



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт: Информационных технологий (ИТ)
Кафедра: Базовая кафедра №234 — Управляющих ЭВМ

КУРСОВАЯ РАБОТА

по дисциплине Программное обеспечение систем управления технологическими процессами
(наименование дисциплины)

Тема курсовой работы: Система управления для мониторинга и контроля работы системы вентиляции в зданиях

Студент группы: ИКМО-05-23 Миронов Д.С.
(учебная группа, фамилия, имя, отчество) (подпись)

Руководитель курсовой работы: доцент, к.т.н., Глухов А.В.
(должность, звание, ученая степень, фамилия, имя, отчество) (подпись)

Работа предоставлена к защите до «__» января 2025 г.

Допущена к защите до «__» января 2025 г.

Москва 2025

Оглавление

ВВЕДЕНИЕ	4
1 ТЕХНИЧЕСКОЕ ЗАДАНИЕ	5
1.1 Общие правила выполнения работы	5
1.2 Вариант задания	5
2 АНАЛИТИЧЕСКАЯ ЧАСТЬ	6
2.1 Анализ общих правил выполнения работы	6
2.2 Анализ варианта задания	6
2.3 Анализ схемы вентиляции	7
2.3.1 Затвор вентиляции	8
2.3.2 Воздушный фильтр	8
2.3.3 • Водяной калорифер	8
2.3.4 Вентилятор	8
2.4 Анализ ограничений вентиляции	8
2.5 Стандарт Modbus-TCP	9
2.5.1 Обзор технологии Modbus-TCP	9
2.5.2 Основные характеристики Modbus-TCP	9
2.5.3 Преимущества протокола	9
2.5.4 Применение в системах вентиляции	10
2.5.5 Функции Modbus-TCP в SCADA-системах	10
3 ПРАКТИЧЕСКАЯ ЧАСТЬ	11
3.1 Создание АРМ	11
3.2 Добавление пользователей и ролей	11
3.3 Окна	12
3.4 Переменные системы	15
3.4.1 Раздел ресурсы	16
3.4.2 Раздел управление	17
3.4.3 Раздел предупреждения (уставки)	17
3.4.5 Раздел предупреждения (групповые)	18
3.4.6 Раздел папка 1	18
3.5 Управляющая программа	18
3.6 Обоснование выбора технических средств	19
3.6.1 Выбор языка программирования — Python	19
3.6.2 Выбор библиотеки — pymodbus	19
3.7 Сервер на Python	20

3.8 Работа системы.....	29
ЗАКЛЮЧЕНИЕ.....	31
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	32

ВВЕДЕНИЕ

Современные здания, особенно коммерческие, промышленные и социальные объекты, требуют наличия эффективных систем вентиляции, обеспечивающих оптимальные условия микроклимата и безопасности для находящихся в них людей. Управление такими системами вручную затруднено из-за высокой сложности технологических процессов, большого количества параметров и необходимости оперативного реагирования на изменения состояния оборудования и среды. В связи с этим возрастает потребность в автоматизированных системах управления (АСУ), которые способны не только контролировать параметры, но и обеспечивать надежную, стабильную и безопасную работу вентиляционного оборудования.

В данной курсовой работе рассматривается проектирование и разработка SCADA-системы, предназначенной для мониторинга и управления системой вентиляции в здании. Система реализуется с использованием архитектуры клиент-сервер и взаимодействует с контроллерами по протоколу Modbus-TCP, что позволяет собирать данные с оборудования и передавать команды управления в реальном времени. Особое внимание в работе уделено обеспечению безопасности эксплуатации, а именно — реализации механизмов аварийного оповещения, регистрации событий, ведения архивов, а также разграничения прав доступа пользователей.

В рамках проекта реализуется визуализация технологического процесса с графическим отображением схемы взаимодействия оборудования и автоматизированного рабочего места (АРМ) оператора. Для каждого контролируемого параметра предусмотрены предупредительные и аварийные уставки, аварийные сообщения сопровождаются звуковой сигнализацией и отображаются на экране АРМ. Все технологические параметры и сообщения архивируются для последующего анализа.

Проектируемая система также обеспечивает управление основными исполнительными механизмами вентиляции — вентиляторами, заслонками, клапанами, регуляторами расхода воздуха — и позволяет оперативно реагировать на возникающие неисправности или отклонения от нормального режима работы.

1 ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1.1 Общие правила выполнения работы

SCADA-систему реализовать в среде MasterSCADA 4D. Сервер Modbus-TCP реализовать на любом языке программирования. Для каждой работы необходимо использовать управляющие функции, которые будут осуществлять отображение и контроль технологического процесса. Параметры технологического процесса получать и отправлять на сервер. Спецификация сервера указана в задании.

Для каждого отслеживаемого параметра необходимо реализовать аварийное сообщение. Для аналоговых параметров реализовать верхние\нижние предупредительные и аварийные уставки.

Выдавать аварийные сообщения на экран АРМ. Сопровождать аварийное сообщение звуковой сигнализацией. Архивировать контролируемые параметры и все сообщения технологического процесса.

Реализовать аутентификацию пользователей и разграничение прав доступа различных категорий пользователей.

Создать сценарии аварийного отказа оборудования.

Разделить события на 2 группы — аварийные и события оборудования. События управления оборудованием вести в отдельном окне и выдавать по запросу оператора.

Реализовать графическое отображение технологического процесса. Отобразить схему взаимодействия компонентов технологического оборудования с АРМ оператора.

1.2 Вариант задания

Разработать систему управления для мониторинга и контроля работы системы вентиляции в зданиях. Данные получать с сервера Modbus-TCP. АСУ должна обеспечивать мониторинг основных параметров работы системы вентиляции, таких как температура, влажность, уровень CO₂, скорость воздушного потока, давление и т.д. SCADA-система должна обеспечивать возможность управления системой вентиляции внутри здания. Управление системой включает: регулирование скорости вентиляторов, контроль открытия и закрытия клапанов, регулирование расхода воздуха для разных зон в здании. SCADA-система должна обеспечивать мониторинг качества воздуха в здании, такого как содержание CO₂, VOC (летучие органические соединения), пыль и другие загрязнители в воздухе.

SCADA-система должна обладать функцией обнаружения аварийных ситуаций, таких как отказ вентилятора, перекрытие вентиляционных отверстий, понижение или повышение давления воздуха и т.д. При обнаружении аварийной ситуации система должна немедленно оповестить оператора и предпринять необходимые меры, например, отправить оповещение или команду на восстановление нормальной работы.

2 АНАЛИТИЧЕСКАЯ ЧАСТЬ

2.1 Анализ общих правил выполнения работы

1. «SCADA-систему реализовать в среде MasterSCADA 4D»

SCADA (Supervisory Control And Data Acquisition) – это система управления и сбора данных, которая широко применяется в различных отраслях промышленности.

Она собирает данные с различных датчиков и устройств контроля, обрабатывает их и отображает в понятном виде. Кроме того, SCADA позволяет удалённо управлять технологическими процессами.

2. «Сервер Modbus-TCP реализовать на любом языке программирования»

3. «Для работы необходимо использовать управляющие функции, которые будут осуществлять отображение и контроль технологического процесса»

Управляющие функции – это программы в среде MasterSCADA 4D реализованные с помощью языков стандарта МЭК 61131-3. К таким языкам относятся:

- a. FBD (Function Block Diagram) — Диаграммы функциональных блоков
- b. LD (Ladder Diagram) — Релейно-контактные схемы
- c. SFC (Sequential Function Chart) — Последовательные функциональные диаграммы
- d. ST (Structured Text) — Структурированный текст

4. «Параметры технологического процесса получать и отправлять на сервер. Спецификация сервера указана в задании»

5. «Для каждого отслеживаемого параметра необходимо реализовать аварийное сообщение»

6. «Для аналоговых параметров реализовать верхние\нижние предупредительные и аварийные уставки»

Уставка – это значение некоторой величины или параметра, по достижении которого происходит изменение состояния системы.

7. «Для каждого отслеживаемого параметра необходимо реализовать аварийное сообщение»

8. «Выдавать аварийные сообщения на экран АРМ»

АРМ – Автоматизированное рабочее место.

9. «Сопровождать аварийное сообщение звуковой сигнализацией»

10. «Архивировать контролируемые параметры и все сообщения технологического процесса»

11. «Реализовать аутентификацию пользователей и разграничение прав доступа различных категорий пользователей»

12. «Создать сценарии аварийного отказа оборудования»

13. «Разделить события на 2 группы — аварийные и события оборудования. События управления оборудованием вести в отдельном окне и выдавать по запросу оператора»

14. «Реализовать графическое отображение технологического процесса»

15. «Отобразить схему взаимодействия компонентов технологического оборудования с АРМ оператора»

2.2 Анализ варианта задания

1. «Данные получать с сервера Modbus-TCP»

В контексте задания, данные — это параметры технологического процесса, которые эмулируются на стороне сервера.

2. *«Вентиляция включается-выключается оператором»*
Интерфейс оператора SCADA системы должен подразумевать наличие кнопки включения и выключения вентиляции.
3. *«Регулирование скорости вентиляторов»*
Увеличение скорости вентиляторов происходит на стороне Modbus-TCP сервера и передаются в SCADA систему для обработки дальнейшей логики.
4. *«Контроль открытия и закрытия клапанов»*
5. *«Открытие и закрытие клапанов происходит по достижению желаемой температуры»*
6. *«Мониторинг основных параметров работы системы вентиляции, таких как температура, влажность, уровень CO2, скорость воздушного потока, давление»*
Интерфейс оператора SCADA системы должен подразумевать наличие полей для отображения данных.
7. *«Желаемую температуру определяет оператор»*
Интерфейс оператора SCADA системы должен подразумевать наличие поля ввода желаемой температуры.
8. *«При возникновении аварийной ситуации — выдавать сообщение оператора»*
9. *«Вероятность аварийной ситуации установить из графического интерфейса АРМ»*

2.3 Анализ схемы вентиляции

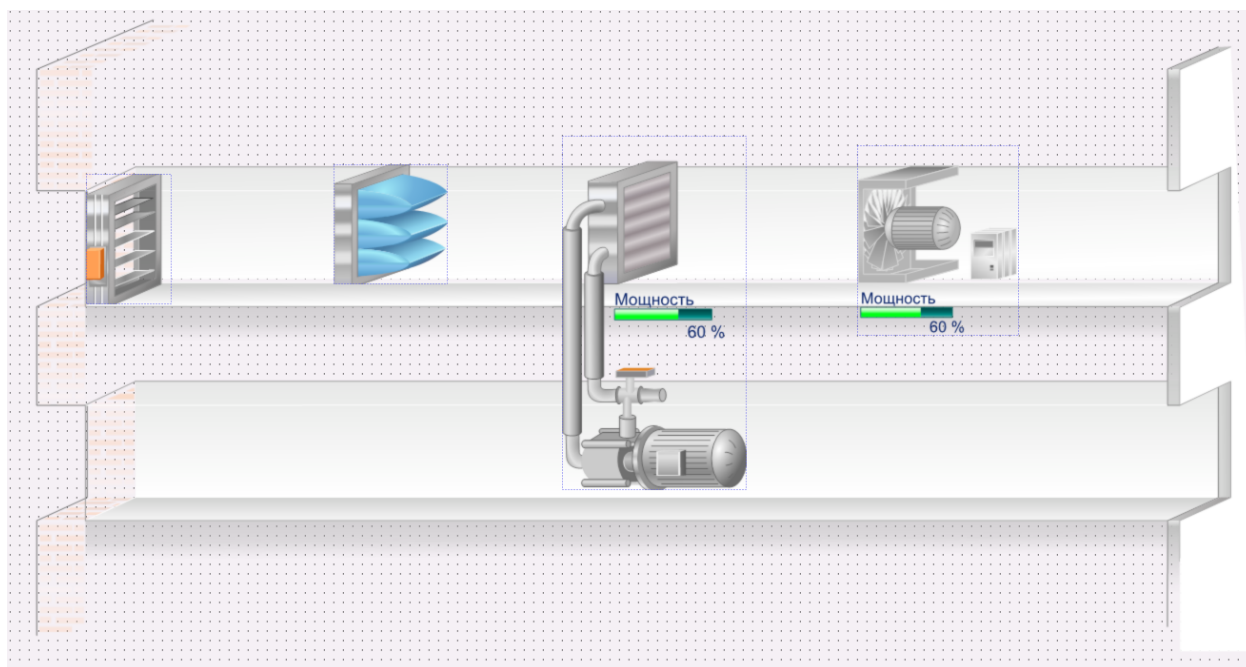


Рисунок 1 - Схема вентиляции

На схеме вентиляции присутствуют следующие элементы (см. Рисунок 1):

- Затвор вентиляции
- Воздушный фильтр
- Водяной калорифер

- Вентильатор

2.3.1 Затвор вентиляции

Затвор вентиляции — это элемент системы вентиляции, предназначенный для полного перекрытия или регулировки потока воздуха в вентиляционных каналах. Затворы применяются для обеспечения герметичности, отключения отдельных участков системы при обслуживании или авариях, а также для управления направлением воздушных потоков. Они могут иметь ручное или автоматическое управление (с сервоприводом) и часто используются в составе противопожарных и противодымных систем.

2.3.2 Воздушный фильтр

Воздушный фильтр — это устройство для очистки воздуха от загрязняющих частиц (пыли, сажи, пыльцы, микрочастиц и пр.) перед его подачей в помещения или обратно в систему вентиляции. Фильтры повышают качество воздуха и защищают оборудование от загрязнения. Существуют фильтры грубой, средней и тонкой очистки, а также высокоэффективные фильтры (HEPA, ULPA), применяемые в чистых помещениях и медицинских учреждениях.

2.3.3 • Водяной калорифер

Водяной калорифер — это теплообменник, использующий горячую воду в качестве теплоносителя для подогрева воздуха в вентиляционной системе. Он состоит из трубок и оребрения, через которые циркулирует вода, отдающая тепло проходящему через калорифер воздуху. Используется в приточных вентиляционных установках, особенно в холодное время года, для создания комфортной температуры воздуха в помещениях.

2.3.4 Вентильатор

Вентильатор — это основной приводной элемент вентиляционной системы, создающий движение воздуха по воздуховодам. Он обеспечивает подачу свежего воздуха (приток) и/или удаление загрязнённого воздуха (вытяжка). Вентильаторы классифицируются по принципу действия на:

- **Осевые** — перемещают воздух вдоль оси вращения (применяются при малом сопротивлении);
- **Центробежные (радиальные)** — создают высокое давление и используются при больших сопротивлениях в системе;
- **Диагональные (смесовые)** — сочетают свойства осевых и радиальных. Могут быть оборудованы системами регулирования оборотов для экономии энергии и точной настройки воздушных потоков.

2.4 Анализ ограничений вентиляции

При проектировании и эксплуатации автоматизированной системы вентиляции необходимо учитывать ряд ограничений, которые влияют на эффективность, надёжность и безопасность работы всей системы. Эти ограничения могут быть связаны как с техническими характеристиками оборудования, так и с внешними условиями эксплуатации, нормативными требованиями и архитектурными особенностями здания.

Ограничения по производительности оборудования

Каждое вентиляционное устройство — вентильаторы, калориферы, фильтры, клапаны — имеет пределы по расходу воздуха, давлению, температуре и другим параметрам. Превышение допустимых значений может привести к выходу оборудования из строя, снижению ресурса и созданию аварийных ситуаций. Минимальная температура – это

температура, при которой система ещё может работать, но ниже которой система может выйти из строя или быть неэффективной.

Нижняя аварийная уставка – это температура, при которой система должна немедленно принять меры для предотвращения повреждений, например, остановить работу, включить обогрев или другие защитные механизмы. Это критический минимум.

2.5 Стандарт Modbus-TCP

2.5.1 Обзор технологии Modbus-TCP

Стандарт Modbus-TCP

Modbus-TCP — это сетевой протокол прикладного уровня, предназначенный для обмена данными между устройствами автоматизации по компьютерной сети Ethernet. Он представляет собой расширение классического протокола Modbus, разработанного в 1979 году компанией Modicon, с сохранением базовой структуры и логики, но с адаптацией к работе по протоколу TCP/IP.

Modbus-TCP функционирует по архитектуре «клиент-сервер» (ранее — «ведущий-подчинённый»). Устройства, выступающие в роли клиента (обычно SCADA-системы или контроллеры), инициируют запросы, а серверы (контролируемое оборудование: датчики, исполнительные механизмы, ПЛК) формируют ответ. Связь осуществляется по протоколу TCP на стандартном порту 502.

В рамках стандарта Modbus-TCP используются те же команды, что и в Modbus RTU: чтение и запись дискретных входов/выходов, чтение/запись регистров, получение информации о состоянии устройства и др. Однако, в отличие от Modbus RTU, Modbus-TCP не использует контрольную сумму (CRC), так как функции контроля целостности данных берёт на себя транспортный уровень протокола TCP/IP.

2.5.2 Основные характеристики Modbus-TCP

- Тип подключения: Ethernet (10/100/1000 Мбит/с);
- Транспортный протокол: TCP/IP;
- Порт по умолчанию: 502;
- Формат сообщений: идентичен Modbus RTU, но без CRC;
- Тип архитектуры: клиент-сервер;
- Максимальное число устройств: до 247, ограничено сетевой топологией.

2.5.3 Преимущества протокола

- Высокая скорость передачи данных и надёжность соединения;
- Поддержка большинством современных ПЛК, датчиков и исполнительных механизмов;
- Простота внедрения и отладки;
- Возможность интеграции в существующие IP-сети без необходимости дополнительного оборудования;

- Хорошая масштабируемость и совместимость с системами диспетчеризации и мониторинга.

2.5.4 Применение в системах вентиляции

Протокол Modbus-TCP широко применяется в автоматизированных системах управления вентиляцией зданий. С его помощью осуществляется:

Сбор данных от датчиков температуры, влажности, концентрации CO₂, давления;

Управление скоростью вращения вентиляторов, положением клапанов и заслонок;

Мониторинг состояния оборудования и реализация аварийных сценариев;

Интеграция с сервером SCADA-системы, где осуществляется визуализация, архивация параметров и формирование тревожных сообщений.

Благодаря высокой универсальности и открытости стандарта, Modbus-TCP стал одним из наиболее популярных протоколов в системах промышленной автоматизации, включая HVAC (отопление, вентиляция и кондиционирование воздуха).

2.5.5 Функции Modbus-TCP в SCADA-системах

Сбор данных: SCADA получает в режиме реального времени значения аналоговых и дискретных сигналов — температура, влажность, давление, концентрация CO₂, состояние вентиляторов и клапанов и др.

Управление оборудованием: Через Modbus-TCP SCADA может передавать управляющие команды на исполнительные устройства: изменение скорости вентиляторов, открытие/закрытие заслонок, включение/выключение модулей.

Архивирование и аналитика: Полученные данные записываются в архив, используются для построения графиков, анализа трендов, формирования отчётов и принятия решений.

Обработка аварийных ситуаций: При выходе параметров за уставки или при неисправности оборудования SCADA получает соответствующий сигнал и активирует тревожные сообщения, световую и звуковую сигнализацию, а также может выполнять автоматические защитные действия.

3 ПРАКТИЧЕСКАЯ ЧАСТЬ

SCADA система была реализована в среде разработки MasterSCADA 4D версии 1.3.6.21852(1.3.6_20241201.1).

3.1 Создание АРМ

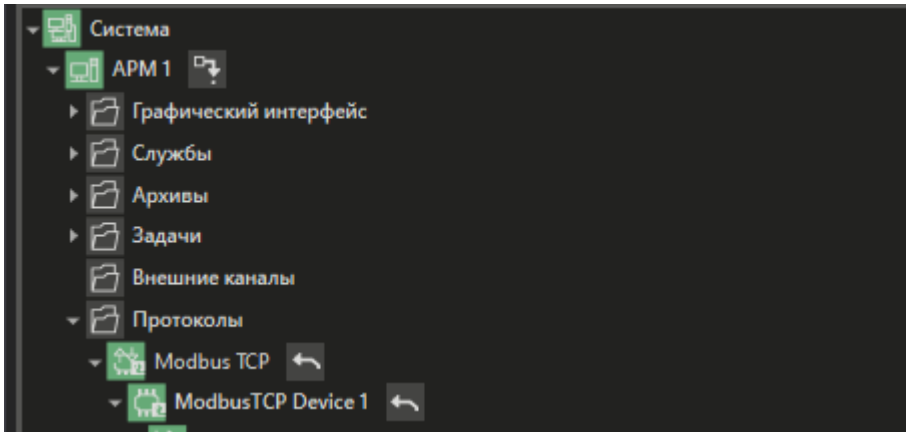


Рисунок 2 – АРМ оператора

В системе предусмотрен один АРМ оператора с протоколом Modbus TCP (см. Рисунок 2).

3.2 Добавление пользователей и ролей

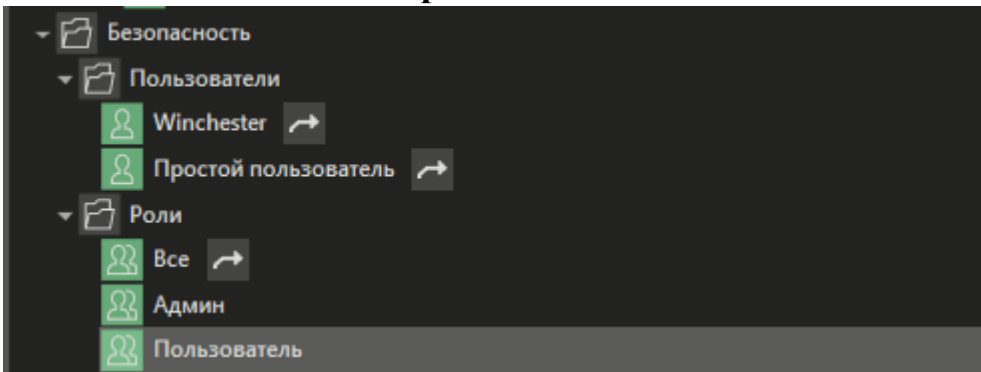


Рисунок 3 – Пользователи и роли

В системе предусмотрены два пользователя: Winchester и Простой пользователь, а также две роли: пользователь и админ. Оператору не доступно редактирование настроек .

Право доступа: Управление					
Пользователь	Разрешить	Запретить	Подтверждение	Журнал	
Winchester	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Простой пользователь	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Все	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Админ	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Пользователь	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Рисунок 4 – Разграничение доступов

Вход в систему

Пользователь

Winchester

Пароль

Показать пароль ☐

Войти

Рисунок 5– Вход в систему

3.3 Окна

В системе реализовано 6 окон (см. Рисунки 7-11).

Приточная установка подвал

Журнал

Завершить сессию

Рисунок 6 – Основное окно выбора действий

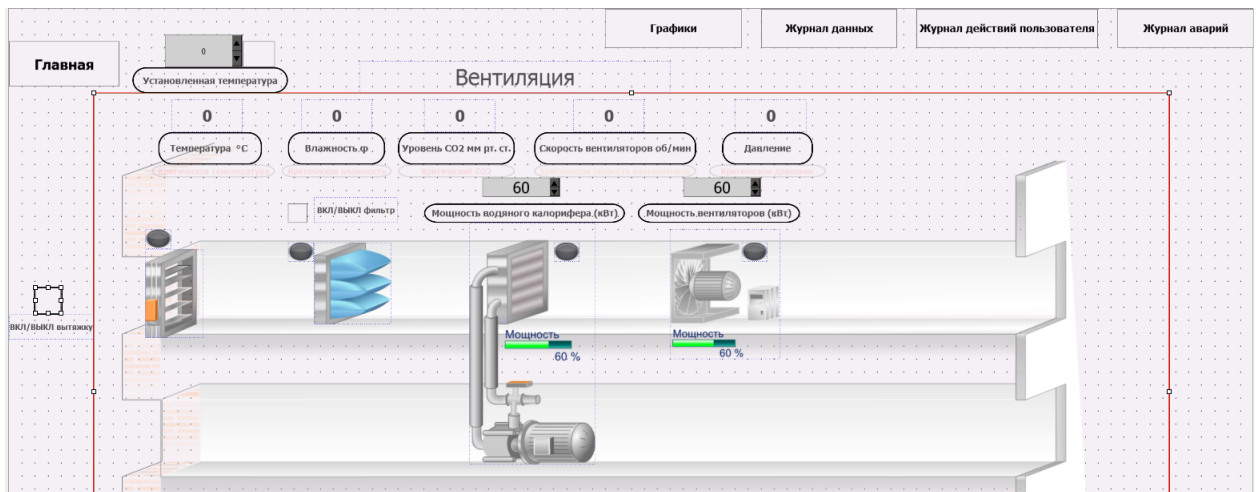


Рисунок 7 – Основное окно вентиляции

В основном окне технологического процесса существует несколько ключевых элементов:

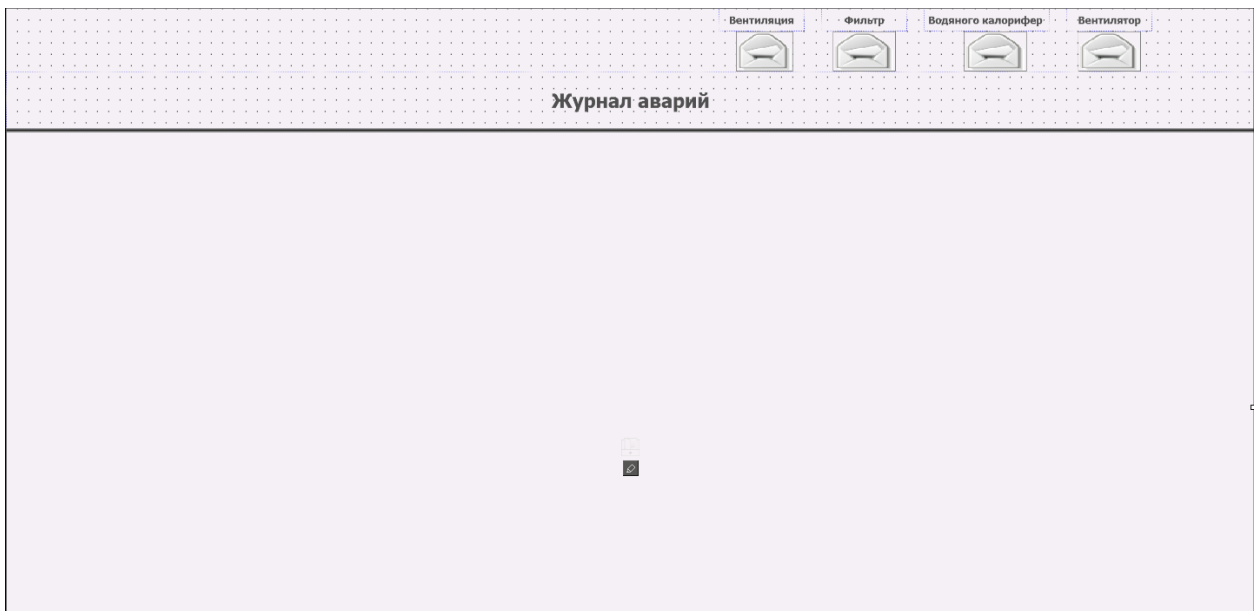


Рисунок 8 – Модальное окно журнал аварий

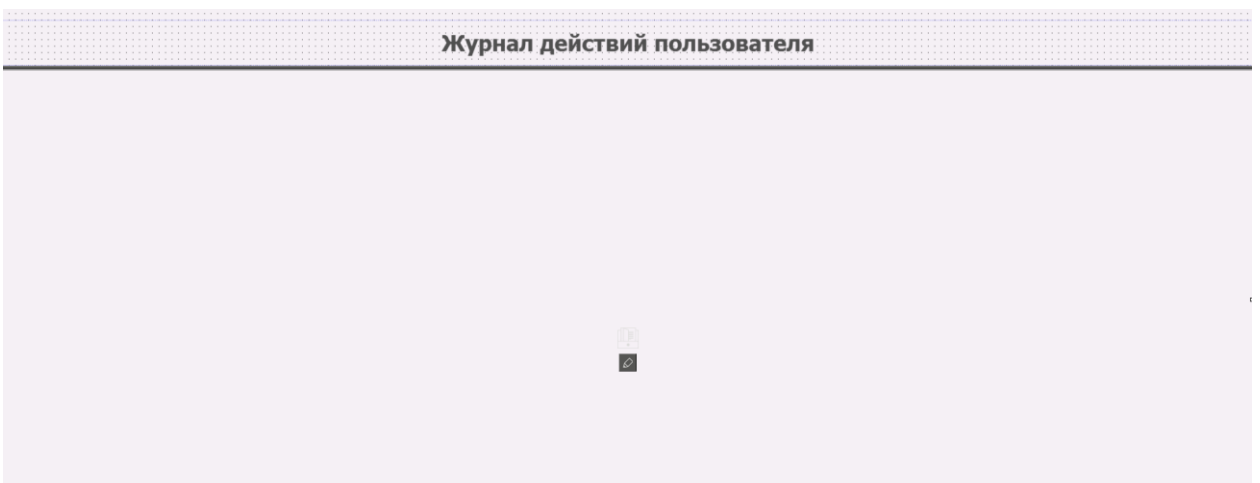


Рисунок 9 – Модальное окно журнал действий пользователя

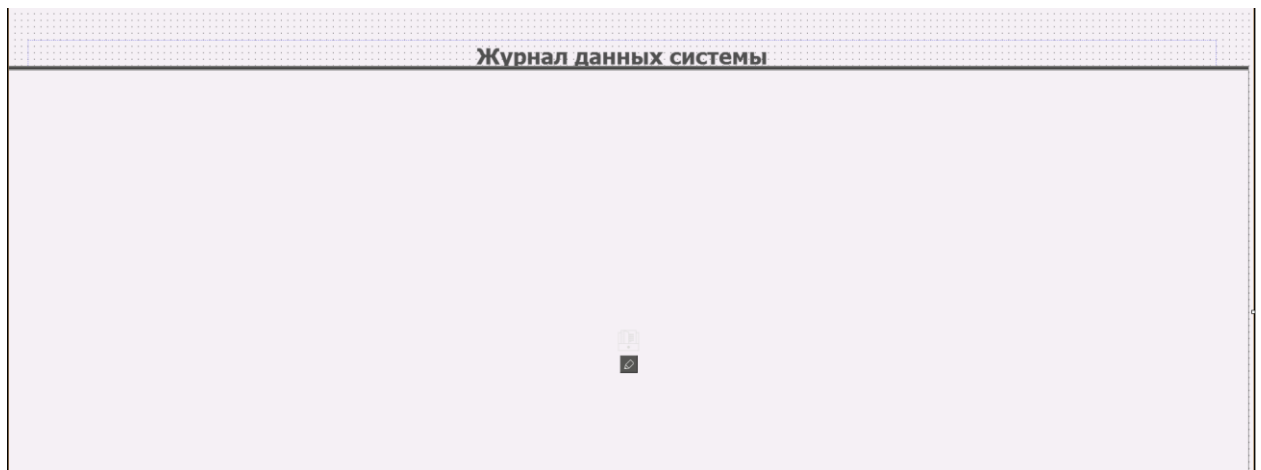


Рисунок 10 – Модальное окно журнал данных системы

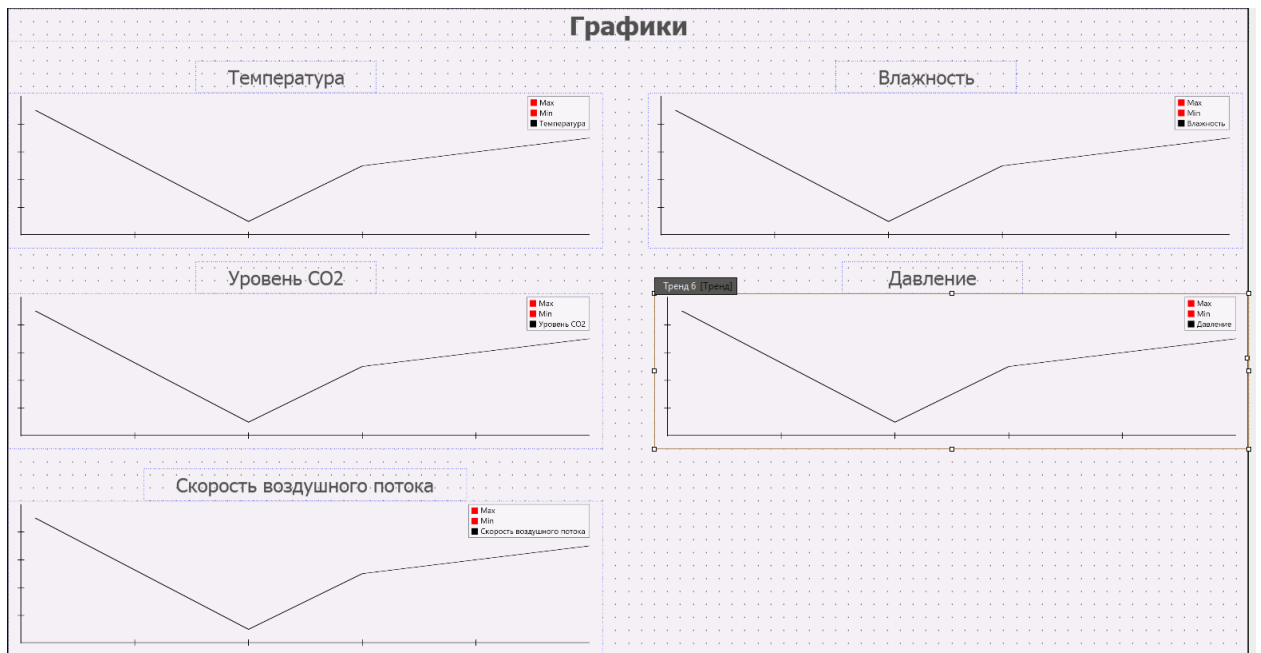


Рисунок 11 – Модальное окно графики

Основное окно включает:

- Настройки температуры
- Настройки затвора вентиляции
- Настройка фильтра
- Настройка водяного калорифера
- Настройка вентилятора
- Данные о текущей температуре
- Данные о текущей влажности
- Данные о текущем уровне CO2
- Данные о текущей скорости вентилятора
- Данные о текущем давлении

- Настройка мощности вентилятора
- Настройка мощности водяного калорифера

3.4 Переменные системы

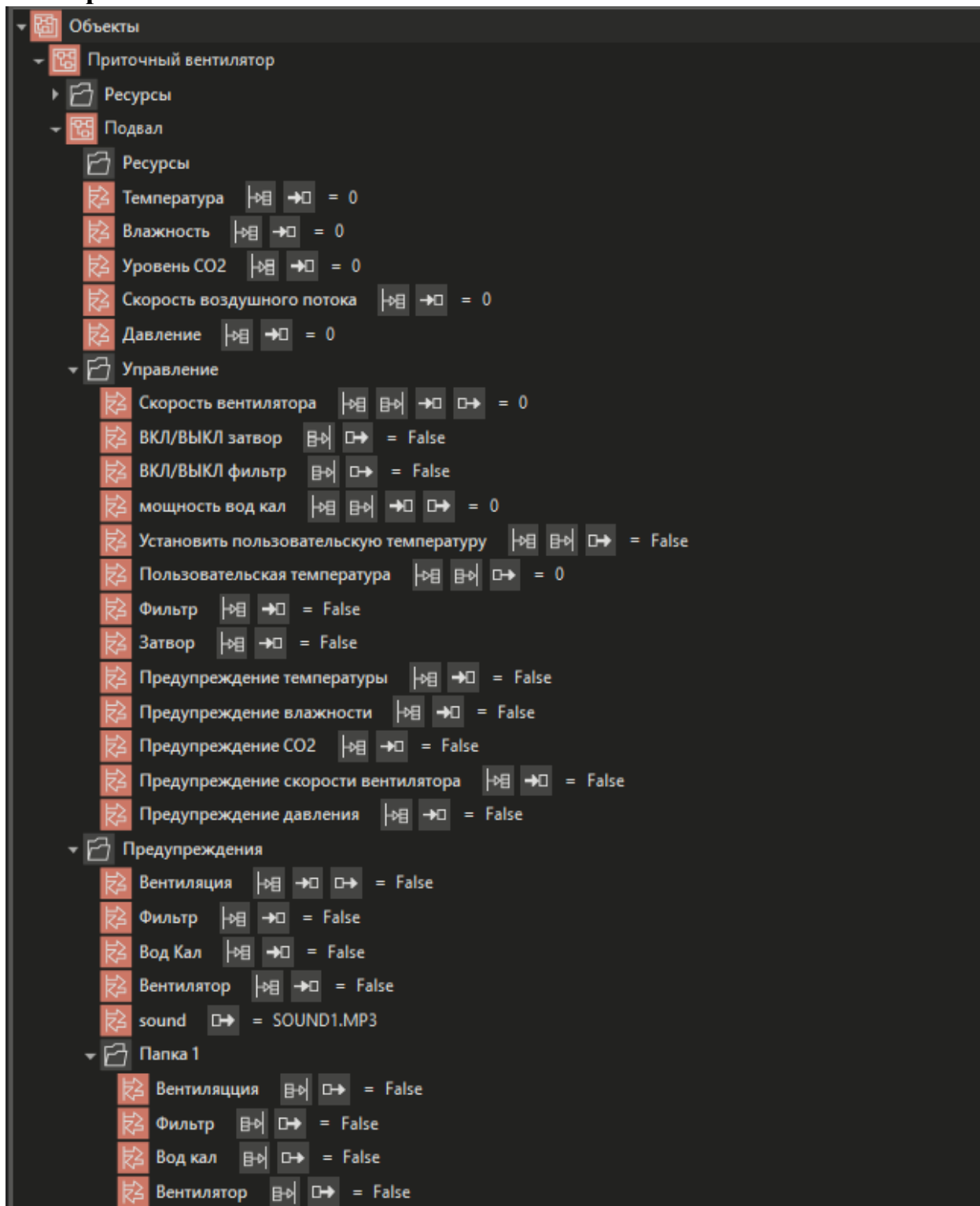


Рисунок 12 - Переменные системы



Рисунок 13 - Переменные сервера в MasterSCADA 4D

3.4.1 Раздел ресурсы

- **Температура:**
Показание температуры воздуха в системе вентиляции, измеряемое в зоне

установки датчика (например, в подвале). Используется для оценки микроклимата и работы водяного калорифера.

- **Влажность:**
Относительная влажность воздуха (%), отражает уровень водяного пара в воздухе. Важно для контроля качества воздуха и предотвращения образования конденсата.
- **Уровень CO₂:**
Концентрация углекислого газа в воздухе, измеряемая в ppm (parts per million). Высокий уровень CO₂ указывает на недостаточную вентиляцию.
- **Скорость воздушного потока:**
Линейная скорость движения воздуха в вентиляционных каналах. Используется для контроля эффективности воздухообмена.
- **Давление:**
Давление воздуха в системе, часто используется для контроля работоспособности вентилятора и наличия препятствий (засоров) в воздуховоде.

3.4.2 Раздел управление

- **Скорость вентилятора:**
Параметр, определяющий текущую скорость вращения вентилятора. Может регулироваться вручную или автоматически.
- **ВКЛ/ВЫКЛ затвор:**
Логический параметр (True/False), управляющий состоянием затвора (заслонки) в системе вентиляции — открытие или закрытие воздушного потока.
- **ВКЛ/ВЫКЛ фильтр:**
Включение или отключение фильтрации воздуха. При отключении фильтр может обходиться или не учитываться в логике управления.
- **Мощность вод кал (водяного калорифера) :**
Мощность нагрева, подаваемая на водяной калорифер (в условных единицах или процентах). Регулируется в зависимости от температуры воздуха.
- **Установить пользовательскую температуру:**
Флаг (True/False), разрешающий установку температуры вручную пользователем (например, оператором на АРМ).
- **Пользовательская температура:**
Значение температуры, заданное пользователем. Используется как целевая уставка для поддержания комфортных условий.
- **Фильтр:**
Отображает состояние фильтра (True — фильтр требует замены или загрязнён, False — фильтр в норме).
- **Затвор:**
Состояние воздушного затвора (открыт/закрыт). Используется для управления потоками воздуха в разных зонах здания.

3.4.3 Раздел предупреждения (уставки)

- **Предупреждение температуры**
Срабатывает при выходе температуры за допустимые границы (верхняя/нижняя уставка). Может указывать на неисправность калорифера.
- **Предупреждение влажности**
Сигнализирует о чрезмерной или слишком низкой влажности воздуха. Может повлиять на комфорт и здоровье пользователей.

- **Предупреждение CO₂:**
Активируется при превышении допустимого уровня углекислого газа. Требуется увеличения вентиляции или вмешательства оператора.
- **Предупреждение скорости вентилятора:**
Возникает при недостаточной или слишком высокой скорости вращения вентилятора. Может быть вызвано механической неисправностью или неправильным управлением.
- **Предупреждение давления:**
Фиксирует аварийные перепады давления (например, при засоре воздухопроводов или отключении вентилятора).

3.4.5 Раздел предупреждения (групповые)

- **Вентиляция:**
Общий флаг состояния аварий вентиляционной системы. Может включаться при срабатывании одного или нескольких из вышеуказанных предупреждений.
- **Фильтр:**
Групповое предупреждение по фильтру — например, превышение времени работы или загрязнение.
- **Вод Кал:**
Групповое предупреждение по водяному калориферу — может сигнализировать о перегреве или неисправности системы нагрева.
- **Вентилятор:**
Общая авария вентилятора — не вращается, перегрузка, обрыв фазы, и т.д.
- **SOUND1.MP3:**
Файл звукового оповещения, проигрываемый при возникновении аварии. Используется для привлечения внимания оператора.

3.4.6 Раздел папка 1

- **Вентиляция, Фильтр, Вод Кал, Вентилятор:**
Те же параметры, собранные в отдельную группу для дополнительной визуализации в другой части интерфейса.

3.5 Управляющая программа

Функция воспроизведения звуков

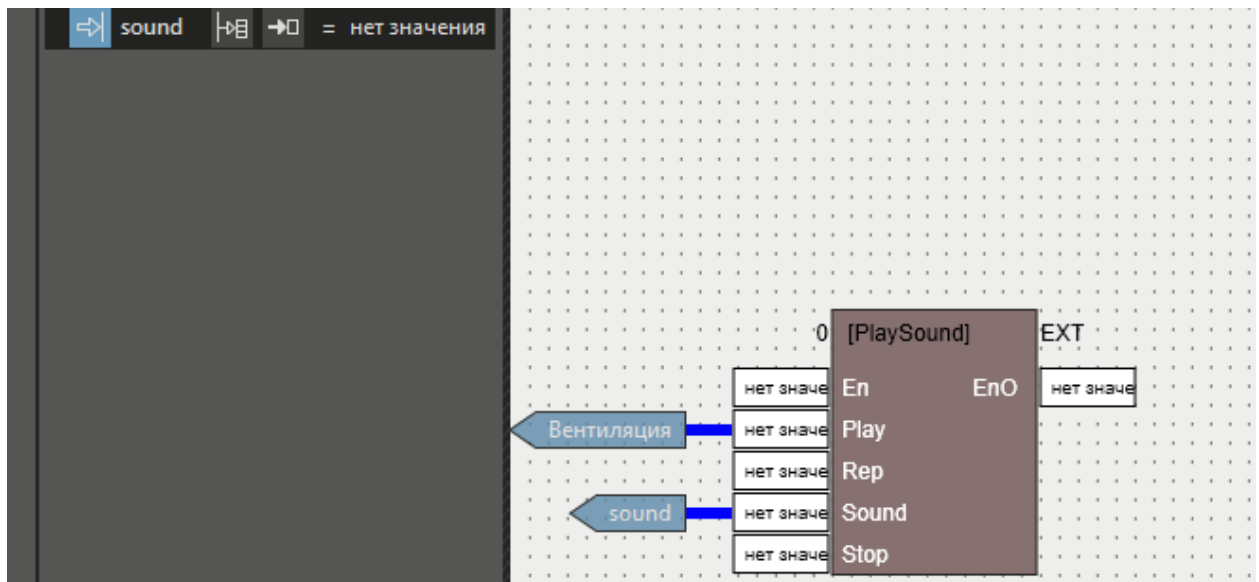


Рисунок 14 – Программа воспроизведения звука

3.6 Обоснование выбора технических средств

Для организации сбора, обработки и передачи данных от контролируемых объектов вентиляционной системы в SCADA-систему был реализован программный сервер по протоколу Modbus TCP. В качестве основного инструмента разработки выбран язык программирования **Python** с использованием специализированных библиотек. Обоснование данного выбора представлено ниже.

3.6.1 Выбор языка программирования — Python

Python является современным, широко используемым и активно развивающимся языком программирования, который обладает рядом преимуществ:

- **Простота синтаксиса** и высокая читаемость кода, что особенно важно при разработке и поддержке промышленных приложений.
- **Большое количество готовых библиотек**, в том числе для промышленной автоматизации и работы с протоколом Modbus.
- **Кроссплатформенность** — возможность запуска сервера на различных операционных системах (Windows, Linux, Raspberry Pi OS и др.).
- **Активное сообщество** и открытый исходный код большинства решений.

3.6.2 Выбор библиотеки — pymodbus

Для реализации сервера Modbus TCP была выбрана библиотека pymodbus, предоставляющая полный стек протокола Modbus и поддержку как **Master (клиента)**, так и **Slave (сервера)**. Основные причины выбора:

- Поддержка всех необходимых функций (чтение/запись регистров, битов, слов и др.).
- Возможность настройки частоты опроса, времени отклика, таймаутов.
- Лёгкость интеграции с другими библиотеками Python (например, NumPy, pandas, sqlite3, MQTT и др.).

- Простая реализация многопоточного или асинхронного сервера.

3.7 Сервер на Python

Листинг 9 – Сервер на python

```
import time

from pymodbus.server import StartTcpServer

from pymodbus.datastore import ModbusSlaveContext, ModbusServerContext,
ModbusSequentialDataBlock

from threading import Thread

import random

from datetime import datetime

# Создаем таблицу регистров (100 значений) с правильной инициализацией
store = ModbusSlaveContext(
    di=ModbusSequentialDataBlock(0, [0] * 100), # Дискретные входы
    co=ModbusSequentialDataBlock(0, [0] * 100), # Катушки (битовые значения)
    hr=ModbusSequentialDataBlock(0, [50] * 100), # 100 значений в Holding
Registers
    ir=ModbusSequentialDataBlock(0, [0] * 100) # Входные регистры
)

context = ModbusServerContext(slaves=store, single=True)

# Таймеры отказа датчиков
sensor_failure_timers = {
    'Вытяжка': 20, # 20
    'Фильтр': 25, # 25
    'Вод_кал': 30, # 30
    'Вентилятор': 35, # 35
}

address_error = {
    'Вытяжка': 15,
    'Фильтр': 16,
    'Вод_кал': 17,
    'Вентилятор': 18,
}

address_error_status = {
    'Вытяжка': False,
    'Фильтр': False,
    'Вод_кал': False,
    'Вентилятор': False,
```

```

}

# Счетчик времени для плавного уменьшения влажности
humidity_decrease_counter = 0

# Флаги отказов датчиков
sensor_failures = {
    "humidity": False,
    "temperature": False,
    "irrigation": False,
    "ventilation": False,
    "check_fan": False,
    "check_speed_fan": 60,
    "check_water_cal": 60,
    "check_filtr": False
}

local_state= {}

logs_actions= []
logs_system=[]
logs_errors =[]

def main_function( humidity_decrease_counter ):

    while True:

        # Чтение значений из Modbus регистров

        humidity = store.getValues(3, 1, count=1)[0]          # Влажность
(адрес 1)

        temperature = store.getValues(3, 2, count=1)[0]      # Температура
(адрес 2)

        c_o_2 = store.getValues(3, 4, count=1)[0]            # CO2 (адрес 4)

        pressure = store.getValues(3, 5, count=1)[0]          # Давление (адрес
5)

        airflow_speed = store.getValues(3, 6, count=1)[0]     # Скорость
воздушного потока (адрес 6)

        check_fan = store.getValues(3, 11, count=1)[0]        # вкл выкл вент
(адрес 11)

        check_fan2 = store.getValues(3, 23, count=1)[0]       # вкл выкл
вент (адрес 23)

        check_speed_fan = store.getValues(3, 12, count=1)[0]  #
мощность вент (адрес 12)

```

```

        check_water_cal = store.getValues(3, 13, count=1)[0]      #
мощность вод кал (адрес 13)

        check_filtr = store.getValues(3, 14, count=1)[0]          # Вкл Выкл
фильтр (адрес 14)

        check_filtr2 = store.getValues(3, 22, count=1)[0]          # Вкл Выкл
фильтр (адрес 22)

        temperature_user = store.getValues(3, 21, count=1)[0]      #
Установить температуры (адрес 21) # int

        temperature_user_use = store.getValues(3, 20, count=1)[0]  #
Установить температуры (адрес 20) # bool

    alarm_temperature = False

    alarm_humidity = False

    alarm_c_o_2 = False

    alarm_pressure = False

    alarm_airflow_speed = False

    went_error = store.getValues(3, 29, count=1)[0] # Вентиляция error
    airflow_error = store.getValues(3, 32, count=1)[0] # Вентилятор
error

    filtr_error = store.getValues(3, 30, count=1)[0] # Фильтр error
    whater_Cal_error = store.getValues(3, 31, count=1)[0] # Вод кал
error

    print(f'went_error {went_error} {type(went_error)} filtr_error
{filtr_error} whater_Cal_error {whater_Cal_error} airflow_error
{airflow_error} temperature_user {temperature_user} ')

    if temperature_user_use==1:
        temperature_user_use=True
    else:
        temperature_user_use=False

    sensor_failures['check_fan']= True if check_fan2 else False
    sensor_failures['check_speed_fan']= check_speed_fan
    sensor_failures['check_water_cal']= check_water_cal
    sensor_failures['check_filtr']= True if check_filtr2 else False
#True if isinstance(check_filtr, int) else check_filtr

    local_state ={          'Вытяжка':
sensor_failures['check_fan'],

```

```

        'мощность вент':
sensor_failures['check_speed_fan'],

        'мощность вод кал':
sensor_failures['check_water_cal'],

        'Вкл/Выкл фильтр':
sensor_failures['check_filtr'],

        'Time':str(datetime.now())

    }

    if logs_system and temperature_user_use:
        temp = logs_system[-1]
        if list(temp.values()) != list(local_state.values()):
            logs_system.append(local_state)
    else:
        logs_system.append(local_state)

# Моделирование отказа датчиков
for name, time_time in sensor_failure_timers.items():
    if sensor_failure_timers[name] > 0:
        sensor_failure_timers[name] -= 1
        store.setValues(3, address_error[name], [False]) # Сигнал об
отказе

    elif sensor_failure_timers[name] == 0:
        store.setValues(3, address_error[name], [True]) # Сигнал об
отказе

        address_error_status[name]=True
        logs_errors.append( address_error | {
'Time':str(datetime.now())} )

    if temperature_user_use and temperature_user: # если пользовательская
настройка

        if temperature_user > temperature:
            sensor_failures['check_fan']= True
            if check_speed_fan>40:
                check_speed_fan=check_speed_fan-1
            elif check_speed_fan<40:
                check_speed_fan=check_speed_fan+1
            sensor_failures['check_speed_fan']= check_speed_fan
            if check_water_cal>40:
                check_water_cal=check_water_cal-1
            elif check_water_cal<40:

```

```

        check_water_cal=check_water_cal+1
    sensor_failures['check_water_cal']=    check_water_cal
    sensor_failures['check_filtr']= True
    temperature =temperature + 1

    if humidity>40:
        humidity =    humidity -1
    elif humidity<40:
        humidity =    humidity +1
    if c_o_2>40:
        c_o_2 =    c_o_2 -1
    elif c_o_2<40:
        c_o_2 =    c_o_2 +1
    if pressure>40:
        pressure =    pressure -1
    elif pressure<40:
        pressure =    pressure +1
    if airflow_speed>40:
        airflow_speed =    airflow_speed -1
    elif c_o_2<40:
        airflow_speed =    airflow_speed +1
    elif temperature_user < temperature:
        sensor_failures['check_fan']= True
        if check_speed_fan<70:
            check_speed_fan=check_speed_fan+1
        elif check_speed_fan>70:
            check_speed_fan=check_speed_fan-1
        sensor_failures['check_speed_fan']=    check_speed_fan
        if check_water_cal<70:
            check_water_cal=check_water_cal+1
        elif check_water_cal>70:
            check_water_cal=check_water_cal-1
        sensor_failures['check_water_cal']=    check_water_cal
        sensor_failures['check_filtr']= True
        temperature =temperature - 1

    if humidity>30:
        humidity =    humidity -1

```



```

        elif humidity<40:
            humidity = humidity +1
        if c_o_2>30:
            c_o_2 = c_o_2 -1
        elif c_o_2<30:
            c_o_2 = c_o_2 +1
        if pressure>30:
            pressure = pressure -1
        elif pressure<30:
            pressure = pressure +1
        if airflow_speed>30:
            airflow_speed = airflow_speed -1
        elif c_o_2<30:
            airflow_speed = airflow_speed +1
    else:
        # sensor_failures['check_filtr']= False
        # sensor_failures['check_fan']= False
        humidity = humidity+ random.randint(1,7) if '+' ==
random.choice(["+", "-"]) else humidity- random.randint(1,7)
        temperature =temperature + random.randint(1,7) if '+' ==
random.choice(["+", "-"]) else temperature - random.randint(1,7)
        c_o_2 = c_o_2 + random.randint(1,7) if '+' ==
random.choice(["+", "-"]) else c_o_2 - random.randint(1,7)
        pressure =pressure + random.randint(1,7) if '+' ==
random.choice(["+", "-"]) else pressure - random.randint(1,7)
        airflow_speed =airflow_speed + random.randint(1,7) if '+' ==
random.choice(["+", "-"]) else airflow_speed - random.randint(1,7)

    if temperature> 80 or temperature <10:
        alarm_temperature=True
    if humidity> 70 or humidity < 10:
        alarm_humidity=True
    if c_o_2> 50 or c_o_2 < 10:
        alarm_c_o_2=True
    if pressure> 60 or pressure < 10:
        alarm_pressure=True
    if airflow_speed> 80 :
        alarm_airflow_speed=True

    if went_error==1:

```

```

        print('dsffsdfsdfsdfsdf')
        store.setValues(3, 15, [True])
    else:
        store.setValues(3, 15, [False])

    if filtr_error==1:
        store.setValues(3, 16, [True])
    else:
        store.setValues(3, 16, [False])

    if whater_Cal_error==1:
        store.setValues(3, 17, [True])
    else:
        store.setValues(3, 17, [False])

    if airflow_error==1:
        store.setValues(3, 18, [True])
    else:
        store.setValues(3, 18, [False])

    # Запись значений в Modbus
    store.setValues(3, 1, [humidity]) # Влажность (адрес 1)
    store.setValues(3, 2, [temperature]) # Температура (адрес 2)
    store.setValues(3, 4, [c_o_2]) # CO2 (адрес 4)
    store.setValues(3, 5, [pressure]) # Давление (адрес 5)
    store.setValues(3, 6, [airflow_speed]) # Скорость воздушного потока
(адрес 6)
    store.setValues(3, 11, [sensor_failures['check_fan']]) # вкл выкл
(адрес 11)
    store.setValues(3, 12, [sensor_failures['check_speed_fan']]) #
мощность вент (адрес 12)
    store.setValues(3, 13, [sensor_failures['check_water_cal']]) #
мощность вод кал (адрес 13)
    store.setValues(3, 14, [sensor_failures['check_filtr']]) # вкл выкл
фильтр (адрес 14)
    store.setValues(3, 21, [temperature_user]) #
    store.setValues(3, 20, [temperature_user_use]) #
    store.setValues(3, 22, [check_filtr2]) #
    store.setValues(3, 24, [alarm_temperature]) #
    store.setValues(3, 25, [alarm_humidity]) #
    store.setValues(3, 26, [alarm_c_o_2]) #

```

```

store.setValues(3, 27, [alarm_airflow_speed]) #
store.setValues(3, 28, [alarm_pressure]) #

print( f"Влажность={humidity}%, Температура={temperature}°C, "
      f"CO2 {c_o_2}%, "
      f"Давление {pressure}мм рт. ст., "
      f"Скорость воздушного потока {airflow_speed}об/мин, "
      f"Заглушка {'открыта' if sensor_failures['check_fan'] else
'закрыта'} {sensor_failures['check_fan']}, "
      f"Фильтр {'вкл' if sensor_failures['check_filtr'] else
'выкл'} {sensor_failures['check_filtr']}, "
      f"check_filtr2 {check_filtr2} "
      f"скорость вод кал {check_water_cal}, "
      f"скорость вент{sensor_failures['check_speed_fan']}, "
      f"Errors: {sensor_failure_timers}, "
      f"temperature_user {temperature_user},"
      f"temperature_user_use {temperature_user_use}"
      )

time.sleep(2) # Обновление каждые 5 секунд

# Запускаем обновление данных в отдельном потоке
Thread(target=main_function, daemon=True, args=(humidity_decrease_counter, )
).start()

# Запускаем сервер
print("Modbus-TCP сервер запущен на 127.0.0.1:5020...")
StartTcpServer(context, address=("127.0.0.1", 5020))

```

Инициализация регистров Modbus:

Создается объект `ModbusSlaveContext`, который представляет данные на сервере Modbus. Он включает дискретные входы, катушки, `holding`-регистры и входные регистры, которые инициализируются значениями по умолчанию.

Сетевые настройки и таймеры отказов:

Создаются словари для управления отказами датчиков, с указанием времени отказа и адресов для каждого устройства (например, вытяжка, фильтр, вентилятор).

Устанавливаются флаги для отказов датчиков и для записи ошибок.

Основная функция `main_function`:

В цикле выполняется чтение значений из регистров Modbus, таких как влажность, температура, уровень CO₂, давление и скорость воздушного потока.

Производится проверка пользовательских настроек (например, заданная температура). Если значения датчиков выходят за пределы допустимых норм, активируются соответствующие тревоги.

Если датчики выходят из строя (на основе таймеров отказов), отправляются сигналы в соответствующие регистры для уведомления о сбое.

Моделирование отказов датчиков:

В коде присутствует логика для имитации отказов датчиков на основе таймеров. Если отказ не произошел, время таймера уменьшается, в противном случае генерируется сигнал об ошибке.

Все изменения состояния датчиков записываются в регистры Modbus.

Регулировка значений в зависимости от настроек пользователя:

Если установлены пользовательские настройки температуры, программа регулирует значения температуры, влажности, CO₂ и других параметров, чтобы поддерживать значения в пределах заданного диапазона.

При этом контролируется работа вентилятора и фильтра, а также других систем.

Запуск потока и сервера:

- Основная функция `main_function` запускается в отдельном потоке, чтобы не блокировать работу сервера.
- Запускается сервер Modbus-TCP, который слушает запросы на порту 5020 и обслуживает запросы от клиентов.

Задержка перед следующим обновлением:

- Между итерациями цикла выполняется задержка в 2 секунды для плавного изменения параметров.

3.8 Работа системы

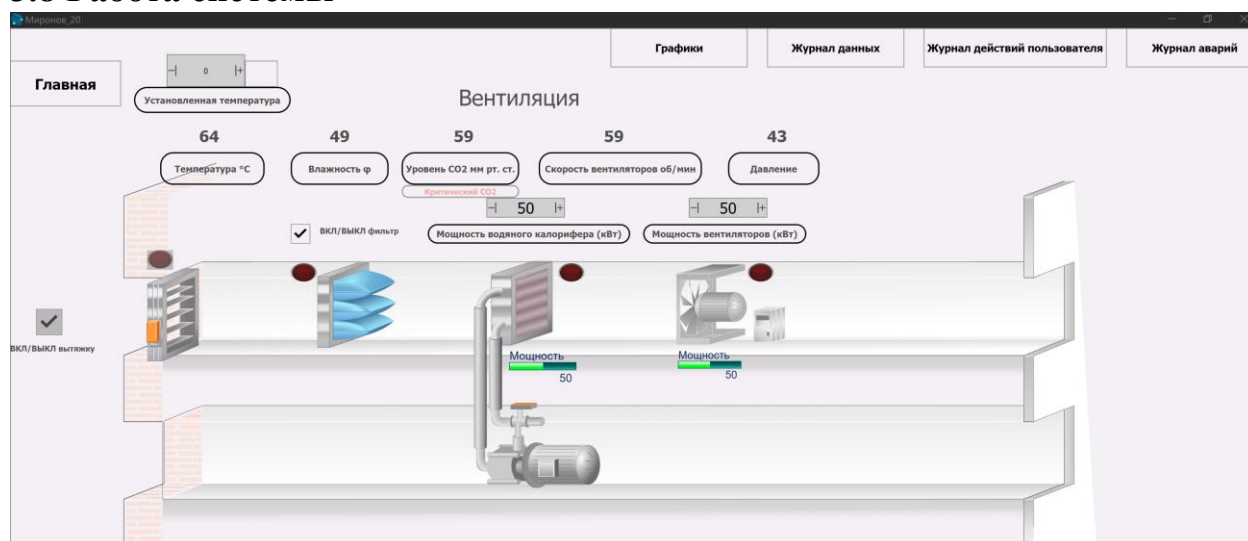


Рисунок 15 - Основной экран системы

Целевая температура: устанавливается пользователем



Рисунок 16 – Установка температуры

Визуальные индикаторы

- Световой индикатор: Авария механизмов
- Предупредительные индикаторы: Параметры подходят к критическим

Звуковые сигналы

- Предупреждение: Включён
- Аварийный сигнал: Включён


Управление и доступ

- Кнопка настроек: Недоступна для текущей роли пользователя (ограничение прав доступа)


Архивный журнал аварий включает следующие колонки (См. Рисунок 21):

- Событие: Указывает на тип события — "Появление" или "Исчезновение" сигнала.
- Время: Дата и время фиксации события.
- Сообщение: Описание события, например, "Нижнее предупредительное", "Нижнее аварийное", "Верхнее предупредительное", "Верхнее аварийное".


- Комментарий




Вентиляция



Фильтр



Водяной калорифер



Вентилятор

Журнал аварий

▼	Событие ▼	Время	Сообщение ▼	Комментарий ▼
info	Исчезновение	08.04.2020 13:38:05	Критическая высокая скорость вдуваемого потока	
	Появление	08.04.2020 13:38:47	Критическая высокая скорость вдуваемого потока	
	Исчезновение	08.04.2020 13:34:43	Критическая высокая скорость вдуваемого потока	
info	Появление	08.04.2020 13:34:35	Критическая высокая скорость вдуваемого потока	
	Исчезновение	08.04.2020 11:52:15	Критическая влажность	
	Появление	08.04.2020 12:32:14	Критическая влажность	
	Восстановление	08.04.2020 12:32:20	Авария в фильтре	208
	Критическая	08.04.2020 13:50:58	Авария в вентиляторе	1
	Исчезновение	08.04.2020 11:52:05	Критическая влажность	
	Восстановление	08.04.2020 11:53:03	Авария в вентиляторе	1
info	Появление	08.04.2020 12:32:00	Критическая влажность	
	Исчезновение	08.04.2020 11:48:58	Авария в вентиляторе	1
	Восстановление	08.04.2020 12:32:58	Авария в водяном калорифере	1
	Исчезновение	08.04.2020 11:51:58	Авария в фильтре	1
	Исчезновение	08.04.2020 11:51:58	Критическая влажность	
	Восстановление	08.04.2020 11:51:58	Авария в вентиляторе	10

Рисунок 17- Архивный журнал событий техпроцесса

Журнал действий пользователя					
Событие ▾	Время ▾	Сообщение ▾	Объект ▾	Пользователь ▾	
Появление	08.04.2025 13:38:20	Управление: Журнал 1	Приточный вентилятор	Winchester	
Появление	08.04.2025 13:37:22	Управление: Журнал	Приточный вентилятор	Winchester	
Исчезновение	08.04.2025 13:35:26	Высокая скорость воздушного потока	Падал		
Появление	08.04.2025 13:35:05	Высокая скорость воздушного потока	Падал		
Исчезновение	08.04.2025 13:35:03	Низкое давление	Падал		
Появление	08.04.2025 13:35:01	Низкое давление	Падал		
Появление	08.04.2025 13:35:00	Управление: Флаг 3	Приточный вентилятор	Winchester	
Появление	08.04.2025 13:35:00	Управление: Числовой ввод 3	Приточный вентилятор	Winchester	
Появление	08.04.2025 13:34:59	Управление: Числовой ввод 3	Приточный вентилятор	Winchester	
Появление	08.04.2025 13:34:59	Управление: Числовой ввод 3	Приточный вентилятор	Winchester	
Появление	08.04.2025 13:34:59	Управление: Числовой ввод 3	Приточный вентилятор	Winchester	
Появление	08.04.2025 13:34:59	Управление: Числовой ввод 3	Приточный вентилятор	Winchester	
Появление	08.04.2025 13:34:59	Управление: Числовой ввод 3	Приточный вентилятор	Winchester	
Появление	08.04.2025 13:34:59	Управление: Числовой ввод 3	Приточный вентилятор	Winchester	
Появление	08.04.2025 13:34:59	Управление: Числовой ввод 3	Приточный вентилятор	Winchester	
Появление	08.04.2025 13:34:59	Низкая температура	Падал		
Появление	08.04.2025 13:34:59	Управление: Числовой ввод 3	Приточный вентилятор	Winchester	
Появление	08.04.2025 13:34:58	Управление: Числовой ввод 3	Приточный вентилятор	Winchester	
Появление	08.04.2025 13:34:58	Управление: Числовой ввод 3	Приточный вентилятор	Winchester	
Появление	08.04.2025 13:34:48	Управление: Флаг 2	Приточный вентилятор	Winchester	
Исчезновение	08.04.2025 13:34:45	Высокая скорость воздушного потока	Падал		
Появление	08.04.2025 13:34:43	Высокая скорость воздушного потока	Падал		
Исчезновение	08.04.2025 13:34:41	Низкое давление	Падал		
Исчезновение	08.04.2025 13:34:35	Высокая скорость воздушного потока	Падал		
Появление	08.04.2025 13:34:35	Низкое давление	Падал		

Рисунок 18 - Архивный журнал действий пользователя

Журнал данных системы					
?	?	?	?	?	?
?	Событие ?	Время ?	Сообщение ?	Значение ?	Приоритет ?
	Исчезновение	08.04.2025 13:35:26	Высокая скорость воздушного потока	100	504
КНИТ	Появление	08.04.2025 13:35:05	Высокая скорость воздушного потока	100	504
	Исчезновение	08.04.2025 13:35:03	Низкое давление	29	505
КНИТ	Появление	08.04.2025 13:35:01	Низкое давление	29	505
КНИТ	Появление	08.04.2025 13:34:59	Низкая температура	26	501
	Исчезновение	08.04.2025 13:34:45	Высокая скорость воздушного потока	100	504
КНИТ	Появление	08.04.2025 13:34:43	Высокая скорость воздушного потока	100	504
	Исчезновение	08.04.2025 13:34:41	Низкое давление	29	505
	Исчезновение	08.04.2025 13:34:35	Высокая скорость воздушного потока	94	504
КНИТ	Появление	08.04.2025 13:34:35	Низкое давление	29	505
	Исчезновение	08.04.2025 13:34:29	Низкое давление	29	505
КНИТ	Появление	08.04.2025 13:34:27	Низкое давление	29	505
КНИТ	Появление	08.04.2025 13:34:25	Высокая скорость воздушного потока	94	504
	Исчезновение	08.04.2025 13:34:23	Высокая скорость воздушного потока	93	504
КНИТ	Появление	08.04.2025 13:34:18	Высокая скорость воздушного потока	93	504
	Исчезновение	08.04.2025 13:34:09	Низкое давление	28	505
КНИТ	Появление	08.04.2025 13:34:04	Низкое давление	28	505
	Исчезновение	08.04.2025 13:34:02	Низкое давление	27	505
	Исчезновение	08.04.2025 13:34:01	Высокая температура	92	501
КНИТ	Появление	08.04.2025 13:34:01	Низкое давление	27	505
	Исчезновение	08.04.2025 13:33:59	Низкое давление	29	505
КНИТ	Появление	08.04.2025 13:33:57	Низкое давление	29	505
КНИТ	Появление	08.04.2025 13:33:55	Высокая температура	92	501
	Исчезновение	08.04.2025 13:33:23	Низкое содержание CO2	0	504
	Исчезновение	08.04.2025 13:33:23	Низкая влажность	0	502

Рисунок 19 - Архивный журнал данных



Рисунок 20 – Архивные графики данных системы

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы была разработана система управления для мониторинга и контроля работы системы вентиляции в здании на основе SCADA-системы. Для реализации был использован сервер Modbus-TCP, который обеспечивал передачу данных о параметрах вентиляции, таких как температура, влажность, уровень CO₂, скорость воздушного потока, давление и другие критически важные показатели.

Основной целью проекта было создание системы, которая не только осуществляет мониторинг работы вентиляционного оборудования, но и предоставляет возможность его управления. В рамках решения были реализованы функциональные возможности регулирования скорости вентиляторов, контроля открытия и закрытия клапанов, а также настройки расхода воздуха для различных зон здания.

Особое внимание уделено обеспечению безопасности работы системы. Для каждого отслеживаемого параметра были настроены предупредительные и аварийные уставки, а также реализована система оповещений с выводом аварийных сообщений на экран АРМ оператора, сопровождаемых звуковой сигнализацией. В случае обнаружения аварийных ситуаций, таких как отказ вентилятора или некорректная работа системы, SCADA-система оперативно уведомляет оператора и предпринимает необходимые меры для восстановления нормальных условий работы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. MasterSCADA 4D. Руководство пользователя. [Электронный ресурс]. – Доступно по ссылке:
<https://www.reallab.ru/images/editor/downloads/software/MasterSCADA%204D/MasterSCADA-4D.pdf> (дата обращения: 16.01.2025).
2. Stack Overflow. Русскоязычное сообщество разработчиков. [Электронный ресурс]. – Доступно по ссылке: <https://ru.stackoverflow.com/> (дата обращения: 16.01.2025).
3. PLCAutomations. Канал на YouTube, посвящённый автоматизации. [Электронный ресурс]. – Доступно по ссылке: <https://www.youtube.com/@PLCAutomations> (дата обращения: 16.01.2025).
4. MasterSCADA. Официальный сайт. [Электронный ресурс]. – Доступно по ссылке: <https://masterscada.ru/> (дата обращения: 16.01.2025).