp']
    )
    # Загружаем таблицу логов (timestamp, message) в отдельный DataFrame
    df_logs = pd.read_sql_query(
        "SELECT timestamp, message FROM logs",
        conn, parse_dates=['timestamp']
    )
    # Закрываем соединение с базой данных
    conn.close()

    # Проверка на наличие данных в таблице измерений
    if df.empty:
        print("Нет данных для отображения.")
        return
    # Сортируем данные по времени
    df = df.sort_values('timestamp')
    df_logs = df_logs.sort_values('timestamp')

    # Создаем фигуру и ось для построения графика
    fig, ax = plt.subplots(figsize=(12, 6))

    # Строим графики напряжения и тока по времени
    ax.plot(df['timestamp'], df['voltage'], label='Voltage, kV')
    ax.plot(df['timestamp'], df['current'], label='Current, A')

    # Настраиваем отображение меток времени на оси X
    ax.xaxis.set_major_locator(mdates.AutoDateLocator())
    ax.xaxis.set_major_formatter(mdates.DateFormatter('%H:%M:%S'))
    plt.xticks(rotation=45) # Поворачиваем подписи по оси X для читаемости

    # Если логи не пустые, добавляем вертикальные линии и подписи к событиям
    if not df_logs.empty:
        # Определяем максимальное значение по оси Y для корректного
        # размещения подписей
        y_max = max(df['voltage'].max(), df['current'].max())
        for _, row in df_logs.iterrows():
            ts = row['timestamp']
            msg = row['message']
            # Добавляем вертикальную пунктирную линию на момент события
            ax.axvline(ts, color='gray', linestyle='--', alpha=0.7)
            # Добавляем текст сообщения над линией
            ax.text(ts, y_max * 1.01, msg, rotation=90,
                    va='bottom', ha='center', fontsize=8)

    # Настраиваем подписи и внешний вид графика
    ax.set_xlabel('Time (UTC)')
    ax.set_ylabel('Value')
    ax.set_title('Transformer Measurements and SCADA Logs')
    ax.legend(loc='upper left')
    ax.grid(True)
    plt.tight_layout()
    plt.show()

```