

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
« ___ » _____ 2024 г.

Разработка библиотеки визуальных компонент мнемосхем и графиков
осциллограмм модуля конфигурации и мониторинга параметров
программного комплекса для производства следящих гидроприводов
с гидростатическими направляющими

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Руководитель работы,
к.т.н., доцент каф. ЭВМ
_____ Д.В. Топольский
« ___ » _____ 2024 г.

Автор работы,
студент группы КЭ-405
_____ Б.Д. Укурчев
« ___ » _____ 2024 г.

Нормоконтролёр,
ст. преп. каф. ЭВМ
_____ С.В. Сяськов
« ___ » _____ 2024 г.

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

«__» _____ 2024 г.

ЗАДАНИЕ

на выпускную квалификационную работу бакалавра

студенту группы КЭ-405

Укурчееву Богдану Дмитриевичу

обучающемуся по направлению

09.03.01 «Информатика и вычислительная техника»

1. Тема работы: «Разработка библиотеки визуальных компонент мнемосхем и графиков осциллограмм модуля конфигурации и мониторинга параметров программного комплекса для производства следящих гидроприводов с гидростатическими направляющими»
утверждена приказом по университету от «__» _____ 2024 г. № _____

2. Срок сдачи студентом законченной работы: 4 июня 2024 г.

3. Исходные данные к работе:

Требования к функционалу разрабатываемого приложения:

- Интерактивный интерфейс,
- Авторизация по паролю,

- Возможность выбрать различную конфигурацию оборудования,
- Язык разработки: C#,
- платформа .NET 8,
- Среда разработки Visual Studio Community.
- операционная система: Windows 10 и выше, Linux Debian 12 и выше, Astra Linux 1.7.3 и выше;

4. Перечень подлежащих разработке вопросов:

1. Обзор литературы;
2. Определение требований;
3. Реализация системы;
4. Тестирование;

Дата выдачи задания: 1 декабря 2023 г.

Руководитель работы _____ / Д.В. Топольский/

Студент _____ / Б.Д. Укурчеев/

Календарный план

Этап	Срок сдачи	Подпись руководителя
Обзор литературы	10.03.2024	
Определение требований	21.03.2024	
Реализация системы	04.04.2024	
Тестирование	25.04.2024	
Компоновка текста работы и сдача на нормоконтроль	16.05.2024	
Подготовка презентации и доклада	24.05.2024	

Руководитель работы _____ / Д.В. Топольский/

Студент _____ / Б.Д. Укурчиев/

Аннотация

Б.Д.Укурчиев. «Разработка кроссплатформенного приложения для мониторинга и конфигурации параметров программного комплекса, для производства следящих гидроприводов с гидростатическими направляющими». – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2024, 36 с., библиогр. список – 15 наим.

В рамках выпускной квалификационной работы производится обзор, анализ, выявление плюсов и минусов существующий программно-аппаратных решений. Обзор показал, что большинство аналогов не являются кроссплатформенными, также, все рассмотренные аналоги представляют собой коммерческие продукты.

Разработано программное обеспечение для мониторинга и конфигурации параметров, для производства следящих гидроприводов с гидростатическими направляющими, для внедрения его в систему предприятия.

Результаты тестирования подтвердили соответствие программы установленным требованиям.

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ.....	7
ВВЕДЕНИЕ.....	8
1. ОБЗОР ЛИТЕРАТУРЫ.....	9
1.1. Анализ основных технологических решений.....	9
1.1.1. Язык программирования С#.....	9
1.1.2. Платформа .Net.....	9
1.1.3. Среда разработки Visual Studio.....	11
1.1.4. Дополнительное программное обеспечение.....	11
1.2. Обзор аналогов.....	12
2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ.....	20
3. РЕАЛИЗАЦИЯ СИСТЕМЫ.....	22
4. ТЕСТИРОВАНИЕ.....	25
ЗАКЛЮЧЕНИЕ.....	26
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	27
ПРИЛОЖЕНИЕ А.....	28

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

СГ – следящий гидропривод

ГСН – гидростатические направляющие

CLR - Common Language Runtime

USB - Universal Serial Bus

XAML - eXtensible Application Markup Language

ВВЕДЕНИЕ

Современные производственные системы все более интегрируют в себя высокотехнологичные управляющие комплексы, предназначенные для обеспечения эффективности и надежности процессов производства. Одним из ключевых элементов таких комплексов являются гидроприводы с гидростатическими направляющими, обеспечивающие точное и стабильное движение в различных машинах и механизмах.

Для обеспечения оптимального функционирования и контроля параметров гидроприводов необходимо иметь специализированные программные средства, позволяющие не только осуществлять конфигурацию системы, но и проводить мониторинг ключевых параметров в реальном времени. В этом контексте возникает актуальная задача разработки библиотеки визуальных компонент мнемосхем и графиков осциллограмм для модуля конфигурации и мониторинга параметров программного комплекса, предназначенного для производства следящих гидроприводов.

Целью данного исследования является разработка инструментария, который позволит оперативно и эффективно настраивать параметры гидроприводов, а также осуществлять визуализацию и анализ ключевых показателей их работы. Чтобы достичь этой цели, требуется решить следующие задачи:

Анализ требований к функциональности и интерфейсу модуля конфигурации и мониторинга параметров гидроприводов.

Проектирование архитектуры библиотеки визуальных компонент, учитывающей специфику работы гидроприводов и требования к пользовательскому опыту.

Разработка и реализация компонентов для конфигурации и визуализации параметров гидроприводов.

Тестирование и оптимизация разработанных компонентов для обеспечения их надежной работы в различных условиях эксплуатации.

1. ОБЗОР ЛИТЕРАТУРЫ

1.1. Анализ основных технологических решений

1.1.1. Язык программирования C#

Программное обеспечение было разработано с использованием языка программирования C#. Его объектно-ориентированный дизайн делает его подходящим для создания сложных программ и облегчает обслуживания кода.

Основные характеристики и возможности C# включают в себя:

- объектно-ориентированный подход;
- мощная стандартная библиотека;
- мультипарадигменность;
- управляемый язык;
- интеграция с другими технологиями Microsoft.

C# — это язык программирования, который охватывает основные концепции объектно-ориентированного программирования, таких как наследование, инкапсуляция и полиморфизм. Эта конструкция дает разработчикам возможность создавать структурированные и масштабируемые приложения.

У C# есть поддержка различных парадигм программирования, таких как:

- процедурное программирование;
- объектно-ориентированное программирование;
- функциональное программирование;
- аспектно-ориентированное программирование.

Благодаря этому C# может решать большое количество задач [1, 2].

1.1.2. Платформа .Net

Платформа .Net, разработанная компанией Microsoft, предназначена для создания программного обеспечения. Она содержит множество инструментов, библиотек и технологий для разработки и запуска разнообразного

программного обеспечения. Например, для работы с .NET разработчики могут использовать такие среды как Visual Studio или Visual Studio Code, чтобы создавать, отлаживать и управлять своими проектами. У платформы есть поддержка нескольких языков программирования, в том числе: C#, F#, VB.NET.

В .NET есть возможность реализовывать приложения используя управляемый код, который в том числе поддерживается C#. Такие приложения управляются общезыковой средой CLR (Common Language Runtime), которая загружает приложение и в случае необходимости освобождает память. Такой код позволяет, среде выполнения в любой момент остановить выполнение программы и получить информацию о текущем состоянии. Существуют и приложения, которые сразу переводятся в машинный код. Такой код называется неуправляемым. Платформа .NET также предоставляет возможности для взаимодействия с ним [3, 4, 15].

Для разработки программного обеспечения используется версия .NET 8, она представляет собой одну из последних и наиболее актуальных версий на момент написания. В данной версии были проведены улучшения производительности в таких областях как:

- ускорение загрузки приложений;
- оптимизация работы с памятью;
- улучшения в JIT-компиляторе и т. д.

.NET 8 предоставляет обновленные версии SDK и инструментов разработки. Также эта версия вводит поддержку C# 11, что позволяет использовать новые функции языка. Предоставляет новые возможности и улучшения в Entity Framework Core, что упрощает работу с базами данных [14].

1.1.3. Среда разработки Visual Studio

Visual Studio, выбранная в качестве интегрированной среды разработки, разработана компанией Microsoft для создания разнообразных приложений. Предоставляет большой набор инструментов для разработки, среди которых: редактор кода с подсветкой синтаксиса и автозавершением, отладчик, систему контроля версий, инструменты для создания пользовательских интерфейсов, тестирования и многое другое [5].

1.1.4. Дополнительное программное обеспечение

Для отрисовки графического интерфейса был использован Avalonia UI. Это кроссплатформенный фреймворк пользовательского интерфейса для языка C#, который предоставляет возможность создавать десктопные приложения. Он основан на платформе .NET и использует язык XAML, который даёт возможность разделения кода на две части, в одной содержится дизайн, в другой бизнес-логика. Также, эта библиотека кроссплатформенная. Avalonia включает в себя широкий спектр predefined элементов управления. Это обеспечивает гибкость при проектировании различных типов пользовательских интерфейсов. Avalonia разрабатывается под открытой лицензией (MIT) и имеет активное сообщество разработчиков, которые постоянно вносят вклад в проект, предоставляют поддержку и создают новые инструменты и расширения. [6].

Библиотека ScottPlot – это библиотека для построения графиков на платформе .NET, специально предназначенная для быстрого и удобного создания высококачественных графиков. Включая линейные графики, точечные графики, столбчатые диаграммы, гистограммы и многое другое. Библиотека предоставляет широкий спектр параметров настройки, позволяя пользователям настраивать внешний вид графиков в соответствии с их потребностями и предпочтениями. ScottPlot оптимизирована для обеспечения высокой производительности даже при работе с большими объемами данных. Она стремится обеспечить быстрое отображение графиков и отзывчивость

интерфейса. Библиотека поддерживает большое количество типов графиков, что позволяет создавать разнообразные визуализации данных в зависимости от задачи. ScottPlot написана на языке программирования C# и предназначена для использования с платформой .NET, что полностью соответствует нашим требованиям [11].

Библиотека JetBrains Annotations — это набор атрибутов и аннотаций, разработанных компанией JetBrains, которые позволяют разработчикам добавлять дополнительную информацию о коде в проектах на языках программирования, таких как Java, Kotlin и C#. Эта информация используется как компилятором, так и статическим анализатором для проверки корректности кода и выявления потенциальных проблем [12].

LibUsbDotNet – это библиотека для работы с USB устройствами на платформе .NET. Она предоставляет программистам возможность взаимодействия с USB устройствами напрямую из кода на C# или других языках .NET, обеспечивая простой и удобный интерфейс для работы с USB-портами. Благодаря низкоуровневому доступу к USB-портам, есть возможность полностью контролировать взаимодействие с USB устройствами. Это позволяет реализовывать различные функции, такие как отправка и прием данных, управление конфигурациями устройств. Библиотека является кроссплатформенной. Это делает ее универсальным инструментом для работы с USB устройствами на различных платформах. Библиотека имеет открытый исходный код и поддерживает устройства с различными версиями USB, включая USB 1.x, USB 2.0 и USB 3.x [13].

1.2. Обзор аналогов

В настоящее время, из-за импортозамещения, требуются отечественные разработки зарубежных аналогов. В том числе системы для управления, настройки и диагностики следящего гидропривода с гидростатическими направляющими. Так как программный комплекс необходимо регулярно обновлять и дорабатывать. Для повышения производительности данного

процесса решено разработать систему диагностики и управления с графическим интерфейсом. Разрабатываемая система будет лучше других аналогов в плане производительности, удобства и простоты обслуживания, так как заточена под определённое оборудование и разработка, и доработка ПО ведётся непосредственно на предприятии. Существуют заграничные аналоги, такие. Однако из-за того, что данное ПО стремится решать целый комплекс задач, не все из которых требуются для определённого производства. Также решение технических проблем упростится, так как техподдержка находится в непосредственной близости. Потенциальными заказчиками могут быть производители, которые ранее эксплуатировали сервисы заграничных компаний, но более не могут прибегать к их услугам, в том числе из-за импортозамещения. Ниже представлено программное обеспечение заграничных компаний, плюсы и минусы систем и их описание.

Hydraulic System Monitoring and Diagnostics Software, разработанное итальянской компанией Atos, представляет собой специализированный инструмент, предназначенный для постоянного контроля состояния и производительности гидравлических систем [7]. Предлагая возможности мониторинга в режиме реального времени, это программное обеспечение позволяет операторам и инженерам наблюдать за работой гидравлической системы и оперативно диагностировать неполадки, способствуя своевременному вмешательству для устранения неисправностей.

Преимущества:

- предотвращение аварийных ситуаций;
- увеличение надежности и производительности;
- экономия времени и ресурсов;
- оптимизация обслуживания и ремонта;
- интеграция с другими системами.

Программное обеспечение позволяет операторам на ранней стадии выявлять потенциальные проблемы и неисправности в гидравлической

системе, помогая предотвращать аварийные ситуации и сводя к минимуму риск дорогостоящего ремонта или простоя в работе.

Благодаря постоянному мониторингу и анализу данных программное обеспечение облегчает поддержание оптимального состояния гидравлических систем, тем самым повышая их надежность и производительность.

Автоматизированная диагностика и анализ данных сокращают необходимость в ручной проверке и техническом обслуживании системы, что приводит к экономии времени и ресурсов операторов и инженеров.

Кроме того, программное обеспечение позволяет получать информацию о состоянии компонентов системы, что позволяет оптимизировать графики технического обслуживания и ремонта, тем самым предотвращая ненужные расходы.

Кроме того, совместимость системы с другими системами позволяет осуществлять всесторонний обзор всего производственного процесса, повышая эффективность работы.

Недостатки:

- сложность внедрения;
- необходимость квалифицированного персонала;
- ограничения совместимости и интеграции;
- зависимость от технологий и поставщиков.

Внедрение и поддержка системы сопряжены со значительными затратами, включая закупку программного обеспечения, обучение персонала и адаптацию к существующей инфраструктуре.

Достижение оптимального использования зависит от наличия опытных экспертов, способных анализировать данные, интерпретировать результаты и принимать обоснованные решения.

Проблемы несовместимости или интеграции с существующими системами управления и мониторинга могут привести к дополнительным

препятствиям и расходам при корректировке или замене существующих решений.

Внедрение программного обеспечения для мониторинга и диагностики делает предприятие зависимым от технологий и поставщиков, что создает потенциальные бизнес-риски в случае сбоев в работе системы или изменений в поддержке поставщиков.

Bosch Rexroth HMI-Software

Немецкая фирма Bosch Rexroth предлагает программное обеспечение HMI (человеко-машинный интерфейс), предназначенное для предоставления операторам интуитивно понятных и эффективных интерфейсов для контроля и управления промышленным оборудованием [8]. Это позволяет интегрировать различные элементы управления, визуализации данных, графику и мультимедийные элементы для обеспечения наилучшего взаимодействия с пользователем. Программное обеспечение предназначено для установки на промышленном контроллере или панели оператора, которые обычно работают на таких платформах, как Windows Embedded или специализированные ОСРВ (операционные системы реального времени). Это не универсальное решение; скорее всего, он оптимизирован для конкретных устройств и платформ.

Преимущества:

- собственная экосистема;
- интуитивный интерфейс;
- гибкие возможности настройки;
- поддержка различных типов данных;
- масштабируемость.

Это программное обеспечение разработано таким образом, чтобы легко интегрироваться с оборудованием Bosch Rexroth, обеспечивая легкую совместимость и интеграцию.

Система упрощает взаимодействие оператора с оборудованием, позволяя создавать интуитивно понятные интерфейсы.

Широкий набор инструментов облегчает настройку интерфейса в соответствии с конкретными требованиями и задачами пользователя.

Программное обеспечение позволяет визуализировать различные типы данных, включая показатели производительности оборудования и параметры процесса.

Оно поддерживает создание интерфейсов для широкого спектра промышленного оборудования и систем автоматизации.

Недостатки:

- зависимость от экосистемы Bosh Rexroth;
- сложность обучения.

Поскольку программное обеспечение разработано специально для оборудования Bosch Rexroth, оно может создавать ограничения для пользователей, предпочитающих использовать оборудование других брендов.

Освоение всех аспектов обширных функций и возможностей программного обеспечения может потребовать дополнительного времени и обучения.

Eaton Pro-FX Software Suite

Программный комплекс Pro-FX, разработанный американской инжиниринговой фирмой Eaton, предлагает широкий набор инструментов для проектирования и настройки систем управления [9]. Его функции включают графический интерфейс для создания логики управления, настройки устройств ввода-вывода, мониторинга и диагностики системы, а также для создания пользовательских интерфейсов. Как правило, он работает на компьютерах под управлением операционных систем Windows и несовместим с другими платформами.

Преимущества:

- графический интерфейс;
- поддержка гибридных систем;
- обширная документация и поддержка.

Программное обеспечение имеет удобный графический интерфейс, предназначенный для разработки логики управления и конфигурации устройства, упрощения процесса программирования и повышения доступности для пользователей.

Пользователи могут создавать гибридные системы управления, интегрирующиеся с гидравлическими и электрическими компонентами, что обеспечивает повышенную гибкость при проектировании и настройке системы.

Кроме того, пакет включает в себя исчерпывающую документацию, учебные ресурсы и техническую поддержку, что упрощает процесс обучения и обеспечивает эффективное использование программного обеспечения.

Недостатки:

- сложность в освоении;
- проблемы с совместимостью.

Учитывая обширный функционал и возможности, освоение и эффективное использование программного пакета может оказаться сложной задачей для пользователей, не имеющих достаточного опыта и подготовки.

Совместимость программного обеспечения и возможности интеграции с устройствами и оборудованием других производителей несколько ограничены.

Parker IQAN Software

Американская корпорация Parker, известная своими технологиями управления, предлагает программные средства IQAN, предназначенные для настройки программного обеспечения систем управления [10]. Эти инструменты включают в себя такие функции, как создание графического интерфейса пользователя, конфигурирование логики управления, мониторинг

и диагностика состояния системы. Программное обеспечение IQAN совместимо с компьютерами Windows. Кроме того, Parker предоставляет мобильное приложение IQAN, доступное на мобильных устройствах iOS и Android, позволяющее пользователям отслеживать гидравлические системы и управлять ими. Это обеспечивает высокий уровень кроссплатформенной совместимости для некоторых функций.

Преимущества:

- собственная экосистема;
- открытая архитектура;
- мониторинг и диагностика;
- документация и поддержка.

Программное обеспечение IQAN предлагает единую платформу для разработки, программирования и диагностики гидравлических и электронных систем управления, упрощая разработку и экономя время.

В отличие от альтернативных решений, программное обеспечение IQAN обеспечивает универсальность интеграции с различным оборудованием и системами.

Отслеживание рабочих данных в режиме реального времени упрощает управление оборудованием и помогает быстро выявлять проблемы.

Кроме того, обширная документация, учебные ресурсы и техническая поддержка упрощают процесс обучения и обеспечивают эффективное использование программного обеспечения.

Недостатки:

- сложность в освоении;
- проблемы с совместимостью.

Учитывая широкий спектр функций и возможностей программного обеспечения IQAN, для овладения навыками использования потребуются определённая подготовка.

Совместимость и интеграция программного обеспечения с оборудованием и системами других производителей могут быть несколько ограниченными.

Выводы по первой главе:

В этой главе были рассмотрены предметная область, а также проведен обзор существующих аналогов. Существующие решения имеют общую проблему с кроссплатформенностью, а также у всех систем не открытый исходный код.

Разрабатываемая библиотека должна охватывать имеющиеся у аналогов достоинства и устранять недостатки. Она должна быть надежной и легко расширяемой. Должна иметь удобный и понятный интерфейс взаимодействия. Размер библиотеки должен быть компактным, что позволит легко разобраться в ее исходном коде, а также расширять ее и делать лучше.

2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

Функциональные требования — это спецификации того, что система должна делать, описывая функции, сервисы и процессы, которые должны быть доступны в программном обеспечении или системе. Они определяют, какие возможности предоставляются системой, как эти возможности будут работать и каким образом пользователи будут взаимодействовать с системой. Функциональные требования:

- вход для пользователя с помощью пароля;
- управление СГ с ГСН в режиме реального времени через локальное (USB) и сетевое (Ethernet) подключение к ЭГУМ;
- сообщения об ошибках СГ с ГСН;
- возможность просмотра в реальном времени сигналов задания и значения с датчиков;
- Ведение журнала ошибок.

Нефункциональные требования – требования, которые определяют аспекты системы, которые не связаны напрямую с её функциональностью, но влияют на её производительность, безопасность, удобство использования и другие характеристики.

Нефункциональные требования:

- возможность масштабирования;
- обработка большого количества данных;
- читаемый интерфейс. Текст не должен сливаться с фоном. Графики должны быть чётко видны;
- поддерживаемые ОС: Windows 10 и выше, Linux Debian 12 и выше, Astra Linux 1.7.3 и выше;
- язык разработки: C#;
- платформа: .NET 8 актуальной версии;

- всё программное обеспечение, необходимое для функционирования программного комплекса – свободное;
- предусматривается вывод данных в реальном времени.

Вывод по второй главе:

Во второй главе были определены функциональные и нефункциональные требования для разрабатываемой системы.

3. РЕАЛИЗАЦИЯ СИСТЕМЫ

Для реализации программной части системы был выбран язык программирования C#. Разработка велась в кроссплатформенной интегрированной среде программирования Visual Studio Community. В качестве основной библиотеки графических компонент использовалась Avalonia UI.

При открытии программы открывается стартовое окно, продемонстрированное на рисунке 1. На нём оператор может выбрать оборудование, с которым будет работать далее. В нижней части окна находится кнопка, с помощью которой можно переключить «ON LINE» и «OFF LINE» режимы. Следом идёт выпадающий список, с возможностью выбора уровня доступа. При выборе какого-либо уровня доступа требуется ввести пароль. И далее расположен ещё один раскрывающийся список. В нём предлагается выбрать способ подключения.

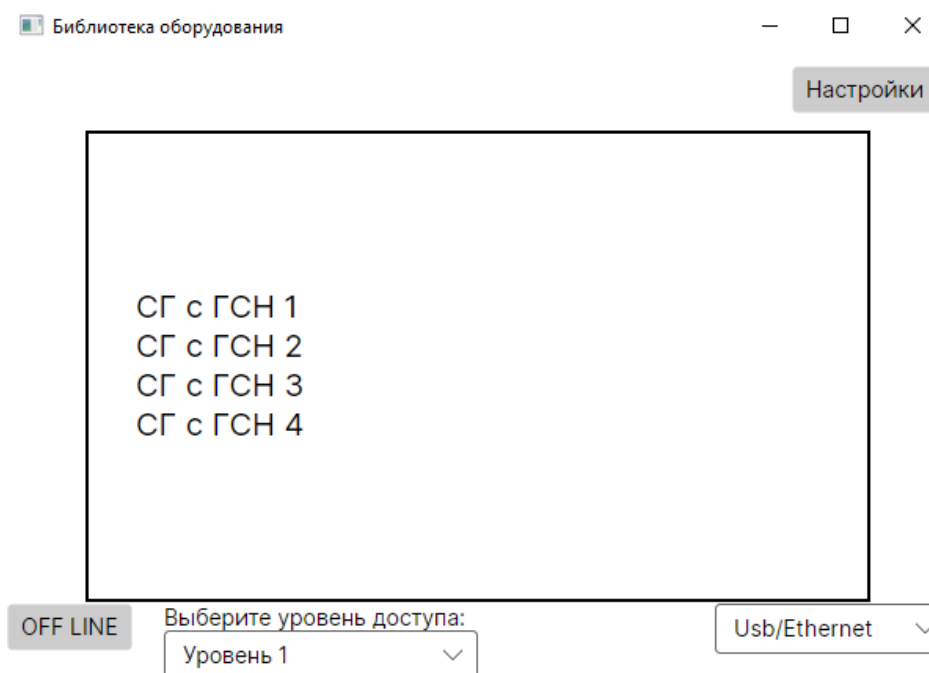


Рисунок 1 – Экранная форма

При вводе пароля, закрывается стартовое и открывается главное окно (рисунок 2).

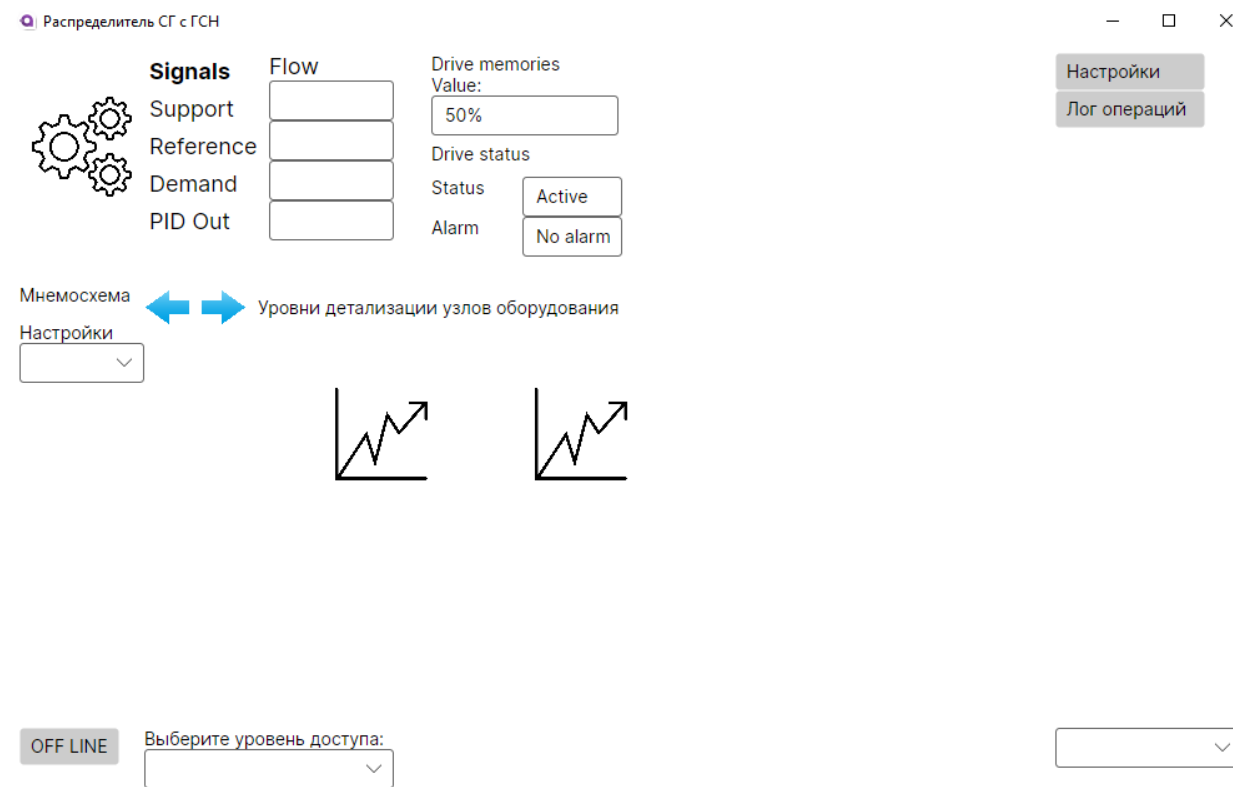


Рисунок 2 – Экранная форма главного окна

Здесь, в верхнем левом углу находится изображение выбранного элемента оборудования. Далее идут блоки: основных настроек, вывода состояния и вывода ошибок. Также благодаря настройкам в левой части экрана можно настраивать PID регуляторы. Нижняя часть окна соответствует стартовому экрану. Стрелки влево и вправо переключают между уровнями детализации. На мнемосхеме отображаются блоки, при нажатии на которые, также выводится более подробная информация (Рисунок 3).

← → Уровни детализации узлов оборудования

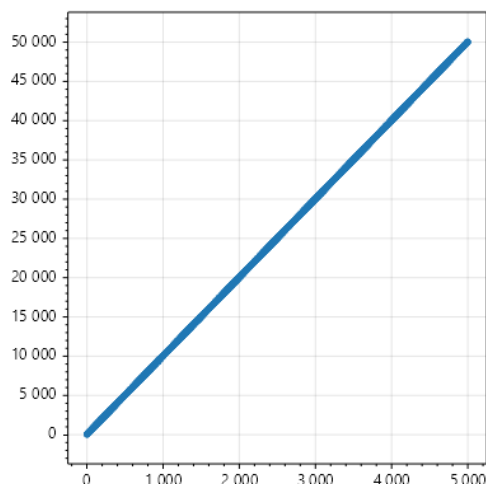


Рисунок 3 – Экранная форма уровней детализации

Как и на стартовом экране, на главном экране присутствует кнопка настроек модуля. При нажатии на неё открывается новое окно (Рисунок 4).

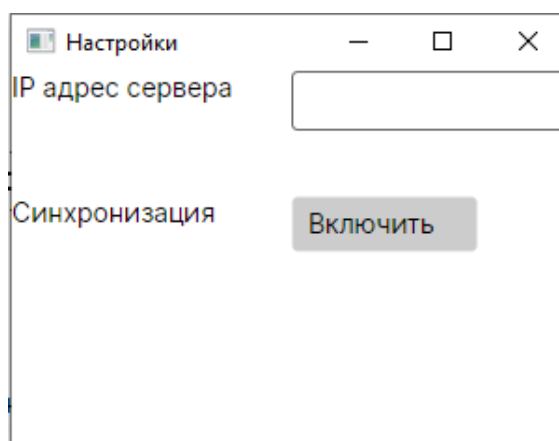


Рисунок 4 – Экранная форма настроек

Вывод по третьей главе:

В рамках данной главы разработано приложение для мониторинга и настройки, с реализацией всех требований, приведённых во второй главе.

4. ТЕСТИРОВАНИЕ

Требования, определённые во второй главе, были полностью соблюдены, что можно увидеть в таблице 1.

Таблица 1 – Таблица сравнения поставленных требований и реализации

Требования	Реализация
Вход для пользователя с помощью пароля.	Вход осуществляется по паролю, есть возможность выбора уровня доступа.
Управление СГ с ГСН в режиме реального времени через локальное (USB) и сетевое (Ethernet) подключение к ЭГУМ.	В программе реализовано управление СГ с ГСН в режиме реального времени.
Сообщения об ошибках СГ с ГСН.	Система предусматривает вывод сообщения об ошибках.
Возможность просмотра в реальном времени сигналов задания и значения с датчиков.	Данные с датчиков считываются в реальном времени и выводятся в виде графиков.
Ведение журнала ошибок.	Есть возможность сохранения данных о произошедших ошибках.

По таблице 1 видно, что все требования были реализованы.

Вывод по разделу 4:

Тестирование показало, что программа соответствует требованиям.

ЗАКЛЮЧЕНИЕ

В ходе выпускной квалификационной работы был проведен анализ аналогов, однако все рассмотренные решения являются коммерческими и не обладают кроссплатформенностью. Также были изучены технологические решения. Были рассмотрены этапы проектирования и разработки. Приведены задачи, которые должны решаться разрабатываемым программным обеспечением.

В итоге было получено:

- вход в приложение осуществляется по паролю, есть возможность выбора уровня доступа;
- в программе реализовано управление СГ с ГСН в режиме реального времени;
- система предусматривает вывод сообщения об ошибках;
- данные с датчиков считываются в реальном времени и выводятся в виде графиков;
- есть возможность сохранения данных о произошедших ошибках.

Произведено тестирование, результаты которого подтвердили, что программа полностью соответствует установленным требованиям.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Руководство по C# - управляемый язык .NET – <https://learn.microsoft.com/ru-ru/dotnet/csharp/>
2. Шилдт, Г. Полный справочник C# / Г. Шилдт - Издательский дом "Вильямс", 2004. - 752 с.
3. Введение в .NET – <https://learn.microsoft.com/ru-ru/dotnet/core/introduction>
4. Metzgar, D. .NET Core in Action / D. Metzgar - Manning Publications, 2018. - 288 p.
5. Документация по семейству продуктов Visual Studio – <https://learn.microsoft.com/ru-ru/visualstudio/>
6. Документация - Avalonia UI – <https://avaloniaui.homesoft.ru/docs/>
7. ПО Русский филиал компании АТОС – <http://www.atos-rf.ru/>
8. Bosh Rexroth Russia – <https://boschrexroth-russia.ru>
9. Pro-FX software – <https://www.eaton.com/gb/en-gb/catalog/electronic-controls-and-software/pro-fx-software-electronic-controls-and-software-eaton.resources.html#tab-3>
10. Mobile Controller Software – <https://ph.parker.com/us/en/product-list/iqan-software>
11. ScottPlot - Interactivity Plotting Library for .NET – <https://scottplot.net>
12. Use annotations to refine code inspection – https://www.jetbrains.com/help/resharper/Code_Analysis_Code_Annotations.html
13. LibUsbDotNet 2.2.8 - Table of Content – <https://libusbdotnet.sourceforge.net/V2/Index.html>
14. Новые возможности .NET 8 – <https://learn.microsoft.com/ru-ru/dotnet/core/whats-new/dotnet-8/overview>
15. Новые возможности C# 11 – <https://learn.microsoft.com/ru-ru/dotnet/csharp/whats-new/csharp-11>

ПРИЛОЖЕНИЕ А

Листинг А.1.1 - Исходный код

```
<Window xmlns="https://github.com/avaloniaui"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:vm="using:Project.ViewModels"
  xmlns:views="using:Project.Views"
  x:Class="Project.Views.StartWindow"
  x:DataType="vm:MainWindowViewModel"
  Title="Библиотека оборудования" Width="600" Height="400">
  <Grid RowDefinitions="50, Auto, 60">
    <Grid Grid.Row="0" ColumnDefinitions="*, 100">
      <Grid Grid.Column="1">
        <Button Click="Click_OpenSettings"
Width="120">Настройки</Button>
      </Grid>
    </Grid>
    <Grid Grid.Row="1">
      <Border BorderBrush="Black" BorderThickness="2" Width="500"
Height="290">
        <ScrollViewer>
          <StackPanel Margin="30 0"
VerticalAlignment="Center">
            <TextBlock FontSize="20">СГ с ГСН
1</TextBlock>
            <TextBlock FontSize="20">СГ с ГСН
2</TextBlock>
            <TextBlock FontSize="20">СГ с ГСН
3</TextBlock>
            <TextBlock FontSize="20">СГ с ГСН
4</TextBlock>
          </StackPanel>
        </ScrollViewer>
      </Border>
    </Grid>
    <Grid Grid.Row="2" ColumnDefinitions="100, 3.5*, *">
      <StackPanel Grid.Column="0">
        <Button Click="Click_OnOffLine" Margin="10, 12" Width="80"
Height="30">OFF LINE</Button>
      </StackPanel>
      <StackPanel Grid.Column="1">
        <TextBlock Text="Выберите уровень доступа:"/>
        <Menu DockPanel.Dock="Bottom">
          <MenuItem Header="Уровень">
            <MenuItem Header="Уровень 1"
Command="{Binding OpenPassword}"/>
            <MenuItem Header="Уровень 2"
Command="{Binding OpenPassword}"/>
            <MenuItem Header="Уровень 3"
Command="{Binding OpenPassword}"/>
          </MenuItem>
        </Menu>
      </StackPanel>
      <StackPanel Grid.Column="2">
        <ComboBox x:Name="usb" VerticalAlignment="Bottom"
HorizontalAlignment="Right" Width="150"/>
      </StackPanel>
    </Grid>
  </Grid>
</Window>
```

Листинг А.1.2

```
using Avalonia.Collections;
using Avalonia.Controls;
using Avalonia.Interactivity;

namespace Project.Views
{
    public partial class StartWindow : Window
    {
        public StartWindow()
        {
            InitializeComponent();
            var password = this.Find<ComboBox>("password");
            var usb = this.Find<ComboBox>("usb");

            usb.ItemsSource = new AvaloniaList<object>
            {
                "Usb/Ethernet",
                "...",
            };

            // Обработчик события при выборе элемента
            usb.SelectionChanged += ComboBox_SelectionChanged;
        }

        public void Click_OpenSettings(object sender, RoutedEventArgs args)
        {
            var window = new Settings();
            window.Show();
        }

        public void Click_OnOffLine(object sender, RoutedEventArgs args)
        {
            (sender as Button)!.Content = "ON LINE";
        }

        private void ComboBox_SelectionChanged(object? sender,
        SelectionChangedEventArgs e)
        {
            var comboBox = (ComboBox)sender!;
            if (comboBox.SelectedIndex != -1)
            {
                var selectedItem = comboBox.SelectedItem;
                // Действия при выборе элемента из списка
            }
        }
    }
}
```

Листинг А.2.1 – Исходный код

```
<Window xmlns="https://github.com/avaloniaui"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:vm="using:Project.ViewModels"
xmlns:views="using:Project.Views"
x:DataType="vm:MainWindowViewModel"
x:Class="Project.Views.Password"
Title="Ввод пароля" Width="200" Height="150">
<Grid RowDefinitions="2*, 2*, *">
    <TextBlock Grid.Row="0" HorizontalAlignment="Center"
VerticalAlignment="Bottom">
```

```

        Пароль :
        </TextBlock>
        <TextBox Grid.Row="1" PasswordChar="*" VerticalAlignment="Top"
Width="120">
        </TextBox>
        <Button Grid.Row="2" Click="OpenMainWindow" HorizontalAlignment="Right"
Width="80" Height="30" >
        Ввести
        </Button>
    </Grid>
</Window>

```

Листинг А.2.2

```

using Avalonia.Controls;
using Avalonia.Interactivity;

namespace Project.Views
{
    public partial class Password : Window
    {
        public Password()
        {
            InitializeComponent();
        }

        public void OpenMainWindow(object sender, RoutedEventArgs args)
        {
            var window = new MainWindow();
            var window2 = new Password();
            var window3 = new StartWindow();

            window2.Close();
            window3.Close();
            window.Show();
        }
    }
}

```

Листинг А.3.1

```

<Window xmlns="https://github.com/avaloniaui"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:vm="using:Project.ViewModels"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:ScottPlot="clr-
namespace:ScottPlot.Avalonia;assembly=ScottPlot.Avalonia"
xmlns:views="using:Project.Views"
mc:Ignorable="d"
x:Class="Project.Views.MainWindow"
x:DataType="vm:MainWindowViewModel"
Icon="/Assets/avalonia-logo.ico"
Title="Распределитель СГ с ГСН" Width="1000" Height="600">

    <Design.DataContext>
        <!-- This only sets the DataContext for the previewer in an IDE,
to set the actual DataContext for runtime, set the DataContext property
in code (look at App.axaml.cs) -->
        <vm:MainWindowViewModel/>
    </Design.DataContext>

```

```

<Grid Margin="10" RowDefinitions="185, 30, 3*, 40" ShowGridLines="False">
  <Grid Grid.Row="0" ColumnDefinitions="100, 200, *, 150"
  ShowGridLines="False">
    <StackPanel Grid.Column="0">
      <Image Height="80" Margin="0 35"
  Source="E:\Storage\AaAaAaAa.jpg"/>
    </StackPanel>
    <Grid Grid.Column="1" ColumnDefinitions="100, *">
      <StackPanel Grid.Column="0">
        <TextBlock Margin="4" FontSize="18"
  FontWeight="Bold" Text="Signals"/>
        <TextBlock Margin="4" FontSize="18" Text="Support"/>
        <TextBlock Margin="4" FontSize="18"
  Text="Reference"/>
        <TextBlock Margin="4" FontSize="18" Text="Demand"/>
        <TextBlock Margin="4" FontSize="18" Text="PID Out"/>
      </StackPanel>
      <StackPanel Grid.Column="1">
        <TextBlock FontSize="18" Text="Flow"/>
        <TextBox Height="30"/>
        <TextBox Height="30"/>
        <TextBox Height="10"/>
        <TextBox Height="10"/>
      </StackPanel>
    </Grid>
    <Grid Grid.Column="2" RowDefinitions="*, 2*">
      <StackPanel Grid.Row="0">
        <TextBlock Margin="30 0" Text="Drive memories"/>
        <TextBlock Margin="30 0" Text="Value:"/>
        <TextBox Margin="30 0" HorizontalAlignment="Left"
  Width="150" Text="50%"/>
      </StackPanel>
      <StackPanel Grid.Row="1">
        <TextBlock Margin="30 10" Text="Drive status"/>
        <WrapPanel>
          <TextBlock Margin="30 0" Text="Status"/>
          <TextBox Margin="0" Width="80"
  Text="Active"/>
        </WrapPanel>
        <WrapPanel>
          <TextBlock Margin="30 0" Text="Alarm"/>
          <TextBox Margin="4 0" Width="80" Text="No
  alarm"/>
        </WrapPanel>
      </StackPanel>
    </Grid>
    <StackPanel Grid.Column="3">
      <Button Click="Click_OpenSettings"
  Width="120">Настройки</Button>
      <Button Click="ClickHandler" Width="120">Лог
  операций</Button>
    </StackPanel>
  </Grid>

  <Grid Grid.Row="1" ShowGridLines="True">
    <Grid ColumnDefinitions="100, Auto">
      <StackPanel Grid.Column="0">
        <TextBlock Text="Мнемосхема" />
      </StackPanel>
      <WrapPanel Grid.Column="1">
        <Button Click="ClickHandler" Background="White"
  Width="40" Height="27">

```

```

                <Image
Source="avares://Project/Assets/Left.png" Margin="-9"/>
                </Button>
                <Button Click="OpenSecondLevel" Background="White"
Width="40" Height="27">
                <Image
Source="avares://Project/Assets/Right.png" Margin="-9"/>
                </Button>
                <TextBlock Text="Уровни детализации узлов
оборудования" Margin="10" />
                </WrapPanel>
            </Grid>
        </Grid>

        <Grid Grid.Row="2" ShowGridLines="True">
            <Grid ColumnDefinitions="100, *">
                <Canvas Grid.Column="1">
                    <WrapPanel Canvas.Left="150" Canvas.Top="50">
                        <Button Click="OpenSecondLevel"
Background="White" Height="80" Width="80">
                            <Image
Source="avares://Project/Assets/Graph.png" Margin="-9"/>
                            </Button>
                            <Button Click="OpenSecondLevel"
Background="White" VerticalAlignment="Bottom" Margin="80,0" Height="80" Width="80">
                                <Image
Source="avares://Project/Assets/Graph.png" Margin="-9"/>
                                </Button>
                            </WrapPanel>
                        </Canvas>
                    </Grid>
                </Grid>

                <Grid Grid.Row="4" ColumnDefinitions="100, 3.5*, *">
                    <StackPanel Grid.Column="0">
                        <Button Click="Click_OnOffLine" Width="80" Height="30">OFF
LINE</Button>
                    </StackPanel>
                    <StackPanel Grid.Column="1">
                        <TextBlock Text="Выберите уровень доступа:"/>
                        <ComboBox x:Name="password" Width="200"/>
                    </StackPanel>
                    <StackPanel Grid.Column="2">
                        <ComboBox x:Name="usb" VerticalAlignment="Bottom"
HorizontalAlignment="Right" Width="150"/>
                    </StackPanel>
                </Grid>
            </Grid>
        </Window>

```

Листинг А.3.2

```

using Avalonia.Collections;
using Avalonia.Controls;
using Avalonia.Interactivity;
using Avalonia.Markup.Xaml;
using ScottPlot;
using ScottPlot.Avalonia;
using Tmds.DBus.Protocol;
namespace Project.Views
{
    public partial class MainWindow : Window
    {

```



```

public MainWindow()
{
    InitializeComponent();
    var password = this.Find<ComboBox>("password");
    var settings = this.Find<ComboBox>("settings");
    var usb = this.Find<ComboBox>("usb");

    // Добавляем элементы в ComboBox
    password.ItemsSource = new AvaloniaList<object>
    {
        "Уровень 1",
        "Уровень 2",
        "Уровень 3"
    };

    settings.ItemsSource = new AvaloniaList<object>
    {
        "Input",
        "PID 1",
        "PID 2",
        "Output"
    };

    usb.ItemsSource = new AvaloniaList<object>
    {
        "Usb/Ethernet",
        "..."
    };

    // Обработчик события при выборе элемента
    password.SelectionChanged += ComboBox_SelectionChanged;
    settings.SelectionChanged += ComboBox_SelectionChanged;
    usb.SelectionChanged += ComboBox_SelectionChanged;
}

private void ComboBox_SelectionChanged(object? sender,
SelectionChangedEventArgs e)
{
    var comboBox = (ComboBox)sender!;
    if (comboBox.SelectedIndex != -1)
    {
        var selectedItem = comboBox.SelectedItem;
        // Действия при выборе элемента из списка
    }
}

public void OpenSecondLevel(object sender, RoutedEventArgs args)
{
    var window = new SecondLevelDetailisation();
    window.Show();
    this.Close();
}

public void ClickHandler(object sender, RoutedEventArgs args)
{
}

public void Click_OpenSettings(object sender, RoutedEventArgs args)
{
    var window = new Settings();
    window.Show();
}

```

```

    }

    public void Click_OnOffLine(object sender, RoutedEventArgs args)
    {
        (sender as Button)!.Content = "ON LINE";
    }
}

```

Листинг А.4.1

```

<Window xmlns="https://github.com/avaloniaui"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:views="using:Project.Views"
        x:Class="Project.Views.Settings"
        Title="Настройки" Width="300" Height="200">
<Grid RowDefinitions="*, *, *">
    <Grid Row="0" ColumnDefinitions="*, *">
        <StackPanel Grid.Column="0">
            <TextBlock Text="IP адрес сервера"/>
        </StackPanel>
        <StackPanel Grid.Column="1">
            <TextBox/>
        </StackPanel>
    </Grid>
    <Grid Row="1" ColumnDefinitions="*, *">
        <StackPanel Grid.Column="0">
            <TextBlock Text="Синхронизация" />
        </StackPanel>
        <WrapPanel Grid.Column="1">
            <Button Click="ClickHandler" Width="100"
                Height="30">Включить</Button>
        </WrapPanel>
    </Grid>
</Grid>
</Window>

```

Листинг А.4.2

```

using Avalonia.Controls;
using Avalonia.Interactivity;

namespace Project.Views
{
    public partial class Settings : Window
    {
        public Settings()
        {
            InitializeComponent();
        }
        public void ClickHandler(object sender, RoutedEventArgs args)
        {
            (sender as Button)!.Content = "Выключить";
        }
    }
}

```

Листинг А.5.1

```

<Window xmlns="https://github.com/avaloniaui"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

```

```

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:ScottPlot="clr-
namespace:ScottPlot.Avalonia;assembly=ScottPlot.Avalonia"
xmlns:views="using:Project.Views"
x:Class="Project.Views.SecondLevelDetalisation"
Title="SecondLevelDetalisation" Width="600" Height="500">

<Grid RowDefinitions="50, Auto">
  <Grid Grid.Row="0">
    <WrapPanel>
      <Button Click="OpenFirstLevel" Background="White"
Width="40" Height="27">
        <Image Source="avares://Project/Assets/Left.png"
Margin="-9"/>
      </Button>
      <Button Click="ClickHandler" Background="White" Width="40"
Height="27">
        <Image Source="avares://Project/Assets/Right.png"
Margin="-9"/>
      </Button>
      <TextBlock Text="Уровни детализации узлов оборудования"
Margin="10" />
    </WrapPanel>
  </Grid>
  <Grid Grid.Row="1" ColumnDefinitions="*, *, *">
    <WrapPanel Grid.Column="1">
      <ScottPlot:AvaPlot Name="AvaPlot1" Height="400"
Width="400"/>
    </WrapPanel>
  </Grid>
</Grid>
</Window>

```

Листинг А.5.2

```

using Avalonia.Controls;
using Avalonia.Interactivity;
using ScottPlot.Avalonia;
using System;

namespace Project.Views
{
    public partial class SecondLevelDetalisation : Window
    {
        public SecondLevelDetalisation()
        {
            InitializeComponent();

            Plot1();
        }

        public void Plot1()
        {
            int[] dataX = new int[5000];
            int[] dataY = new int[5000];
            Random rand = new Random();
            for (int i = 0; i < 5000; i++)
            {
                dataX[i] = i;
                dataY[i] = i * 10 + rand.Next(100);
            }
            AvaPlot avaPlot1 = this.Find<AvaPlot>("AvaPlot1");
            avaPlot1.Plot.Add.Scatter(dataX, dataY);
        }
    }
}

```

```
        avaPlot1.Refresh();
    }

    public void OpenFirstLevel(object sender, RoutedEventArgs args)
    {
        (sender as Button)!.Content = "Ginger";
        var window = new MainWindow();
        window.Show();
        this.Close();
    }

    public void ClickHandler(object sender, RoutedEventArgs args)
    {
        (sender as Button)!.Content = "Dw";
    }
}
}
```