**ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет автоматики и вычислительной техники

Кафедра систем автоматизации управления

**ВЫПУСКНАЯ**

**КВАЛИФИКАЦИОННАЯ РАБОТА**

на тему

*Модернизация системы управления электрической печью отжига металла*

Пояснительная записка

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет автоматики и вычислительной техники

Кафедра систем автоматизации управления

|  |  |  |
| --- | --- | --- |
| Допускаю к защите |  |  |
| И.о. заведующего кафедрой САУ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | /Семеновых В.И../ |
|  | (подпись) | (Ф.И.О) |

МОДЕРНИЗАЦИЯ СИСТЕМЫ УПРАВЛЕНИЯ

ЭЛЕКТРИЧЕСКОЙ ПЕЧЬЮ ОТЖИГА МЕТАЛЛА

Пояснительная записка выпускной квалификационной работы

ТПЖА.090302.182 ПЗ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Разработал студент  группы ИТб-4302-02-00 |  | / Уткин Н.А./ |  |  |
| (подпись) | (Ф.И.О) |  | (дата) |
| Руководитель, к.т.н., доцент |  | / Семеновых В.И./ |  |  |
| (подпись) | (Ф.И.О) |  | (дата) |
| Консультанты: |  |  |  |  |
| по организационно-экономическому разделу, к.т.н., доцент |  | / Семеновых В.И./ |  |  |
| (подпись) | (Ф.И.О) |  | (дата) |
| Нормоконтролер, к.т.н., доцент |  | / Семеновых В.И./ |  |  |
| (подпись) | (Ф.И.О) |  | (дата) |

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

И. о. зав. кафедрой САУ                        /Семеновых В. И./

(подпись) (ФИО)

«29» апреля 2024 г.

**ЗАДАНИЕ**

**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

Студенту *Уткину Никите Андреевичу\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Тема: *Модернизация системы управления электрической печью отжига*

*металла*

(Утверждена приказом по университету от 20.11.2023 № 13-02/162/3 )

1. Исходные данные к ВКР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

а) Функциональные требования. Система должна обеспечивать возможность точного контроля температуры на всех этапах процесса отжига, включая нагрев, выдержку при определенной температуре и контролируемое охлаждение. Это включает в себя автоматическое регулирование температуры. Система должна обеспечивать автоматическую подачу защитного газа для создания контролируемой атмосферы внутри печи. Регулирование подачи газа должно осуществляться на основе показаний датчиков давления и расхода газа. Необходимо обеспечить возможность проведения белого отжига металла для предотвращения окисления. Оператор должен иметь возможность мониторинга всех ключевых параметров процесса в реальном времени через интуитивно понятный графический интерфейс. Система должна записывать и хранить данные о процессе для последующего анализа и отчетности. Система должна обеспечивать сигнализацию при отклонении параметров процесса от заданных значений.                                                                                                         

б) Системные требования. Операционные системы: Windows 10 и выше. Коммуникационные интерфейсы: Ethernet для связи ПЛК с SCADA-системой и другими компонентами системы, а также поддержку протоколов Modbus TCP, OPC UA для обмена данными между компонентами предприятия. Программное обеспечение: должно включать среду разработки для ПЛК для создания и отладки программного обеспечения управления, SCADA-систему для реализации интерфейсов мониторинга и визуализации данных, а также базы данных для хранения и анализа исторических данных о процессе отжига.

2. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов) Обзор и анализ предметной области. Проектирование информационной системы. Описание прикладных процессов и моделирование информационной системы. Разработка моделей данных. Проектирование и моделирование архитектуры и программных средств. Программная реализация информационной системы. Обзор автоматизированной системы. Разработка схем алгоритмов программы. Разработка пользовательских интерфейсов, реализация обработки данных.

3. Задание по организационной и экономической части ВКР

Расчёт затрат на создание информационной системы, цены и прибыли от ее реализации; расчёт выручки и прибыли от реализации информационной; расчёт затрат, связанных с внедрением и использованием информационной системы.

4. Прочие разделы

5. Перечень графического материала (с точным указанием обязательных чертежей)

6. Руководитель и консультант по ВКР (с указанием фамилии, имени, отчества, места работы и должности)

а) руководитель ВКР Семеновых В.И.

б) консультант по организационно-экономической Семеновых В.И.

7. Дата выдачи задания

Руководитель                                                            (подпись)

Студент                                                                      (подпись)

**КАЛЕНДАРНЫЙ ГРАФИК**

работы над ВКР на весь период проектирования с указанием

объема выполнения и трудоемкости отдельных этапов по месяцам

1. Обзор и анализ предметной области                                                        29.04.2024 г.

2. Проектирование информационной системы                                           30.04.2024 г.

3. Анализ автоматизированной системы                                                     15.05.2024 г.

4. Программная реализация и отладка                                                         20.05.2024 г.

5. Оформление ПЗ и графической части                                                      28.05.2024 г.

6. Даты: предзащиты ВКР                               защиты

Руководитель                                             (подпись)

Студент                                              (подпись)

**Реферат**

Уткин Н. А. Модернизация системы управления электрической печью отжига металла: ТПЖА.090302.182 ПЗ: Выпускная квалификационная работа / ВятГУ, каф. САУ; рук. Семеновых В. И. – Киров, 2024. ПЗ 98 с., 30 рис., 12 источников, 11 прил., 9 табл.

ТЕХНОЛОГИЧЕСКИЙ ПРОЦЕСС, АСУТП, SCADA-СИСТЕМА, ПЛК, CODESYS, БАЗА ДАННЫХ, ДАТЧИК, ИСПОЛНИТЕЛЬНЫЙ МЕХАНИЗМ, ЭЛЕКТРИЧЕСКАЯ ПЕЧЬ, МУФЕЛЬ.

Объект исследования и разработки – система управления электрической печью отжига металла.

Цель работы – решение проблемы процесса отжига металла для достижения требуемых механических свойств и качества металла путем модернизации системы управления электрической печью процесса отжига металла.

Разработана автоматизированная система управления технологическим процессом производства отжига металла. Проведена работа по исследованию различных литературных источников для решения проблем модернизации системы управления электрической печью металла и внедрения ее в производственный процесс. Разработана структурная схема системы управления. Разработаны диаграммы в нотации IDEF0, DFD, DFD, IDEF1x. Проведен анализ существующей автоматизированной системы и предлагаемых технических средств. Разработана функциональная схема автоматизации АСУТП и алгоритм её работы. Разработаны схемы алгоритмов программ. Разработано программное обеспечение АСУТП с использованием SCADA-системы и среды разработки прикладных программ для ПЛК. Проведен экономический расчет для обоснования разработки системы.

**Содержание**

ТПЖА. 090302.182 ПЗ

Изм.

Лист

№ докум.

Подпись

Дата

Лист

2

Разраб.

Уткин Н. А.

Проверил

Семеновых В. И.

Т. контр

Семеновых В. И.

Н. контр.

Утв.

Семеновых В. И.

Модернизация системы управления электрической печью отжига металла

Лит.

Листов

98

*Кафедра САУ,*

*группа ИТб-4302-02-00*

[Введение 4](#_Toc170474318)

[1 Предметная область 6](#_Toc170474319)

[1.1 Описание предметной области 6](#_Toc170474320)

[1.2 Анализ предметной области 7](#_Toc170474321)

[1.3 Цели и задачи 9](#_Toc170474322)

[1.4 Вывод к разделу 1 10](#_Toc170474323)

[2 Проектирование информационной системы 11](#_Toc170474324)

[2.1 Определение основных принципов и подходов к реализации проекта 11](#_Toc170474325)

[2.2 Выбор конкретных методов и подходов реализации проекта 13](#_Toc170474326)

[2.3 Разработка функциональных моделей 18](#_Toc170474330)

[2.4 Разработка моделей данных 32](#_Toc170474334)

[2.5 Вывод к разделу 2 36](#_Toc170474337)

[3 Автоматизированная система 37](#_Toc170474338)

[3.1 Анализ функциональной схемы автоматизации 37](#_Toc170474339)

[3.2 Анализ комплекса технических средств 41](#_Toc170474342)

[3.3 Разработка алгоритмов программ 49](#_Toc170474345)

[3.4 Вывод к разделу 3 54](#_Toc170474348)

[4 Разработка программного обеспечения 56](#_Toc170474349)

[4.1 Разработка модулей управления 57](#_Toc170474350)

[4.2 Разработка экранных форм 64](#_Toc170474354)

[4.3 Вывод к разделу 4 67](#_Toc170474355)

[5 Экономическое обоснование разработки программного обеспечения 69](#_Toc170474356)

[5.1 Расчет затрат на создание ПО и его цены 69](#_Toc170474357)

[5.2 Расчет затрат, связанных с внедрением и использованием ПО 74](#_Toc170474358)

[5.3 Вывод к разделу 5 77](#_Toc170474359)

[Заключение 78](#_Toc170474360)

[Приложение А (обязательное). Программа main.cfc 80](#_Toc170474361)

[Приложение Б (обязательное). Программа модуля нагрева 81](#_Toc170474362)

[Приложение В (обязательное). Программа PWM.st 86](#_Toc170474363)

[Приложение Г (обязательное). Программа посредник между ПЛК и БД 87](#_Toc170474364)

[Приложение Д (обязательное). Схема алгоритма нагрева 91](#_Toc170474365)

[Приложение Ж (обязательное). Функциональная схема автоматизации электри-ческой печи отжига металла 92](#_Toc170474366)

[Приложение К (обязательное). Экранная форма панели трендов температуры 93](#_Toc170474367)

[Приложение Л (обязательное). Экранная форма панели InfluxDB 94](#_Toc170474368)

[Приложение М (обязательное). Авторская справка 95](#_Toc170474369)

[Приложение Н (справочное). Список обозначений и сокращений 96](#_Toc170474370)

[Приложение Р (справочное). Библиографический список 97](#_Toc170474371)

# Введение

Современная промышленность стремительно развивается, требуя эффективных и инновационных подходов к управлению производственными процессами. В этом контексте особое внимание уделяется процессам термической обработки металла, таким как отжиг, который играет ключевую роль в обеспечении желаемых механических свойств металла.

ПАО «Омутнинский металлургический завод» специализируется на производстве стальных фасонных профилей высокой точности и горячекатаных профилей. Предприятие охватывает весь производственный цикл – от выплавки стали до глубокой переработки черных металлов.

Для обеспечения качества и безопасности производства завод руководствуется соответствующими нормативными документами, включая ГОСТы и технические условия, гарантирующие соответствие выпускаемой продукции высоким стандартам качества.

Актуальность данного проекта обусловлена необходимостью повышения эффективности и точности производственного процесса отжига металла на промышленном предприятии. Отжиг металла является ключевым этапом в производстве, влияющим на механические свойства и качество металла. Точное контролирование температуры и времени отжига не только оптимизирует процесс, но и обеспечивает стабильное качество конечного продукта. В современных условиях повышается требование к автоматизации и контролю производственных процессов с целью сокращения затрат на производство, увеличения производительности и обеспечения стабильного качества продукции.

Объектом автоматизации является электрическая муфельная печь, которая служит для отжига металла. Данный процесс состоит из 5 этапов. В настоящее время для управления технологическим процессом используется устройства контроля и регистрации ФЩЛ-501.

Целью данного проекта является решение проблемы процесса отжига металла для достижения требуемых механических свойств и качества металла путем модернизации системы управления электрической печью процесса отжига металла. Задача заключается в создании визуализации процесса, современной системы управления, обеспечивающей точный контроль температуры, эффективное распределение защитного газа, и оптимизацию временных параметров отжига.

Для достижения поставленной цели необходимо решить следующие задачи:

* проведение анализа текущей системы управления и выявление её недостатков;
* определение требований к новой;
* выбор и обоснование оборудования для новой системы;
* разработка алгоритмов управления процессом отжига.
* создание программного обеспечения для ПЛК и систем мониторинга, включая разработку интерфейсов для оператора;

Настоящий проект объединяет в себе аналитический подход к изучению предметной области, разработку программных модулей для управления процессом отжига, и применение инженерных решений для повышения эффективности и надежности производственных операций. Внедрение разработанных решений ожидается приведет к улучшению качества обработки металла, снижению затрат и повышению общей производительности производства.

# 1 Предметная область

## 1.1 Описание предметной области

Предметная область данного проекта связана с технологическим процессом отжига металла [3] в электрических муфельных печах, используемых в металлургическом производстве. Отжиг металла представляет собой стадию термической обработки, целью которой является улучшение механических свойств металла, таких как пластичность, прочность и снижение твердости. Этот процесс включает нагревание металла до определенной температуры, выдержку при этой температуре и последующее контролируемое охлаждение.

Описание предметной области представлено технологическим процессом.

Первый этап включает в себя подготовку процесса. Перед началом отжига металлические заготовки загружаются внутрь печи. Затем заготовки накрываются муфелем, что предотвращает контакт с внешней средой. Отжиг металла бывает черный и белый, что ведёт к следующему этапу процесса отжига.

Второй этап включает в себя создание защитной атмосферы. Перед началом нагрева система подает инертный газ внутрь печи. Это создает защитную атмосферу, что снижает окисление металла под воздействием высоких температур.

Следующий, третий этап – нагрев. Оператор задает необходимые параметры нагрева, такие как температура, время выдержки. Печь пошагово нагревается в соответствии с установленными параметрами.

Четвертый этап – контроль параметров. В процессе нагрева оператор следит за температурой внутри печи, скоростью нагрева и другими ключевыми параметрами. Полученные данные используются для обеспечения стабильности и эффективности процесса.

Пятый этап – выдержка и охлаждение. После достижения заданной температуры материал выдерживается в течение определенного времени для обеспечения равномерности нагрева. Затем начинается контролируемый процесс охлаждения.

Завершающим этапом система автоматически оповещает о готовности заготовок. После этого заготовки выгружаются из печи.

На рисунке 1.1 представлена схема, демонстрирующая ключевые шаги процесса отжига металла и является отправной точкой для более детального анализа технологических аспектов данного процесса.

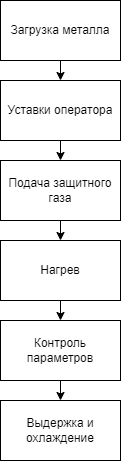


Рисунок 1.1 – Упрощенная технологическая схема процесса отжига металла

Каждый этап важен для обеспечения эффективности и качества процесса отжига, а детализация каждой задачи помогает лучше понять, как система функционирует.

## 1.2 Анализ предметной области

Анализ предметной области выявил несколько ключевых проблем, с которыми сталкивается существующая система управления электрической печью для отжига металла.

Одной из основных проблем является техническое устаревание оборудования. Устройство ФЩЛ-501 [2] (рисунок 1.2), используемое для контроля и регулирования температуры, является устаревшим и больше не выпускается. Это устройство предназначено для автоматизации контроля, отображения и регистрации параметров технологических процессов, но его технические характеристики не соответствуют современным требованиям. Например, оно обладает ограниченными возможностями по точности измерений и быстродействию и не может обеспечивать необходимую надежность и стабильность работы. Устаревшие компоненты склонны к частым поломкам и требуют значительных затрат на обслуживание и ремонт, что увеличивает эксплуатационные расходы и снижает общую производительность системы.



Рисунок 1.2 – Устройство ФЩЛ-501

Существующая система управления не предусматривает возможности автоматического регулирования подачи защитного газа. Это означает, что невозможно осуществить белый отжиг металла, который требует создания контролируемой атмосферы внутри печи для предотвращения окисления металла и обеспечения стабильных условий для процесса отжига. Отсутствие возможности подачи защитного газа приводит к простоям печи, что ведет к неоптимальному использованию ресурсов и увеличению производственных затрат.

Еще одной значимой проблемой является отсутствие современных систем мониторинга и хранения данных. Существующая система не позволяет операторам в реальном времени отслеживать параметры процесса и оперативно реагировать на изменения. Ограниченные возможности визуализации данных и отсутствие интеграции с информационными системами предприятия затрудняют сбор и анализ данных, что снижает возможность принятия обоснованных управленческих решений. Это приводит к задержкам в обнаружении и устранении проблем, снижая общую эффективность и надежность производственного процесса.

Отсутствие интеграции с другими информационными системами предприятия также является серьезной проблемой. Система управления процессом отжига не связана с другими системами, что затрудняет координацию работы различных подразделений и обмен данными. Это приводит к низкой эффективности работы и увеличению времени на принятие решений, что негативно сказывается на общей производительности предприятия.

## 1.3 Цели и задачи

Модернизируемая система управления электрической печи отжига металла предназначена для повышения эффективности и точности производственного процесса отжига металла.

При анализе предметной области была выделена следующая цель проекта: решение проблемы процесса отжига металла для достижения требуемых механических свойств и качества металла путем модернизации системы управления электрической печью процесса отжига металла.

Основываясь на цели разработки системы, были выделены соответствующие задачи:

* провести замену устаревшего оборудования ФЩЛ-501 на современный программируемый логический контроллер;
* разработать алгоритмы управления нагревом и подачей защитного газа, включающие автоматическое регулирование в зависимости от измерений датчиков;
* разработать автоматизированное рабочее место оператора посредством установки SCADA-системы для мониторинга и визуализации данных;
* интегрировать систему управления с базами данных для хранения и анализа исторических данных о процессе отжига.

## 1.4 Вывод к разделу 1

В первой главе рассмотрена и проанализирована предметная область, что привело к решению о проектировании информационной системы и разработке автоматизированной системы управления процессом отжига металла. Также представлены основные определения, касающиеся данной области, обозначены цели и задачи системы.

# 2 Проектирование информационной системы

Проектирование информационной системы играет ключевую роль в обеспечении функциональности и эффективности разрабатываемого проекта. Здесь представлены основные этапы проектирования, которые направлены на создание структурированной и интегрированной системы, способной эффективно управлять производственными.

Разрабатываемая информационная система направлена на автоматизацию процесса ввода и хранения отчетных данных, а также на обеспечение пользователей системы управления электрической печью для отжига металла возможностью ведения учета статистики по контролю за датчиками и исполнительными механизмами. Эта информация может быть использована для генерации отчетов и анализа работы системы.

## 2.1 Определение основных принципов и подходов к реализации проекта

В данном разделе будет рассмотрено определение основных принципов и подходов к реализации проекта по автоматизации системы управления электрической печью для отжига металла. Основываясь на собранных данных о технологическом процессе, оборудовании и требованиях к системе, будет проведен анализ различных методов и подходов, которые позволят эффективно реализовать поставленные задачи проекта.

Концепция данного проекта направлена на разработку автоматизированной системы управления электрической печи для отжига металла. Основной целью является обеспечение точного контроля температуры и времени отжига для достижения требуемых механических свойств и качества металла. Для достижения этой цели выделяются основные требования к разрабатываемой системе.

В рамках данного проекта выделены следующие основные требования к разрабатываемой системе:

* требования к функциональности: необходимость постоянного контроля и поддержания установленных значений температуры в каждой из зон, определение временных режимов для минимизации расходов ресурсов, идентификация аварийных ситуаций и формирование пользовательских отчетов;
* требования к надежности и безопасности: обеспечение стабильной работы системы при различных условиях эксплуатации, реализация механизмов защиты от аварийных ситуаций и гарантирование безопасности персонала;
* требования к производительности: обеспечение быстрого отклика системы на изменения параметров и максимизация времени работы оборудования;
* требования к масштабируемости и расширяемости: необходимость добавления дополнительных функций и компонентов в будущем, совместимость с различными типами оборудования и средами;
* требования к пользовательскому интерфейсу: создание интуитивно понятного интерфейса для оператора, предоставление информации о текущем состоянии системы и возможность настройки параметров через удобный интерфейс.

По результатам анализа указанных требований можно заключить, что необходимо разработать автоматизированную систему, которая минимизирует необходимость в человеческом вмешательстве, оставляя его только для параметров по технологическому процессу.

Основой автоматизированной системы должен быть программируемый логический контроллер (ПЛК), который должен соответствовать следующим требованиям:

* производительность: ПЛК должен обеспечивать достаточную производительность для обработки всех входящих сигналов, выполнения алгоритмов управления и обеспечения своевременной реакции на изменения в системе;
* надежность: ПЛК должен быть надежным и стабильным в работе, чтобы обеспечить непрерывную работу системы без сбоев и перебоев;
* модульность и расширяемость: ПЛК должен поддерживать модульную архитектуру, позволяющую легко расширять функциональность системы путем добавления новых модулей в соответствии с требованиями проекта;
* совместимость с протоколами связи: ПЛК должен поддерживать различные промышленные протоколы связи, такие как Modbus, OPC UA, Ethernet/IP и другие, для обеспечения совместимости с другими устройствами и системами;
* удобство программирования: ПЛК должен иметь удобное и интуитивно понятное программное обеспечение для разработки и отладки программ управления, что упростит процесс разработки и снизит время внедрения;
* защита от внешних воздействий: ПЛК должен быть защищен от воздействия внешних факторов, таких как электромагнитные помехи, вибрации, перепады напряжения и другие, чтобы обеспечить стабильную работу даже в условиях неблагоприятной окружающей среды;
* встроенная диагностика и мониторинг: ПЛК должен обладать возможностью проведения встроенной диагностики состояния системы и мониторинга работы всех подключенных устройств для своевременного обнаружения и устранения неисправностей;
* безопасность: ПЛК должен обеспечивать высокий уровень безопасности системы управления, включая защиту от несанкционированного доступа и возможность быстрого восстановления после сбоев;
* поддержка стандартов и нормативов: ПЛК должен соответствовать промышленным стандартам и нормативам, таким как IEC 61131, для обеспечения совместимости и безопасности работы системы.

В результате анализа требований есть возможность создать систему управления электрической печи отжига металла.

## 2.2 Выбор конкретных методов и подходов реализации проекта

Для обеспечения проектирования и избегания ошибок в разработке автоматизированной системы необходимо разработать проектное решение, включающее в себя детальное описание объекта проектирования, схемы выполнения задач, а также варианты реализации требуемого функционала. Проектные решения также включают в себя разработку структурной схемы системы, выбор подходящих вариантов интерфейса доступа к системе, а также проектирование процесса обмена информацией между системой и интерфейсом доступа.

### 2.2.1 Структурная схема системы управления

В данном пункте представлена структурная схема (рисунок 2.1) автоматизированной системы, разработанной для управления системой электрической печи отжига металла.

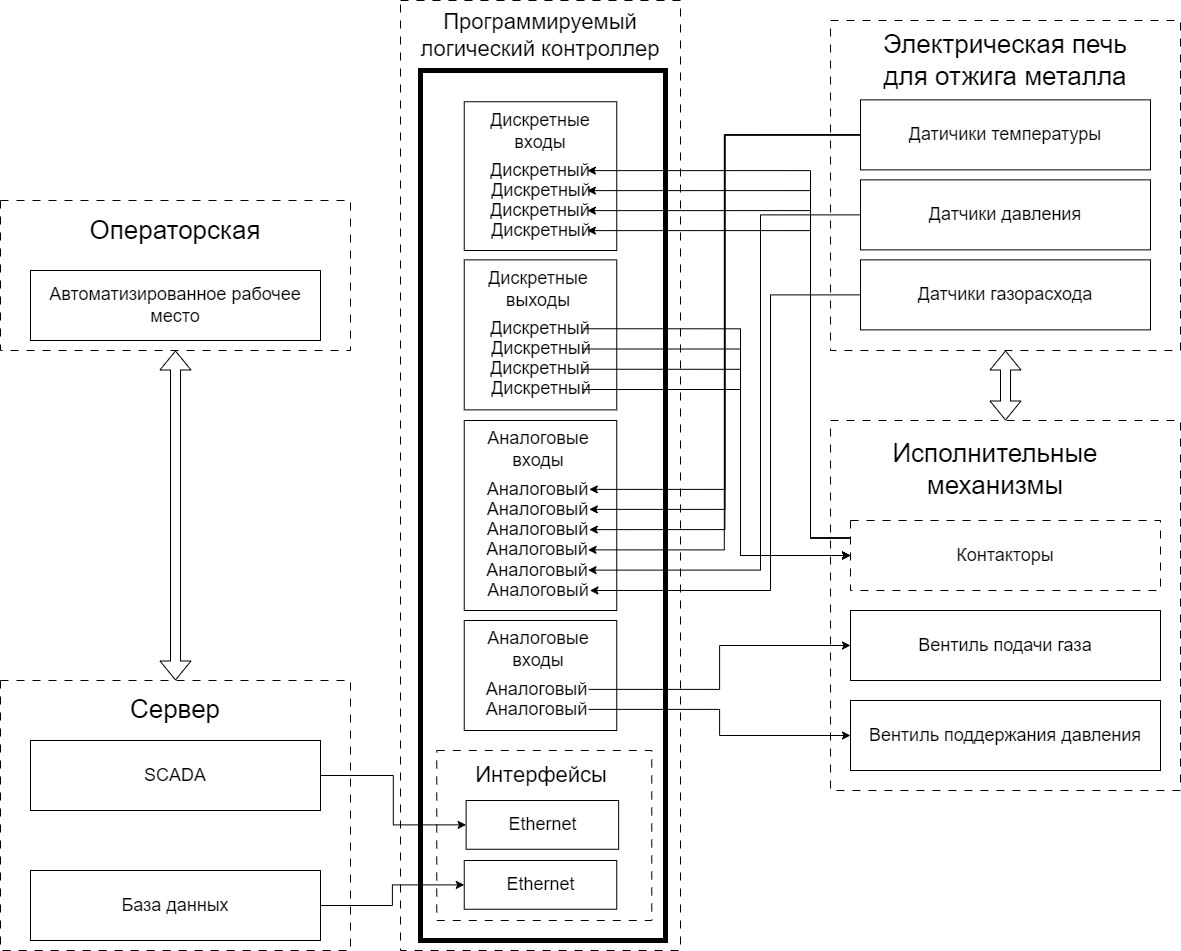


Рисунок 2.1 – Структурная схема

Структурная схема автоматизированной системы управления электрической печи отжига металла иллюстрирует взаимодействие между различными компонентами системы, включая операционное рабочее место, сервер, программируемый логический контроллер, электрическую печь и исполнительные механизмы. В центре схемы находится ПЛК, который является ключевым компонентом для обработки данных и управления процессом.

Операторская оснащена системой, которая предоставляет интерфейс для визуализации данных и взаимодействия с контроллером. Оператор может задавать уставки, контролировать параметры процесса и реагировать на возникающие ситуации. Это автоматизированное рабочее место обеспечивает оператору контроль над процессом отжига.

Сервер содержит программное обеспечение SCADA и базу данных для хранения исторических данных о процессе отжига. Сервер взаимодействует с ПЛК через интерфейс Ethernet, обеспечивая центральное управление и хранение данных. Сервер получает данные от ПЛК, сохраняет их в базе данных и предоставляет оператору доступ к этим данным для анализа и управления процессом.

ПЛК является центральным элементом системы, обеспечивая обработку сигналов от датчиков и управление исполнительными механизмами. ПЛК принимает дискретные входы от различных датчиков и переключателей, которые используются для мониторинга состояния системы. Дискретные выходы ПЛК управляют реле и контакторами, которые включают и выключают нагревательные элементы и другие устройства. Аналоговые входы ПЛК принимают сигналы от датчиков температуры, давления и расхода газа, что позволяет контролировать и регулировать параметры процесса. Аналоговые выходы используются для управления регулирующими устройствами, такими как вентилями подачи газа и давления.

Электрическая печь оборудована датчиками температуры, давления и расхода газа. Датчики температуры измеряют температуру в различных зонах печи, датчики давления контролируют давление газа внутри печи, а датчики газорасхода измеряют количество подаваемого защитного газа. Все эти данные передаются на ПЛК для анализа и управления процессом отжига.

Исполнительные механизмы включают контакторы и вентили. Контакторы используются для включения и выключения нагревательных элементов печи. Вентиль подачи газа регулирует подачу защитного газа в печь, а вентиль поддержания давления контролирует давление газа внутри печи. ПЛК управляет этими механизмами на основе данных от датчиков и заданных уставок, обеспечивая точное регулирование температуры, подачи газа и давления в процессе отжига.

Взаимодействие компонентов системы осуществляется следующим образом: ПК оператора взаимодействует с сервером, предоставляя оператору доступ к визуализации данных и управлению процессом. Сервер через интерфейс Ethernet обменивается данными с ПЛК, получая от него информацию о состоянии системы и отправляя команды управления. ПЛК получает данные от датчиков, установленных на печи, и управляет исполнительными механизмами на основе этих данных, обеспечивая выполнение технологического процесса отжига.

### 2.2.2 Графические интерфейсы доступа к системе

Управление системой электрической печи, а также своевременное получение данных о показаниях и состоянии системы и ее исполнительных механизмов обеспечивается благодаря интерфейсу доступа к системе.

В качестве интерфейса доступа выступает совокупность средств, при помощи которых пользователь взаимодействует с различными программами и устройствами. Такими средствами для проектируемой системы являются графический интерфейс (функции представляются графическими элементами экрана).

Одним из графических интерфейсов является SCADA, доступ к которой представлен на персональном компьютере, который расположен в операторской. Данная SCADA позволяет устанавливать параметры процесса отжига металла, также получать текущую информацию о состоянии автоматизированной системы электрической печи отжига металла.

Другим графическим интерфейсом является HMI-дисплей. Этот дисплей предоставляет оператору текущую информацию о состоянии системы управления электрической печью для отжига металла и позволяет осуществлять управление ее температурными, газовыми и временными параметрами.

### 2.2.3 Средства доступа к данным

Средства доступа к данным являются ключевым компонентом системы управления процессом отжига, обеспечивая сбор, хранение и анализ данных, необходимых для мониторинга и управления технологическим процессом. В этом разделе рассматриваются используемые технологии и базы данных, обеспечивающие управление данными, включая протокол OPC UA [6], InfluxDB [7] и PostgreSQL.

OPC UA (Open Platform Communications Unified Architecture) является протоколом связи, предназначенным для обмена данными между устройствами и системами автоматизации. Этот протокол обладает универсальностью, поддерживая обмен данными между устройствами различных производителей и системами на разных платформах. Высокий уровень безопасности в OPC UA достигается за счет шифрования данных и аутентификации пользователей, что критически важно для промышленной автоматизации.

OPC UA обеспечивает масштабируемость, позволяя интегрировать устройства и системы различной сложности и масштаба. В системе управления процессом отжига протокол OPC UA используется для интеграции различных устройств и систем, обеспечивая обмен данными между контроллерами и SCADA-системой FUXA [10]. Это позволяет не только мониторить состояние системы в реальном времени, но и управлять ею, отправляя команды на устройства.

InfluxDB – это высокопроизводительная база данных временных рядов, оптимизированная для хранения и обработки временных данных, таких как температурные тренды и другие параметры процесса отжига. Специально разработанная для работы с временными рядами данных, InfluxDB идеально подходит для хранения параметров технологического процесса, которые изменяются с течением времени. База данных обеспечивает высокую скорость записи и чтения данных, что позволяет эффективно обрабатывать большие объемы данных в реальном времени. Ее масштабируемость позволяет легко поддерживать увеличение объема данных и числа запросов.

В системе управления InfluxDB используется для записи и хранения временных данных о температуре, давлении и других параметрах процесса. Она предоставляет инструменты для анализа временных рядов данных, что позволяет выявлять тенденции и аномалии, а также интегрируется с SCADA-системой FUXA для отображения данных в виде графиков и диаграмм.

PostgreSQL – это мощная реляционная база данных с открытым исходным кодом, используемая для хранения структурированных данных, таких как рецепты технологического процесса и данные энергопотребления. PostgreSQL обеспечивает высокий уровень надежности и безопасности данных, что важно для критически важных приложений. Ее расширяемость позволяет адаптировать систему под специфические требования, поддерживая расширения и пользовательские функции. Мощные средства выполнения сложных SQL-запросов и анализа данных делают PostgreSQL идеальной для хранения и управления структурированными данными.

В системе управления PostgreSQL используется для сохранения параметров рецептов, которые используются для настройки процесса отжига, и записи данных о потреблении энергии.

Для обеспечения доступа к данным и их анализа используются различные инструменты. Интерфейс SCADA-системы FUXA предоставляет графический интерфейс для мониторинга и анализа данных в реальном времени, интегрированный с InfluxDB, PostgreSQL и OPC UA.

## 2.3 Разработка функциональных моделей

Создание современных информационных систем представляет собой сложную задачу, требующую применения специализированных методик и инструментов. Существуют технологии и инструменты, которые способствуют систематизации и автоматизации этапов разработки программного обеспечения. К примеру, BPWin поддерживает методологии IDEF0, IDEF3 и DFD, в то время как ERWin предоставляет возможности для прямого и обратного проектирования базы данных и поддерживает методологию IDEF1x.

### 2.3.1 Функциональная модель IDEF0

Методология IDEF0 [4] предоставляет возможность подробного описания бизнес-процессов автоматизированной системы управления электрической печи отжига металла, используя формальный и наглядный графический язык.

Одним из ключевых преимуществ IDEF0 является ее упор на взаимосвязь между объектами, а не только на последовательность событий во времени. Это позволяет рассматривать логические отношения между функциями системы.

Основной концепцией в стандарте является функция, которая представляет собой действие над входными данными, обрабатываемое определенным механизмом, и результатом являются выходные данные. В контексте проектируемой системы управления, основной функцией является обработка команд от пользователя для управления работой системы. Таким образом, единственным элементом на контекстной диаграмме будет автоматизированная система управления электрической печью для отжига металла.

Для управления системой электрической печи отжига металла необходимо получать от пользователя соответствующие параметры и обрабатывать их.

На вход схемы поступают два основных типа данных: уставки оператора и сигналы с датчиков. Уставки оператора представляют собой вводимые оператором параметры, такие как желаемые температуры, временные параметры, газорасход и уставки давления. Сигналы с датчиков представляют данные о текущем состоянии системы, такие как текущая температура, количество подаваемого газа и давление в печи, измеренные датчиками. Управление системой осуществляется на основе технологического процесса. Технологический процесс определяет последовательность операций, необходимых для достижения желаемых результатов производственного процесса, включая нагрев металла до определенной температуры и удержание его на этой температуре в течение определенного времени.

Механизм управления включает в себя программируемый логический контроллер (ПЛК), который является центральным устройством управления системой. ПЛК обрабатывает входные данные, принимает решения на основе технологического процесса, и выдает управляющие сигналы исполнительным механизмам. Исполнительные механизмы, в свою очередь, выполняют действия, необходимые для поддержания заданных параметров процесса, такие как управление нагревом.

Выходные данные системы включают в себя управляющие сигналы, которые управляют исполнительными механизмами, а также аналитические данные о текущем состоянии и ходе выполнения производственного процесса. Эти данные могут использоваться персоналом для мониторинга процесса, а также для анализа и оптимизации производственных операций.

Таким образом, на рисунке 2.2 изображена основная функциональная модель, которая отображает ключевую функцию, входные и выходные данные, а также элементы управления и управляемые устройства.



Рисунок 2.2 – Функциональная диаграмма IDEF0

Затем следует декомпозировать функциональную модель IDEF0 (рисунок 2.3), описав последовательность управления системой электрической печи отжига металла, для этого нужно выделить следующие блоки:

* определение установленных параметров и состояния датчиков;
* обработка параметров температуры газа и давления;
* визуализация;
* установка параметров для поддержания процессов.

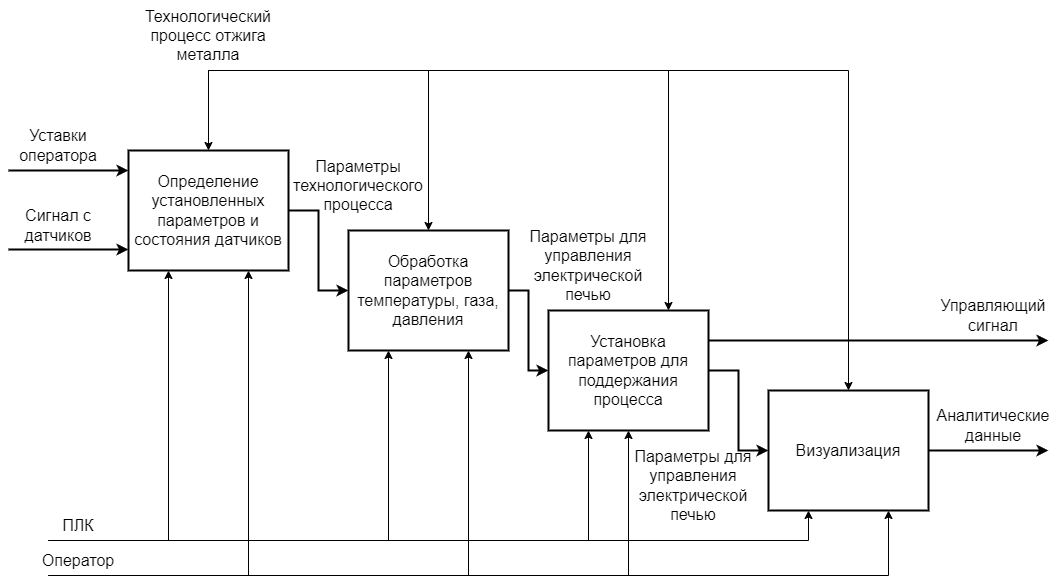


Рисунок 2.3 – Декомпозиция функциональной диаграммы IDEF0

После выделения основных модулей в декомпозиции функциональной модели IDEF0, следует провести более подробное рассмотрение каждого модуля для полного понимания процесса управления системой электрической печи для отжига металла.

Декомпозиция функциональной диаграммы IDEF0 автоматизированной системы управления электрической печью отжига металла включает несколько ключевых этапов, каждый из которых выполняет определенную функцию для обеспечения корректной работы системы.

Определение установленных параметров и состояния датчиков: на этом этапе происходит сбор информации о текущих параметрах, заданных для процесса отжига. Это включает данные о температуре, уровне газа и давлении в системе. Кроме того, проверяется состояние датчиков, которые мониторят эти параметры, чтобы убедиться в их корректной работе и отсутствии сбоев.

Обработка параметров температуры, газа и давления: полученные данные о параметрах температуры, газа и давления проходят обработку. Этот процесс включает анализ текущих значений и сравнение их с заданными уставками. Если выявляются отклонения от установленных значений, система генерирует соответствующие команды для корректировки параметров, обеспечивая тем самым поддержание необходимых условий для оптимального процесса отжига.

Визуализация: обработанные данные и текущие параметры процесса визуализируются на экране оператора. Визуализация включает отображение текущей температуры, уровня газа, давления и других ключевых параметров, что позволяет оператору в реальном времени контролировать состояние системы и принимать оперативные решения при необходимости.

Установка параметров для поддержания процесса: на основе обработанных данных система автоматически устанавливает необходимые параметры для поддержания процесса отжига. Это включает регулировку температуры, подачи газа и давления в соответствии с заданными уставками. Автоматическая настройка параметров позволяет обеспечить стабильность процесса и минимизировать влияние человеческого фактора.

Эти этапы взаимодействуют между собой, обеспечивая надежную и эффективную работу системы управления электрической печью отжига металла. Сбор и обработка данных, визуализация текущих параметров и автоматическая настройка системы позволяют поддерживать оптимальные условия для отжига, повышая производительность и качество продукции.

Далее декомпозируем блок обработка параметров температуры, газа и давления. Декомпозиция представлена на рисунке 2.4.

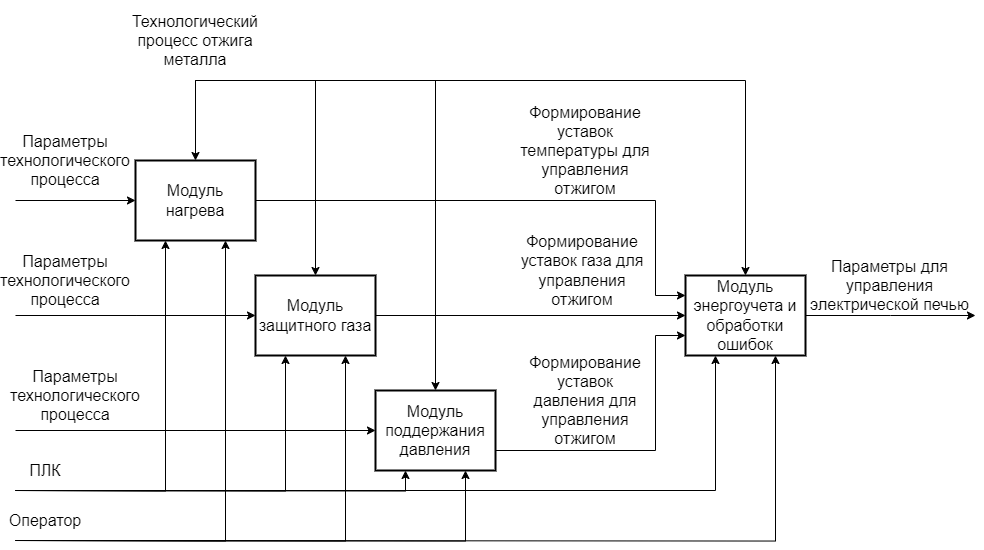


Рисунок 2.4 – Декомпозиция обработки параметров температуры, газа, давления

Далее следует рассмотреть блок обработки параметров температуры, газа и давления более детально.

Параметры технологического процесса: эти параметры подаются на вход трёх основных модулей системы: модуля нагрева, модуля защитного газа и модуля поддержания давления. Эти модули взаимодействуют с технологическим процессом отжига металла, обеспечивая необходимые условия для его осуществления.

ПЛК и оператор: программируемый логический контроллер (ПЛК) и оператор также взаимодействуют с модулями системы. ПЛК принимает и обрабатывает сигналы, а оператор может управлять системой вручную, задавая параметры и контролируя процесс.

Модуль нагрева: этот модуль отвечает за контроль и поддержание необходимой температуры в процессе отжига. Он получает параметры технологического процесса и формирует уставки температуры для управления отжигом.

Модуль защитного газа: этот модуль регулирует подачу защитного газа, необходимого для предотвращения окисления металла в процессе отжига. Он также получает параметры технологического процесса и формирует уставки газа для

управления отжигом.

Модуль поддержания давления: этот модуль управляет давлением в системе отжига, поддерживая его на необходимом уровне. Параметры технологического процесса поступают в модуль, и он формирует уставки давления для управления отжигом.

Формирование уставок для управления отжигом: формирование уставок температуры, газа и давления происходит в соответствующих модулях. Эти уставки затем передаются в модуль энергоучета и обработки ошибок.

Модуль энергоучета и обработки ошибок: этот модуль интегрирует уставки температуры, газа и давления, контролируя энергетическое потребление системы и обрабатывая возможные ошибки. Он выдает параметры для управления электрической печью, обеспечивая стабильность и эффективность процесса отжига.

Таким образом, данная система обеспечивает автоматизированное управление процессом отжига металла в электрической печи, поддерживая оптимальные условия температуры, защитного газа и давления.

### 2.3.2 Функциональная модель IDEF3

Методология IDEF3 является стандартом для документирования технологических процессов на предприятии, предоставляя инструменты для наглядного анализа и моделирования их сценариев – последовательности изменений свойств объекта в рамках рассматриваемого процесса. IDEF3-моделирование дополняет традиционное моделирование по методологии IDEF0.

Диаграмма IDEF3 включает следующие основные элементы:

* работы: основные процессы или объекты, которые описываются;
* стрелки или связи: отображение перемещений между блоками в процессе;
* перекрёстки: отображение логики взаимодействия потоков при слиянии и разветвлении, а также множества событий, которые могут или должны быть завершены перед началом следующей работы;
* объекты ссылок: внешние сущности.

Основная функция описываемой системы – управление системой процесса отжига металла посредством команд от пользователя и правил их обработки.

На рисунке 2.5 представлена модель IDEF3, отражающая временную последовательность выполняемых действий.

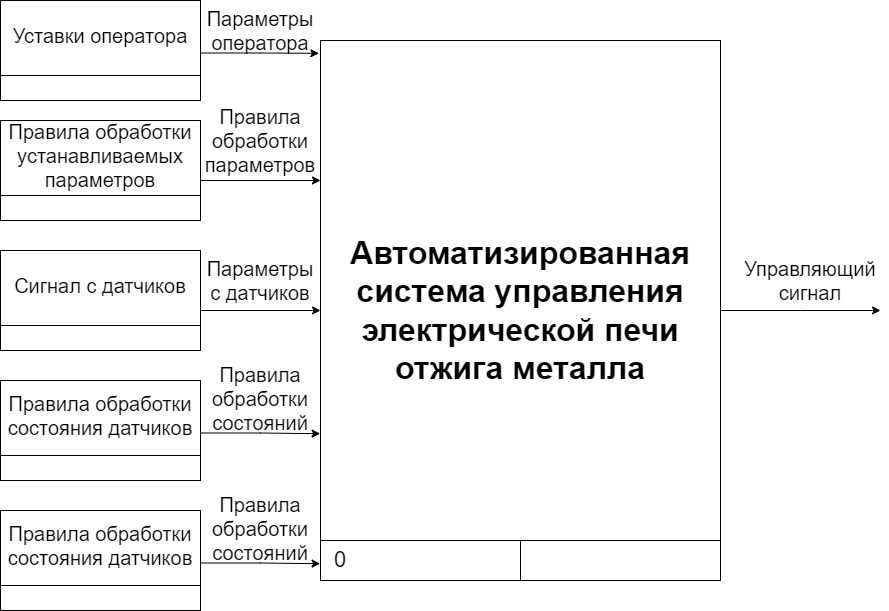


Рисунок 2.5 – Функциональная диаграмма IDEF3

Основные элементы данной модели включают: уставки оператора: оператор задает необходимые параметры процесса отжига, включая целевые температуры, временные интервалы и другие технологические параметры. Эти уставки служат основными входными данными для системы управления.

Правила обработки устанавливаемых параметров: после задания уставок оператором, система применяет заранее определенные правила обработки для проверки корректности и адекватности установленных значений. Это включает в себя проверку на соответствие технологическим требованиям и допустимым диапазонам.

Сигнал с датчиков: в процессе работы печи, система непрерывно получает сигналы с различных датчиков, установленных внутри печи и на её компонентах. Эти датчики измеряют текущие параметры, такие как температура, давление, расход газа и другие важные показатели.

Правила обработки состояния датчиков: система обрабатывает полученные с датчиков сигналы в соответствии с установленными правилами. Это необходимо для выявления отклонений от нормальных рабочих параметров и определения текущего состояния системы. Например, если температура выходит за установленные пределы, система регистрирует это событие и предпринимает соответствующие действия.

Автоматизированная система управления электрической печи отжига металла: центральный элемент модели, который интегрирует все входные данные и применяет их для управления процессом отжига. На основе уставок оператора и сигналов с датчиков система автоматически регулирует работу печи, корректируя параметры нагрева, подачи газа и других процессов для достижения оптимальных условий отжига.

Таким образом, функциональная модель IDEF3 описывает, как автоматизированная система управления электрической печи отжига металла функционирует в реальном времени, обрабатывая входные данные от оператора и датчиков, применяя правила обработки и обеспечивая точный контроль технологического процесса. Это позволяет поддерживать стабильные условия отжига, что важно для достижения требуемых механических свойств и качества металла.

Далее стоит провести декомпозицию функциональной модели IDEF3 для автоматизированной системы управления электрической печи отжига металла для описания процессов и их взаимосвязей. Декомпозиция представлена на рисунке 2.6.

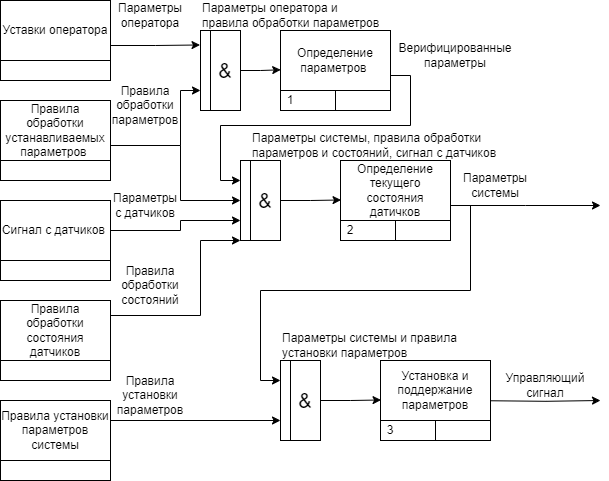


Рисунок 2.6 – Декомпозиция функциональной диаграммы IDEF3

Определение параметров: в этот блок поступают уставки оператора и правила обработки устанавливаемых параметров. Оператор задает целевые значения температуры, времени отжига и других критически важных параметров. Эти значения проверяются и обрабатываются согласно установленным правилам для обеспечения их корректности и соответствия требованиям технологического процесса. После обработки параметры передаются в следующий блок для дальнейшего использования.

Определение текущего состояния датчиков: этот блок отвечает за сбор и анализ данных, поступающих с различных датчиков, установленных в системе. Сигналы с датчиков включают в себя текущие значения температуры, давления, расхода газа и другие параметры. Эти данные обрабатываются согласно правилам обработки состояния датчиков для определения текущего состояния системы. Информация о состоянии датчиков используется для мониторинга процесса отжига и выявления любых отклонений от нормальных параметров работы.

Установка и поддержание параметров: в этот блок поступают обработанные параметры из блока определения параметров и данные о текущем состоянии системы из блока определения состояния датчиков. На основе этой информации система автоматически устанавливает и поддерживает необходимые параметры для процесса отжига. Это включает регулирование температуры, подачу газа и других технологических параметров для обеспечения оптимальных условий отжига металла. Система также осуществляет мониторинг и контроль за поддержанием установленных параметров, обеспечивая стабильность и эффективность процесса.

Таким образом, декомпозиция функциональной модели IDEF3 описание основных этапов работы автоматизированной системы управления электрической печи отжига металла. Это помогает в структурировании и понимании всех процессов, их взаимосвязей и последовательности выполнения для достижения оптимальных результатов в технологическом процессе отжига металла.

### 2.3.3 Функциональная модель DFD

При разработке функциональной модели системы альтернативой методологии IDEF0 является методология графического структурного анализа, известная как диаграмма потоков данных (DFD). Она предоставляет описание внешних источников и адресатов данных, логических функций, потоков данных и хранилищ данных, к которым осуществляется доступ.

Модель системы в нотации DFD [5] представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм, каждая из которых является единицей описания системы и размещается на отдельном листе. Эта модель включает контекстную диаграмму и диаграммы декомпозиции.

Принципы построения функциональной модели с помощью DFD аналогичны принципам методологии IDEF0. Вначале создается контекстная диаграмма, на которой отображаются взаимосвязи системы с внешним окружением. Затем осуществляется декомпозиция основных процессов и подсистем с построением иерархии диаграмм.

Центральной и единственной работой проектируемой системы является ав-

томатизированная система управления электрической печью для отжига металла. На ее вход поступают уставки оператора и сигналы с датчиков, а на выходе происходит формирование управляющего сигнала для исполнительных механизмов, а также сохранение данных в архив энергоучета, трендов, рецептов и ошибок.

Система накапливает данные путем обработки команд и параметров от пользователя и контроллера, применения правил обработки данных, а также управления процессом отжига металла. Помимо этого, непрерывно ведется накопление и хранение информации о возникающих неполадках или аварийных ситуациях системы электрической печи отжига металла, а также аналитических данных, которые записываются в архив трендов.

В качестве хранилища данных в системе используется база данных, где сохраняются тренды.

Общая функциональная диаграмма DFD управления электрической печью отжига металла представлена на рисунке 2.7.

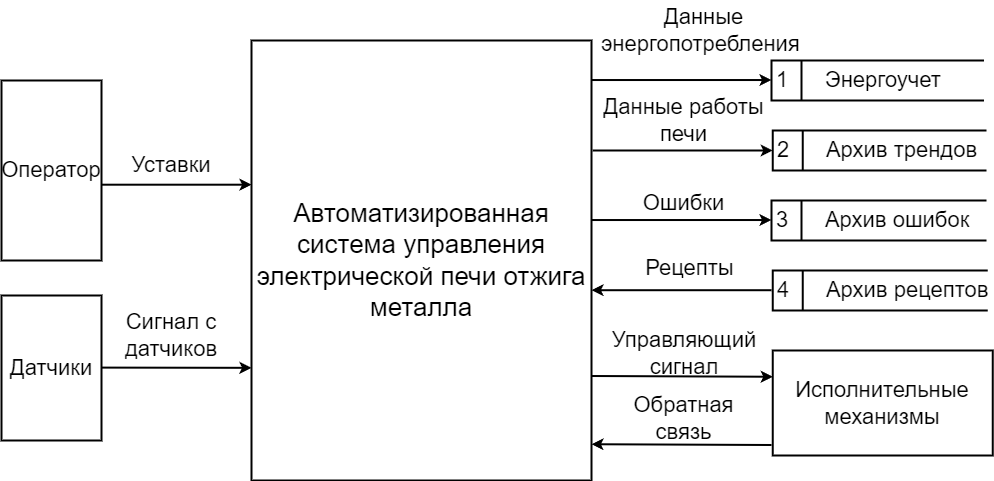


Рисунок 2.6 – Функциональная диаграмма DFD

Из выделенных процессов и хранилищ данных следует провести более детальное рассмотрение каждого компонента и его функций, а также определить потоки данных. Декомпозированная функциональная диаграмма DFD представлена на рисунке 2.7.

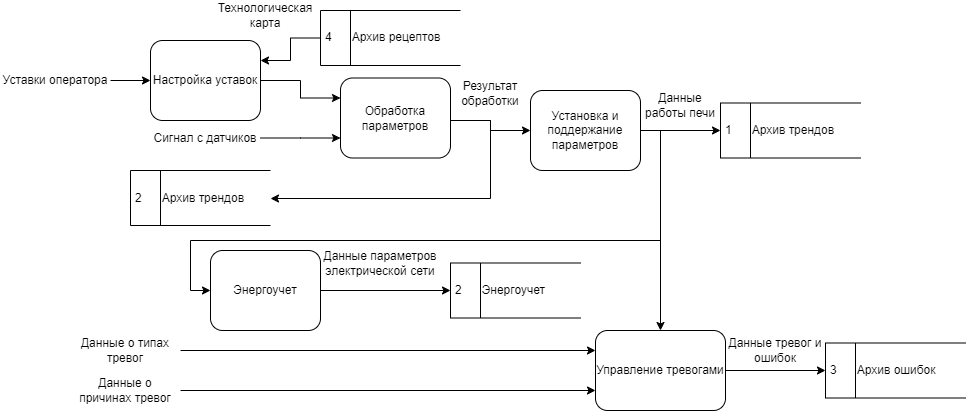


Рисунок 2.7 – Декомпозиция функциональной диаграммы DFD

Декомпозиция функциональной модели DFD для автоматизированной системы управления электрической печи отжига металла позволяет описать отдельные процессы и их взаимодействия. В данном случае рассмотрим основные функции и потоки данных, включенные в декомпозированную модель.

Настройка уставок: оператор задает уставки, определяя целевые параметры процесса отжига металла, такие как температура, время, давление и расход газа. Эти уставки вводятся в систему и передаются в блок обработки параметров.

Обработка параметров: в этом блоке система обрабатывает полученные уставки и сравнивает их с текущими параметрами, поступающими с датчиков. Обработка параметров включает в себя проверку их корректности, соответствия технологическим требованиям и возможность внесения корректив для обеспечения оптимального процесса отжига. Обработанные параметры передаются в блок установки и поддержания параметров.

Установка и поддержание параметров: на основе обработанных параметров система управляет процессом отжига, регулируя работу исполнительных механизмов. Это включает в себя регулирование температуры, подачи газа и других критических параметров для поддержания стабильных условий отжига. Блок установки и поддержания параметров также взаимодействует с архивами данных для записи и анализа параметров.

Энергоучет: система энергоучета собирает данные о потреблении энергии в процессе отжига. Эти данные используются для анализа энергоэффективности системы и выявления возможностей для оптимизации энергозатрат. Информация о потреблении энергии передается в соответствующий архив для последующего анализа.

Архив трендов: архив трендов хранит временные ряды данных, собранных с датчиков и системы энергоучета. Это включает в себя данные о температуре, давлении, расходе газа и потреблении энергии. Эти данные используются для анализа поведения системы во времени, выявления закономерностей и оптимизации процесса отжига.

Архив рецептов: архив рецептов хранит наборы уставок для различных режимов отжига металла. Оператор может выбирать нужный рецепт, который затем передается в блок настройки уставок.

Архив ошибок: архив ошибок фиксирует все отклонения и аварийные ситуации, произошедшие в процессе работы системы. Данные об ошибках и неисправностях оборудования сохраняются для последующего анализа и устранения причин сбоев. Информация из этого архива помогает улучшить надежность и стабильность системы.

Управление тревогами: система управления тревогами мониторит работу системы и в случае обнаружения отклонений или неисправностей генерирует соответствующие предупреждения и тревоги. Эти тревоги передаются оператору для принятия оперативных мер по устранению проблем. Данные о тревогах также сохраняются в архиве ошибок.

Исполнительные механизмы: выполняют команды, поступающие от блока установки и поддержания параметров. Это включает в себя управление нагревателями, клапанами для подачи газа и другими устройствами, обеспечивающими поддержание необходимых условий отжига.

Таким образом, декомпозиция функциональной модели DFD описывает процессы и потоки данных между различными компонентами системы управления электрической печи отжига металла.

## 2.4 Разработка моделей данных

### 2.4.1 Логическая модель IDEF1x

Логическая модель данных определяет концепции, взаимосвязи и ограничения данных в определенной области знаний. Она состоит из сущностей, их атрибутов и связей. Каждый атрибут представляет собой характеристику сущности, описывающую её свойства, в то время как связи отображают логические взаимосвязи между сущностями.

Логическая модель представлена на рисунке 2.8.

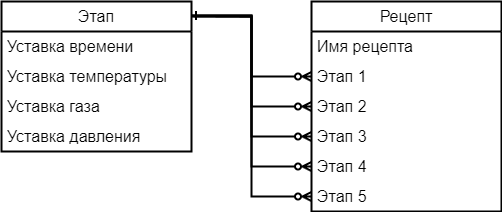


Рисунок 2.8 – Логическая модель данных

Описание отношений логической модели системы и их атрибутов представлено в таблице 2.1.

Таблица 2.1 – Отношения и атрибуты логической модели

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Ограничения |
| Этап | | |
| Номер этапа | Целое | PRIMARY KEY, NOT NULL |
| Уставка времени | Реальное | NOT NULL |
| Уставка температуры | Реальное | NOT NULL |
| Уставка газа | Реальное | NOT NULL |
| Уставка давления | Реальное | NOT NULL |
| Рецепт | | |
| Имя рецепта | Текст | NOT NULL |
| Этап 1 | Целое | FOREIGN KEY, NOT NULL |
| Этап 2 | Целое | FOREIGN KEY, NOT NULL |
| Этап 3 | Целое | FOREIGN KEY, NOT NULL |
| Этап 4 | Целое | FOREIGN KEY, NOT NULL |
| Этап 5 | Целое | FOREIGN KEY, NOT NULL |

Логическая модель IDEF1x для системы управления электрической печи отжига металла описывает структуру данных и взаимосвязи между ключевыми сущностями, которые участвуют в управлении процессом отжига. На диаграмме представлены две основные сущности: Этап и Рецепт. Сущность Этап включает в себя атрибуты, такие как уникальный идентификатор этапа, название этапа, порядок выполнения и параметры этапа. Уникальный идентификатор этапа служит ключевым полем, которое однозначно определяет каждый этап процесса отжига. Название этапа представляет собой описание процесса, происходящего на данном этапе, и помогает операторам и системе понимать, какой именно процесс осуществляется. Порядок выполнения указывает на последовательность выполнения этапов, что важно для правильного выполнения технологического процесса. Параметры этапа включают целевые значения температуры, времени выдержки, скорости нагрева или охлаждения, которые должны быть достигнуты или поддержаны на данном этапе, обеспечивая необходимые условия для успешного выполнения этапа.

Сущность Рецепт включает в себя атрибуты, такие как уникальный идентификатор рецепта, название рецепта, описание рецепта и набор этапов. Уникальный идентификатор рецепта однозначно определяет каждый рецепт, используемый в процессе отжига металла, и является ключевым полем для идентификации рецептов. Название рецепта представляет собой описание процесса отжига для определенного типа металла или условий обработки, что помогает быстро идентифицировать нужный процесс. Описание рецепта включает детальное описание процесса, последовательность этапов и параметры, необходимые для достижения требуемых характеристик металла. Набор этапов представляет собой список этапов, составляющих данный рецепт, и включает ссылки на этапы и их порядок выполнения в рамках рецепта.

Логическая модель устанавливает связь один ко многим между сущностями Этап и Рецепт. Это означает, что один рецепт может состоять из нескольких этапов, тогда как каждый этап может принадлежать только одному рецепту. Такая структура позволяет организовать данные таким образом, чтобы поддерживать сложные процессы отжига, состоящие из нескольких этапов. В процессе управления отжигом металла оператор выбирает конкретный рецепт, соответствующий типу обрабатываемого металла и требуемым характеристикам. Система, используя структуру данных, представленную на диаграмме, идентифицирует все этапы, связанные с выбранным рецептом, и последовательно выполняет их, обеспечивая соблюдение параметров на каждом этапе.

### 2.4.2 Физическая модель IDEF1x

Физический уровень данных представляет собой конкретное отображение логической модели данных в системном каталоге, которое тесно связано с реализацией системы управления базой данных. На этом уровне объекты базы данных могут иметь имена, соответствующие требованиям и ограничениям конкретной системы управления базой данных.

Физическая модель данных представлена на рисунке 2.9.

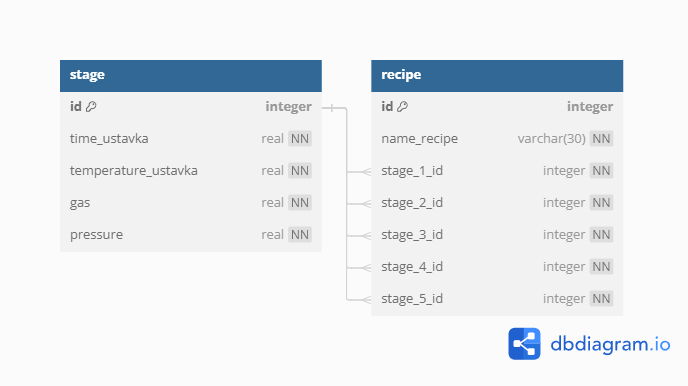


Рисунок 2.9 - Физическая модель данных IDEF1x

В таблице 2.2 приведены данные, которые были введены для описания полей созданных таблиц.

Таблица 2.2 – Описание физической модели

|  |  |
| --- | --- |
| Атрибут | Тип |
| 1 | 2 |
| Stage | |
| Id | integer |
| time\_ustavka | real |
| temperature\_ustavka | real |
| gas | real |

Продолжение таблицы 2.2

|  |  |
| --- | --- |
| 1 | 2 |
| pressure | real |
| recipe | |
| name\_recipe | varchar(30) |
| stage\_1\_id | integer |
| stage\_2\_id | integer |
| stage\_3\_id | integer |
| stage\_4\_id | integer |
| stage\_5\_id | integer |

Физическая модель данных IDEF1x для системы управления электрической печи отжига металла представляет собой структуру базы данных, описывающую взаимосвязи между сущностями Этап и Рецепт. Эта модель помогает в организации хранения данных и обеспечивает управление процессом отжига.

Сущность stage включает в себя несколько атрибутов: уникальный идентификатор этапа (id), временную уставку, температурную уставку, параметр подачи газа и параметр давления. Эти атрибуты определяют условия, необходимые для успешного выполнения каждого этапа процесса отжига.

Сущность recipe включает в себя уникальный идентификатор рецепта, название рецепта, а также идентификаторы пяти этапов. Эти атрибуты связывают каждый рецепт с конкретными этапами, составляющими полный процесс отжига. Взаимосвязь между сущностями stage и recipe представляет собой связь один ко многим, что означает, что один рецепт может включать несколько этапов, в то время как каждый этап может быть частью только одного рецепта. В данной модели каждый рецепт состоит из пяти этапов, идентификаторы которых хранятся в полях stage\_1\_id, stage\_2\_id, stage\_3\_id, stage\_4\_id и stage\_5\_id. Эти поля ссылаются на соответствующие записи в таблице stage, что позволяет связать каждый рецепт с конкретными этапами.

При выполнении процесса отжига металла оператор выбирает определенный рецепт, соответствующий типу обрабатываемого металла и требуемым условиям. Система, используя структуру базы данных, идентифицирует все этапы, связанные с выбранным рецептом, и последовательно выполняет их, следуя установленным параметрам для каждого этапа. Это обеспечивает соблюдение технологического процесса, что важно для достижения необходимых механических свойств и качества металла.

## 2.5 Вывод к разделу 2

Во второй главе определены ключевые принципы и подходы для автоматизации системы управления электрической печью для отжига металла. Основные требования включают функциональность, надежность, производительность, масштабируемость и удобство пользовательского интерфейса. Выбран программируемый логический контроллер (ПЛК) с высокой производительностью, надежностью, модульностью и совместимостью с различными протоколами связи.

Структурная схема системы представлена в виде блоков, отражающих основные элементы и их взаимосвязи. Графические интерфейсы, такие как web SCADA и HMI-дисплей, обеспечивают удобное управление и мониторинг системы. Средства доступа к данным включают использование протокола OPC UA, баз данных InfluxDB и PostgreSQL, что обеспечивает надежное хранение, быстрый доступ и эффективный анализ данных.

Реализация этих методов и подходов обеспечит высокую степень автоматизации, надежность и эффективность системы управления процессом отжига металла.

В разделе проектирования информационной системы представлены ключевые этапы проектирования системы, начиная с разработки функциональных моделей и заканчивая созданием моделей данных и использованием диаграмм UML для анализа и проектирования.

Разработаны функциональные модели IDEF0, IDEF3 и DFD. Эти модели позволяют описать функциональные аспекты системы с различных точек зрения, что помогает понять ее работу и взаимодействие компонентов.

Разработаны модели данных, включая логическую и физическую модели IDEF1x. Логическая модель помогает структурировать данные в системе на высоком уровне абстракции, а физическая модель представляет собой конкретную реализацию данных в базе данных.

# 3 Автоматизированная система

## 3.1 Анализ функциональной схемы автоматизации

Функциональная схема разрабатывается на основании исходных материалов по созданию АСУТП и в первую очередь материалов технологического регламента или отдельных документов, включаемых в технологический регламент.

### 3.1.1 Функциональная схема АСУТП

Функциональная схема автоматизации [12] (ФСА) процесса отжига металла в электрической муфельной печи представляет собой систему, включающую различные датчики, контроллеры и исполнительные механизмы, обеспечивающие управление и контроль за технологическим процессом. ФСА представлена в приложении Л.

Электрическая муфельная печь: это центральный элемент системы, где происходит процесс отжига металла. Внутри печи установлены нагревательные элементы, контролирующие температуру на разных зонах.

Датчики температуры (TE): TE 1a, TE 2a, TE 3a и TE 5a расположены в зонах нагрева печи и измеряют текущую температуру в этих зонах. Они передают данные о температуре на контроллер для мониторинга и регулирования нагрева.

Датчики расхода газа (FEI): FEI 4a и PEI 6a измеряют расход защитного газа, поступающего в печь. Эти данные также передаются на контроллер для управления подачей газа.

Установка защитного газа: обеспечивает подачу инертного газа внутрь печи для создания защитной атмосферы, предотвращающей окисление металла.

Контроллер: центральный управляющий элемент, обрабатывающий данные с датчиков и управляющий исполнительными механизмами. Контроллер получает данные о температуре, расходе газа и других параметрах, обрабатывает их и отправляет команды на исполнительные механизмы для регулирования процесса.

Исполнительные механизмы: включают в себя клапаны для регулирования подачи газа и нагревательные элементы. Эти механизмы управляются контроллером на основе данных, полученных от датчиков, и обеспечивают поддержание заданных параметров процесса отжига.

Приборы местные: TT 3в, NS и FT 4в, PT 6в отображают текущие параметры процесса (температура, скорость потока, давление) и используются для локального контроля и управления.

Щит управления: включает контроллер и интерфейсы для взаимодействия оператора с системой. Оператор может вводить уставки, контролировать текущие параметры и управлять процессом через щит управления.

В процессе отжига металла оператор устанавливает необходимые параметры через щит управления. Контроллер получает эти уставки и начинает процесс, регулируя нагревательные элементы в различных зонах печи на основе данных, получаемых с температурных датчиков. Датчики температуры (TE) измеряют текущие значения температуры и передают их на контроллер, который сравнивает их с уставками и, при необходимости, корректирует мощность нагревательных элементов для поддержания заданной температуры.

Одновременно с этим, установка защитного газа подает инертный газ в печь. Датчик расхода газа (FEI) измеряют объем подаваемого газа и передают эти данные на контроллер. Контроллер регулирует клапаны подачи газа, обеспечивая необходимый уровень защитной атмосферы внутри печи. Датчик давления (PEI) измеряет уровень давления в печи, эти данные передаются на контроллер и если давление превышает уставку, то часть давления выводится.

### 3.1.2 Схема информационных потоков

С учетом описания моделируемого процесса, приведенного в предыдущих разделах, можно выделить и классифицировать потоки информации в системе. Обычно помимо деления на входные и выходные потоки делятся еще на дискретные и аналоговые. Схема информационных потоков для взаимодействия АСУТП с разрабатываемой системой приведена на рисунке 3.1.



Рисунок 3.1 – Схема информационных потоков

Схема информационных потоков для системы управления электрической печи отжига металла описывает передачу данных между различными компонентами системы, включая датчики, контроллеры и исполнительные механизмы. Основные информационные потоки в системе включают обратную связь с контакторами, данные о температуре из зон нагрева, уровень газа и уровень давления, а также управление исполнительными механизмами. Рассмотрим подробнее каждый из потоков.

Обратная связь с контакторами: контакторы 1, 2, 3 и 4 передают обратную связь в систему управления. Это позволяет контроллеру получать информацию о состоянии контакторов и корректировать работу нагревательных элементов. Обратная связь с каждого контактора обеспечивает точное управление и защиту оборудования от перегрузок.

Управление контакторами ТЭНов: управление контакторами ТЭНов 1, 2, 3 и 4 осуществляется на основе данных, полученных от датчиков температуры и других параметров. Контроллер посылает команды для включения или отключения контакторов, что позволяет регулировать нагрев в различных зонах печи.

Температура с зон нагрева: датчики температуры, установленные в зонах нагрева 1, 2, 3 и 4, передают данные о текущей температуре в контроллер. Эти данные используются для мониторинга и регулирования процесса отжига, обеспечивая поддержание заданных температурных режимов в каждой зоне.

Уровень газа: датчик уровня газа передает данные в систему управления, обеспечивая контроль и регулирование подачи защитного газа в печь. Это необходимо для поддержания необходимого состава атмосферы внутри печи и предотвращения окисления металла.

Уровень давления: датчик уровня давления передает данные о текущем давлении в системе. Эти данные используются для контроля и регулирования давления в печи, обеспечивая безопасные и стабильные условия работы.

Управление заслонками защитного газа: управление заслонками защитного газа осуществляется на основе данных о температуре и уровне газа. Контроллер посылает команды для открытия или закрытия заслонок, регулируя подачу газа и поддерживая оптимальные условия внутри печи.

Управление заслонкой спуска давления: управление заслонкой спуска давления осуществляется на основе данных о температуре и уровне давления. Контроллер посылает команды для открытия или закрытия заслонки, обеспечивая сброс избыточного давления и поддержание безопасного уровня внутри печи.

Эта схема информационных потоков иллюстрирует, как данные передаются между различными компонентами системы и как эти данные используются для управления процессом отжига металла.

## 3.2 Анализ комплекса технических средств

### 3.2.1 Выбор ПЛК

Согласно требованиям к проекту следует выбрать ПЛК.

В рамках данной работы рассматривается ПЛК Fastwel CPM723-01 (рисунок 3.2), также были проведены сравнения с ОВЕН ПЛК210 (рисунок 3.3) и Siemens s7-1200 (рисунок 3.4).



Рисунок 3.2 – Fastwel CPM723-01

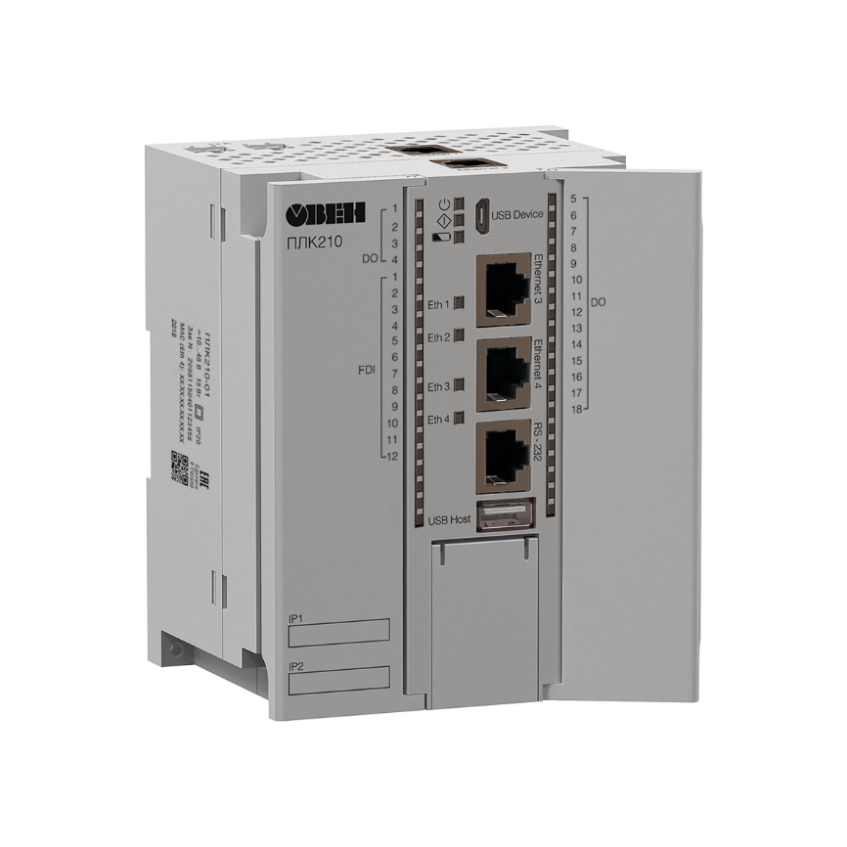


Рисунок 3.3 – Овен ПЛК 210



Рисунок 3.4 - Siemens s7-1200

В таблице 3.1 приведены технические характеристики рассматриваемых ПЛК.

Таблица 3.1 – Рассматриваемые ПЛК

|  |  |  |  |
| --- | --- | --- | --- |
| Параметры | Fastwel CPM723 | ОВЕН ПЛК210 | Siemens s7-1200 |
| Интерфейсы | Ethernet | RS-485, RS-232, USB, Ethernet | Ethernet |
| Поддержка протоколов | Modbus TCP, Modbus RTU/ASKII, Протокол OPC UA, Протокол FTP, HTTP, DHCP, NTP, IEEE | Modbus TCP, Протокол MQTT, Протокол NTP, Протокол OPC UA, Протокол RSTP | Modbus TCP, Протокол OPC UA |
| Основная среда разработки | Codesys v3 | Codesys v3 | TIA Portal Step-7 |
| Дискретные и аналоговые входы-выходы | Способен обслуживать до 64 каналов | Способен обслуживать до 30 каналов | Способен обслуживать до 284 каналов |
| Доступность в России | Да | Да | Нет |
| Цена, руб. | 60 000 | 64 600 | 30 000 |
| Наработка на отказ, ч | 500 000 | 60 000 | 50 000 |

Контроллер Fastwel CPM723-01 оснащен мощным процессором, который обеспечивает высокую производительность при низком энергопотреблении. Среднее время наработки на отказ составляет 500000 часов, что указывает на его высокую надежность и долговечность. Это особенно важно для промышленных приложений, где недопустимы частые сбои и необходимо поддерживать непрерывную работу системы.

Fastwel CPM723-01 поддерживает множество промышленных протоколов, включая Modbus TCP/RTU, FTP, HTTP и МЭК 60870-5-104. Контроллер имеет два порта Ethernet 10/100 Мбит/с, что позволяет организовать отказоустойчивую сеть кольцевой топологии без дополнительного оборудования. Это обеспечивает гибкость и надежность при интеграции в различные промышленные сети и топологии.

Контроллер поддерживает до 64 периферийных модулей на локальной шине FBUS и до 128 модулей на удалённой шине FBUS. Это могут быть как аналоговые, так и дискретные модули ввода-вывода, что позволяет легко масштабировать систему под конкретные задачи и требования. Такая возможность расширения делает Fastwel CPM723-01 универсальным решением для различных автоматизированных систем.

Контроллер поддерживает среду разработки приложений CODESYS V3, которая является мощным инструментом для разработки и отладки приложений. Поддержка новых типов данных и языковых конструкций стандарта IEC 61131-3:2013 сокращает время разработки и повышает надежность создаваемых приложений. Это позволяет инженерам быстро создавать и настраивать системы управления, а также упрощает интеграцию контроллера с оборудованием других производителей.

Контроллер оснащен энергонезависимой памятью объемом 131056 байт и поддерживает съемные накопители типа MicroSD с объемом до 32 Гбайт. Это обеспечивает сохранность данных даже при отключении питания и позволяет хранить большие объемы информации для последующего анализа и отчетности.

Выбор контроллера Fastwel CPM723-01 обоснован его высокой производительностью, надежностью, широкими коммуникационными возможностями, поддержкой большого количества периферийных модулей, устойчивостью к экстремальным условиям эксплуатации, удобством разработки и интеграции, а также расширенными возможностями хранения данных.

### 3.2.2 Выбор модулей для ПЛК и измерительного преобразователя

Для корректной работы системы необходимо выбрать модули ввода-вывода, которые обеспечат сбор данных с датчиков и управление исполнительными устройствами. Пример модуля аналогового ввода представлен на рисунке 4.5.



Рисунок 3.5 – Модуль аналогового ввода

Модуль аналогового ввода AIM724 используются для подключения датчиков температуры. Имеет два дифференциальных канала аналогового ввода и предназначен для измерения температуры при помощи термопар или напряжения постоянного тока малой величины.

Модуль аналогового ввода AIM723 используются для подключения датчиков расхода газа и давления. Имеет четыре однопроводных канала аналогового ввода сигналов постоянного тока в диапазоне от 4 до 20 мА, реализованных на базе дельта-сигма АЦП.

Модули аналогового вывода AIM726 применяются для управления заслонкой защитного газа и заслонкой контроля давления. Модуль имеет два канала для измерения напряжения постоянного тока в диапазоне от 0 до 40 вольт. Режим измерения – однопроводный.

Модуль дискретного ввода DIM717 предназначен для приема обратной связи контактора. Модуль предназначен для приема дискретных сигналов датчиков и содержит восемь однопроводных каналов дискретного ввода с общим «минусом» (положительной логикой) и полевым питанием 24 В постоянного тока.

Модуль дискретного вывода DIM718 предназначен для управления контактором. Предназначен для коммутации нагрузок на общий (минусовый) провод полевого питания при напряжении полевого питания 24 В постоянного тока. Имеет восемь каналов дискретного вывода с максимальным током нагрузки 0,5 А и с защитой от короткого замыкания.

Для измерения параметров электрической сети выбран ЭНИП-2. Устройство представляет собой многофункциональный измерительный преобразователь, разработанный для использования в системах телемеханики, автоматизации технологических процессов и автоматизированных системах управления подстанциями и электростанциями. Его ключевой особенностью является сочетание высокой скорости обработки данных и точности измерений, что позволяет обеспечивать синхронизированные измерения параметров режима электрической сети. Это способствует повышению наблюдаемости и управляемости электрическими сетями, а также реализации распределенных автоматизированных систем управления и регулирования. Устройство представлено на рисунке 4.6.



Рисунок 3.6 – Устройство ЭНИП-2

Поддерживаются стандартные протоколы передачи данных, такие как Modbus RTU/TCP, ГОСТ Р МЭК 60870-5-101-2006, ГОСТ Р МЭК 60870-5-104-2004 и МЭК 61850-8-1.

ЭНИП-2 проводит измерения различных параметров электрической сети, включая напряжение, ток, мощность, частоту, коэффициенты несимметрии и искажения синусоидальности. Он также осуществляет сигнализацию о состоянии сети и может управлять коммутационными аппаратами через встроенные дискретные выходы.

Система управления электрической печи отжига металла должна обеспечивать определенный объем обработки информации по входам и выходом, эта информация представлена в таблице 3.2.

Таблица 3.2 – Входы-выходы автоматизированной системы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Тип сигнала | Наименование параметра | Тип канала | Модуль ввода-вывода | Диапазон | Ед. измерения | Сигнализация предупреждения (min, max) |
|  |  |  |  |  |  |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| AI | Температура с зоны нагрева 1 | – | AIM724 | Тип K | ℃ | ±50 ℃ от уставки |
| AI | Температура с зоны нагрева 2 | – | AIM724 | Тип K | ℃ | ±50 ℃ от уставки |
| AI | Температура с зоны нагрева 3 | – | AIM724 | Тип K | ℃ | ±50 ℃ от уставки |
| AI | Температура с зоны нагрева 4 | – | AIM724 | Тип K | ℃ | ±50 ℃ от уставки |
| AI | Датчик газа | – | AIM723 | 4 – 20 мА | м³/ч | ±5 м³/ч от уставки |

Продолжение таблицы 3.2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| AI | Датчик давления | – | AIM723 | 4 – 20мА | Па | ±20 Па от уставки |
| DI | Обратная связь контактора 1 | – | DIM717 | -3…+5 – лог. 0, 15…30 – лог. 1 | Бит | ¬ уставка |
| DI | Обратная связь контактора 2 | – | DIM717 | -3…+5 – лог. 0, 15…30 – лог. 1 | Бит | ¬ уставка |
| DI | Обратная связь контактора 3 | – | DIM717 | -3…+5 – лог. 0, 15…30 – лог. 1 | Бит | ¬ уставка |
| DI | Обратная связь контактора 4 | – | DIM717 | -3…+5 – лог. 0, 15…30 – лог. 1 | Бит | ¬ уставка |
| AO | Управление заслонкой защитного газа | – | AIM726 | 0 – 40В | % | – |
| AO | Управление заслонкой контроля давления | – | AIM726 | 0 – 40В | % | – |
| DO | Управление контактором тэна 1 | – | DIM718 | 20,4…28,8В | Бит | – |
| DO | Управление контактором тэна 2 | – | DIM718 | 20,4…28,8В | Бит | – |

Продолжение таблицы 3.2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| DO | Управление контактором тэна 3 | – | DIM718 | 20,4…28,8В | Бит | – |
| DO | Управление контактором тэна 4 | – | DIM718 | 20,4…28,8В | Бит | – |
| AI | ЭНИП-2 1 | Modbus TCP | – | – | – | – |
| AI | ЭНИП-2 2 | Modbus TCP | – | – | – | – |
| AI | ЭНИП-2 3 | Modbus TCP | – | – | – | – |
| AI | ЭНИП-2 4 | Modbus TCP | – | – | – | – |

Модули входов и выходов системы управления электрической печью отжига металла представляют собой точки взаимодействия системы с внешними элементами, такими как датчики и исполнительные механизмы. Эти взаимодействия обеспечивают контроль параметров процесса и отправку управляющих сигналов для выполнения заданных операций.

## 3.3 Разработка алгоритмов программ

Концепция работы автоматизированной системы управления технологическим процессом описывает последовательность шагов, начиная от запуска системы до ее управления в процессе работы. На рисунке 3.7 представлена схема алгоритма этого производства.

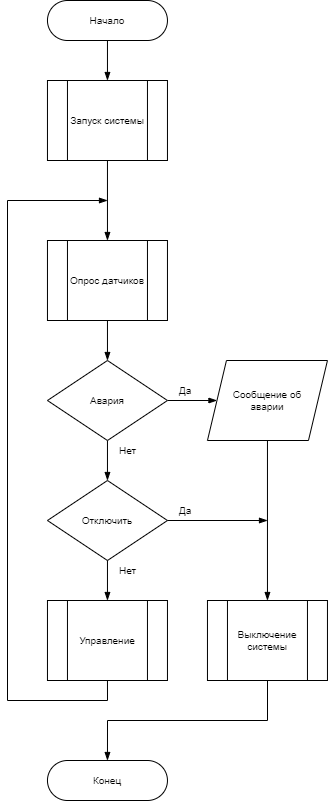


Рисунок 3.7 – Схема алгоритма производства

Система запускается в обратном порядке от производственного процесса и затем входит в цикл работы, включающий опрос датчиков, проверку на аварии и управление системой. Если обнаружены проблемы, система выводит сообщение об аварии и выключается. В противном случае она продолжает работу и переходит к управлению. Схема алгоритма декомпозиции блока управления представлена на рисунке 3.8.



Рисунок 3.8 – Декомпозиция блока управления

Блок управления представлен модулями подачи газа, поддержания давления и нагрева, а также модуль контроля временных уставок и модуль энергоучета.

### 3.3.1 Алгоритм управления газоподачей

Далее следует произвести декомпозицию блока газоподачи (рисунок 3.9).

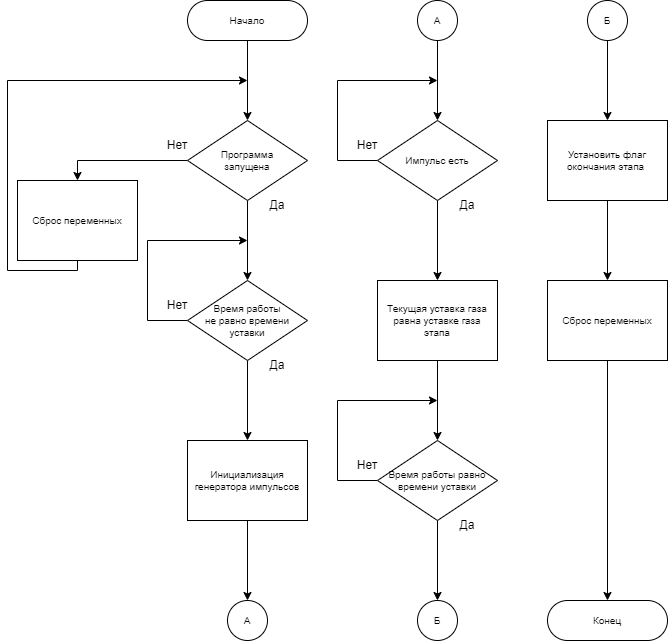


Рисунок 3.9 – Декомпозиция блока газоподача

Алгоритм начинается с инициализации процесса и сброса всех переменных в их начальные значения. Далее происходит проверка состояния программы: если программа не запущена, алгоритм возвращается к началу. В случае если программа запущена, осуществляется проверка наличия импульса. Если импульс отсутствует, алгоритм также возвращается к началу. При наличии импульса проверяется, соответствует ли текущее время работы установленному времени уставки. Если время работы не равно времени уставки, алгоритм переходит к следующему этапу.

На этом этапе осуществляется проверка соответствия текущей уставки газа уставке для данного этапа. Если уставки не совпадают, происходит возврат к началу алгоритма. Если же уставки совпадают, запускается генератор импульсов и происходит переход к следующей проверке времени работы. Если время работы снова не равно времени уставки, алгоритм возвращается к предыдущему этапу для повторной проверки уставки газа.

Когда время работы соответствует времени уставки, устанавливается флаг, сигнализирующий об окончании этапа, и все переменные снова сбрасываются в начальные значения. Таким образом, завершается текущий этап процесса. Алгоритм заканчивается завершением процесса, если все условия выполнены.

Этот алгоритм обеспечивает последовательное выполнение этапов процесса, управляя состоянием программы, генерацией импульсов и контролем времени работы, что позволяет корректно управлять процессом и завершать этапы по мере достижения установленных уставок.

### 3.3.2 Алгоритм управления нагревом

Далее представлена декомпозиция блока нагрева, который представлен на рисунке 3.10.

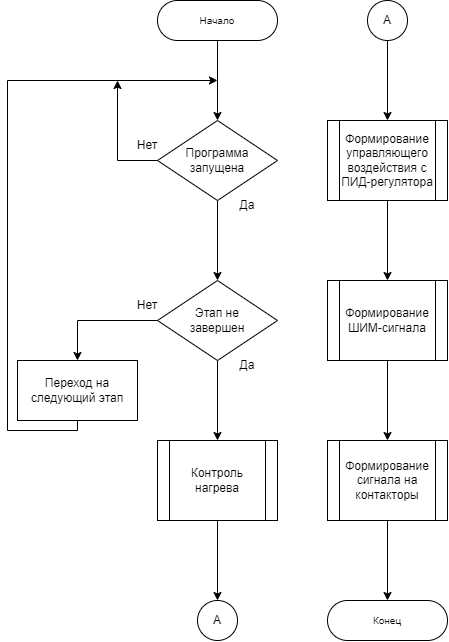


Рисунок 3.10 – Декомпозиция блока нагрева

Алгоритм управления процессом нагрева в электрической печи отжига металла представленный на схеме алгоритма включает несколько этапов, обеспечивающих последовательное выполнение технологического процесса. Рассмотрим каждый элемент алгоритма подробно.

Алгоритм начинается с начального этапа, обозначенного на схеме как Начало. Этот этап инициирует запуск программы управления нагревом.

Первым действием алгоритма является проверка состояния программы. Если программа запущена, то происходит переход к следующему шагу.

Контроль нагрева формирует уставки уставки температуры в зависимости от заданных оператором параметров температуры и времени конкретного этапа.

На следующем этапе происходит формирование управляющего воздействия на основе данных с ПИД-регулятора. ПИД-регулятор (пропорционально-интегрально-дифференциальный регулятор) анализирует текущую температуру и вычисляет корректирующее воздействие для поддержания заданной температуры.

Далее формируется ШИМ-сигнал (широтно-импульсная модуляция). ШИМ-сигнал используется для регулирования мощности нагревательных элементов, обеспечивая точный контроль температуры.

После формирования ШИМ-сигнала осуществляется контроль нагрева. На этом этапе контролируется работа нагревательных элементов и корректируется их мощность в соответствии с рассчитанным ПИД-регулятором воздействием. Контакторы управляют включением и выключением нагревательных элементов, обеспечивая их безопасную и эффективную работу.

Если текущий этап завершен, алгоритм переходит к следующему этапу нагрева. Это позволяет обеспечить последовательное выполнение всех стадий технологического процесса отжига.

Алгоритм управления процессом нагрева в электрической печи отжига металла представляет собой последовательное выполнение действий, направленных на поддержание заданной температуры на каждом этапе процесса. Использование ПИД-регулятора и ШИМ-сигналов позволяет обеспечить точный контроль температуры и эффективное управление нагревательными элементами.

Далее рассмотрена декомпозиция блока контроля нагрева, который представлен в приложении Д.

Алгоритм нагрева начинается с инициализации процесса и сброса всех переменных в их начальные значения. Далее проверяется, запущен ли процесс. Если процесс не запущен, алгоритм возвращается к началу. Если процесс запущен, осуществляется проверка флага установки температуры. Если флаг установки температуры не установлен, алгоритм переходит к состоянию 4, где запоминается текущее состояние. Если флаг температуры установлен, то текущая температура устанавлиавает температуре прошлого этапа. Алгоритм переходит в состояние 2.

В состоянии 2 проверяется равенство времени работы и времени уставки этапа. Если время работы равно времени уставки, то запоминается текущее состояние и происходит переход в состояние 4. Иначе инициализируется генератор импульсов и рассчитывается прирост температуры, после чего алгоритм переходит к состоянию 3. В состоянии 3 также проверяется наличие импульса, переодичность которого 1 секунда, если испульса нет, то запоминается состояние 3 и происходит переходи в состояние 4. Далее рассчитывается текущая температура, которая равна сумме текущей температуры и изменения температуры. После этого проверяется равенство времени работы и времени уставки, если условие не выполняется, то происходит переход в состояние 4, иначе переход в состояние 5.

Алгоритм завершается, когда все условия выполнены и происходит завершение процесса. Таким образом, алгоритм обеспечивает последовательное выполнение этапов процесса нагрева, управляя состоянием программы, генерацией импульсов и контролем температуры, что позволяет корректно управлять процессом нагрева.

## 3.4 Вывод к разделу 3

В данном разделе проведен анализ автоматизированной системы управления электрической печи отжига металла. Рассмотрены ключевые компоненты системы, включая контроллер, функциональную схему автоматизации, алгоритмы программ.

Контроллер Fastwel CPM723-01 выделяется своей производительностью, надежностью и широкими коммуникационными возможностями. Он поддерживает модулей, что обеспечивает гибкость в масштабировании системы. Благодаря поддержке среды разработки CODESYS V3, этот контроллер является оптимальным решением для сложных промышленных приложений.

Функциональная схема автоматизации и схема информационных потоков демонстрируют, как данные передаются между различными компонентами системы и как они используются для управления процессом отжига металла.

Разработка алгоритмов программ описывает создание алгоритмов управления, которые реализуются на контроллере. Эти алгоритмы обеспечивают поддержание заданных параметров процесса отжига, включая температуру, время выдержки и другие критические параметры. Разработка программного обеспечения в среде CODESYS V3 позволяет создавать надежные и гибкие решения для управления технологическим процессом.

# 4 Разработка программного обеспечения

Раздел разработка программного обеспечения включает в себя обоснование выбора программных продуктов для реализации системы управления процессом отжига. Для данного проекта было принято решение использовать Codesys 3.5 [8] и FUXA в качестве основных инструментов разработки и визуализации соответственно.

Codesys 3.5 выбран в качестве программного обеспечения для разработки управляющего программного кода (ПЛК) блоков и логики системы. Это интегрированная среда разработки, предоставляющая широкие возможности для создания и отладки программного кода на различных языках программирования, таких как функциональный блочный язык (FBD), структурный текст (ST), логический граф (LD) и язык непрерывных функциональных схем (CFC). Codesys 3.5 обладает высокой гибкостью и расширяемостью, что позволяет эффективно реализовывать сложные управляющие алгоритмы и обеспечивать надежное функционирование системы.

FUXA выбран в качестве программного обеспечения для визуализации процесса управления отжигом. Это веб-ориентированная SCADA-система, которая предоставляет интуитивно понятный интерфейс для мониторинга и управления процессом через веб-браузер. FUXA обладает широкими возможностями конфигурации и настройки интерфейса, что позволяет создавать графические элементы, отображающие текущее состояние процесса, а также реагировать на изменения параметров в реальном времени.

Для хранения и управления данными в системе были выбраны базы данных InfluxDB и PostgreSQL. Эти системы управления базами данных (СУБД) дополняют друг друга и обеспечивают надежное и эффективное хранение данных.

InfluxDB используется для хранения временных рядов данных, таких как тренды температур, давления и других параметров процесса отжига. Это высокопроизводительная СУБД, оптимизированная для работы с временными рядами данных. InfluxDB обеспечивает быструю запись и чтение данных, а также мощные инструменты для анализа и визуализации временных рядов, что делает её идеальным выбором для мониторинга и анализа динамических данных процесса.

PostgreSQL используется для хранения рецептов технологического процесса и данных об энергопотреблении. Это объектно-реляционная СУБД, известная своей надежностью, масштабируемостью и поддержкой сложных запросов. PostgreSQL обеспечивает высокую консистентность данных и возможность выполнения сложных аналитических задач. Хранение рецептов и данных об энергопотреблении в PostgreSQL позволяет эффективно управлять и анализировать эти данные, обеспечивая оптимизацию и контроль технологического процесса.

## 4.1 Разработка модулей управления

Разработка модулей управления является ключевым этапом в создании автоматизированной системы управления процессом отжига металла. Основной задачей данных модулей является обеспечение точного и надежного контроля за параметрами технологического процесса, такими как температура, подача газа и давление, а также управление исполнительными механизмами.

В данном разделе рассматриваются разработанные модули управления, включая использование программного ПИД-регулятора и алгоритма ШИМ (широтно-импульсной модуляции). Программный ПИД-регулятор (пропорционально-интегрально-дифференциальный регулятор) применяется для поддержания заданного уровня параметров процесса путем минимизации отклонений от уставки. Алгоритм ШИМ используется для управления мощностью, подаваемой на исполнительные механизмы, что позволяет более точно регулировать их работу и оптимизировать энергетические затраты.

Программный ПИД-регулятор является основным инструментом для поддержания стабильности и точности в системах управления. Он использует математическую модель для вычисления корректирующих воздействий на основе текущих отклонений от заданного значения. В системе ПИД-регулятор контролирует температуру в печи, обеспечивая плавное и точное управление нагревом. Благодаря этому достигается оптимальный режим работы, что важно для качества отжига металла.

Алгоритм ШИМ применяется для управления мощностью, подаваемой на исполнительные механизмы, такие как нагреватели. ШИМ позволяет регулировать среднюю мощность путем изменения соотношения времени включения и выключения сигнала, что обеспечивает более точное управление по сравнению с аналоговыми методами. В системе ШИМ используется для управления нагревательными элементами, позволяя поддерживать требуемую температуру с высокой точностью и минимальными колебаниями.

### 4.1.1 Реализация модуля нагрева

Логика нагрева реализована с помощью функционального блока HeatingControl (листинг Б.3).

Функциональный блок HeatingControl предназначен для управления процессом нагрева. Он регулирует температуру в заданных уставках и отслеживает время этапа нагрева.

Входные переменные (VAR\_INPUT)

start: BOOL – сигнал запуска процесса нагрева.

stop: BOOL – сигнал остановки процесса нагрева.

ustavka1\_temperature: REAL – первая уставка температуры (начальная температура).

ustavka2\_temperature: REAL – вторая уставка температуры (конечная температура).

ustavka\_time: REAL – время, за которое должна измениться температура от первой до второй уставки.

back\_stage\_temperature: REAL – температура на предыдущем этапе.

Выходные переменные (VAR\_OUTPUT)

actual\_temperature: REAL – текущая температура.

time\_stage\_min: REAL – время этапа нагрева в минутах.

time\_stage\_sec: REAL – время этапа нагрева в секундах.

stage\_up: BOOL := FALSE – флаг завершения этапа нагрева.

stage: BOOL – флаг текущего этапа нагрева.

current\_ustavka\_temperature: REAL – текущая уставка температуры.

Внутренние переменные (VAR)

delta\_temperature: REAL – изменение температуры за единицу времени.

debugging\_time\_divider: REAL := 30 – коэффициент для отладки времени (ускорение процесса в симуляции).

blink: Util.BLINK – экземпляр функции для создания мигания.

ustavka\_time\_temp: REAL – временная переменная для отслеживания времени уставки.

flag\_set\_temperature: BOOL := TRUE – флаг установки начальной температуры.

Логика работы: при подаче сигнала start, если flag\_set\_temperature установлен в TRUE, текущая температура (actual\_temperature) устанавливается равной температуре предыдущего этапа (back\_stage\_temperature), и флаг flag\_set\_temperature сбрасывается в FALSE. Если временная переменная ustavka\_time\_temp не равна значению, рассчитанному как ustavka\_time \* debugging\_time\_divider, начинается этап нагрева (stage := TRUE). Активируется мигающий сигнал с низким и высоким временем (blink(ENABLE := TRUE, TIMELOW := T#1000MS, TIMEHIGH := T#1MS)).

Рассчитывается изменение температуры (delta\_temperature) за единицу времени. Если сигнал blink.OUT активен (TRUE), текущая температура (actual\_temperature) увеличивается на значение delta\_temperature, обновляется время этапа в секундах и минутах, и увеличивается временная переменная ustavka\_time\_temp.

Когда временная переменная ustavka\_time\_temp достигает значения ustavka\_time \* debugging\_time\_divider, мигание отключается (blink.ENABLE := FALSE), устанавливается флаг завершения этапа (stage\_up := TRUE), сбрасывается флаг текущего этапа (stage := FALSE), и текущая уставка температуры (current\_ustavka\_temperature) обнуляется.

Далее текущая уставка температуры (current\_ustavka\_temperature) передается в функциональный блок PID.

Функциональный блок PID предназначен для реализации пропорционально-интегрально-дифференциального (PID) регулирования. Он используется для управления процессами, требующими точного поддержания заданных параметров. Текущее значение контролируемой величины сравнивается с заданным значением.

На вход функционального блока PWM (приложение В) устанавливается текущее процентное значение с ПИД-регулятора. Функциональный блок PWM предназначен для реализации широтно-импульсной модуляции (ШИМ), которая используется для управления нагрузкой путем включения и выключения выходного сигнала с определенной частотой и коэффициентом заполнения.

Далее переменная с выхода функционального блока PWM устанавливается на вход функционального блока ContactorControl, он предназначен для управления контактором, основываясь на простом входном сигнале. Логика блока реализована таким образом, что если входной сигнал km\_work установлен в TRUE, то выходной сигнал km\_out также будет установлен в TRUE, что приведет к включению контактора. Если входной сигнал km\_work установлен в FALSE, выходной сигнал km\_out также будет установлен в FALSE, что приведет к отключению контактора.

### 4.1.2 Реализация модуля газовой части

Логика заполнения защитным газом реализована с помощью функционального блока GasControl.

Блок управления подачей газа (FUNCTION\_BLOCK GasControl) предназначен для регулирования подачи газа в процессе отжига. Этот функциональный блок обеспечивает управление подачей газа на основе заданных уставок и времени, поддерживая необходимый уровень газа в течение всего процесса.

Логика работы FUNCTION\_BLOCK GasControl основана на условии запуска (входной параметр start). Когда сигнал start имеет значение TRUE, начинается процесс регулирования подачи газа.

Проверка условия времени уставки: если текущее значение временной переменной ustavka\_time\_temp не равно произведению заданного времени ustavka\_time и коэффициента debugger\_time\_divider, выполняется основной цикл регулирования подачи газа.

Инициация временной задержки: объект blink используется для создания временной задержки. Включение объекта осуществляется с параметрами TIMELOW := T#1000MS и TIMEHIGH := T#1MS.

Увеличение временной переменной: если blink.OUT имеет значение TRUE (завершение временной задержки), временная переменная ustavka\_time\_temp увеличивается на 1.

Установка текущей уставки газа: в течение выполнения цикла значение current\_ustavka\_gas устанавливается равным значению входного параметра ustavka\_gas.

Завершение этапа регулирования: если временная переменная ustavka\_time\_temp достигает значения произведения ustavka\_time и debugger\_time\_divider, процесс регулирования завершен. Объект blink отключается, флаг gas\_stage\_up устанавливается в TRUE, а значение current\_ustavka\_gas сбрасывается в 0.

Далее выход с функционального блока поступает в функциональный блок программного ПИД-регулятора, где сравнивается с сигналом с датчика. Выход с функционального блока ПИД-регулятора поступает в функциональный блок GasVentil.

Логика работы FUNCTION\_BLOCK GasVentil основана на управлении открытием и закрытием вентиля подачи газа на основе входного значения от PID-регулятора.

Расчет значения тока: Величина amperage рассчитывается как 0.16 умноженное на значение pid\_to\_processing\_gas, к результату добавляется 4.0.

Управление положением вентиля: если значение amperage больше 4, устанавливается сигнал plug\_open в TRUE, а plug\_close в FALSE, что означает открытие вентиля. Если значение amperage меньше или равно 4, устанавливается сигнал plug\_open в FALSE, а plug\_close в TRUE, что означает закрытие вентиля.

Преобразование значения тока: Величина converte\_amperage вычисляется как значение amperage минус 4.0, деленное на 0.00024414.

Преобразование значения тока в WORD: Преобразованное значение converte\_amperage приводится к типу WORD и присваивается переменной plug.

### 4.1.3 Реализация программы посредника между ПЛК и БД

Описание программного обеспечения посредника между ПЛК и БД.

Программное обеспечение посредника предназначено для обеспечения взаимодействия между программируемым логическим контроллером (ПЛК) и базой данных (БД), что позволяет автоматизировать процесс передачи данных о рецептах технологических процессов отжига металла. Это программное обеспечение выполняет следующие основные функции: чтение названия текущего рецепта из ПЛК, выбор соответствующего рецепта из базы данных, формирование объекта рецепта и отправка параметров рецепта обратно на ПЛК.

Основной скрипт запускает бесконечный цикл, в котором происходит подключение к OPC UA серверу, чтение названия текущего рецепта, выборка соответствующего рецепта из базы данных и отправка параметров рецепта обратно на ПЛК. Если рецепт успешно получен и прочитан, то данные отправляются на ПЛК. Программный код представлен в листинге Г.1.

Для подключения к базе данных PostgreSQL используется библиотека SQLAlchemy. Конфигурационные параметры, такие как пользователь, пароль, хост, порт и имя базы данных, загружаются из конфигурационного файла. В случае успешного подключения устанавливается сессия для взаимодействия с базой данных. В противном случае программа сообщает об ошибке подключения. Программный код представлен в листинге Г.2.

Модели данных для таблиц рецептов и стадий процесса определены с использованием SQLAlchemy. Таблица Stage хранит информацию о каждой стадии процесса, включая время уставки, температуру, расход газа и давление. Таблица Recipe хранит информацию о рецептах, включая имя рецепта и ссылки на пять стадий, из которых состоит рецепт. Это позволяет хранить и организовывать данные о рецептах и их стадиях в реляционной базе данных. Программный код представлен в листинге Г.3.

Для связи с OPC UA сервером используется библиотека OPC UA. Программа устанавливает соединение с сервером по указанному URL и определяет путь к узлу, откуда будут считываться данные. Это обеспечивает доступ к данным ПЛК и позволяет взаимодействовать с ними. Программный код представлен в листинге Г.4.

Для выборки рецепта из базы данных используется функция, которая выполняет запрос к базе данных для получения рецепта по его идентификатору. Это позволяет извлекать данные о рецептах и их стадиях для дальнейшего использования в системе. Программный код представлен в листинге Г.5.

Для получения названия текущего рецепта из ПЛК используется функция, которая подключается к узлу OPC UA сервера и считывает значение узла, содержащего название рецепта. Это название затем используется для поиска соответствующего рецепта в базе данных. Программный код представлен в листинге Г.6.

Для формирования объекта рецепта из данных, извлеченных из базы данных, используется функция, которая создает структуру данных, включающую идентификатор рецепта, имя и параметры стадий. Эта структура затем передается функции, отвечающей за отправку данных на ПЛК. Программный код представлен в листинге Г.7.

Для отправки параметров рецепта на ПЛК используется функция, которая обновляет значения уставок для каждой стадии процесса. Функция получает объект рецепта, извлекает из него параметры (время, температура, расход газа и давление) и отправляет их на соответствующие узлы OPC UA сервера. Это позволяет передавать данные о рецепте на ПЛК для их выполнения. Программный код представлен в листинге Г.8.

Разработанное программное обеспечение посредника между программируемым логическим контроллером и базой данных обеспечивает автоматизированное управление процессом отжига металла. Оно позволяет считывать текущий рецепт из ПЛК, извлекать соответствующие параметры из базы данных и отправлять их обратно на ПЛК для выполнения.

## 4.2 Разработка экранных форм

В разделе разработка экранных форм описывается процесс создания графических интерфейсов пользователя для системы управления электрической печи отжига металла. Экранные формы играют важную роль в обеспечении взаимодействия оператора с системой. Они предоставляют визуальные средства для мониторинга параметров, настройки уставок и управления процессом. В данном разделе рассматриваются основные принципы и этапы разработки экранных форм с использованием FUXA.

В автоматизированном рабочем месте реализована авторизация. Экранная форма представлена на рисунке 4.1.

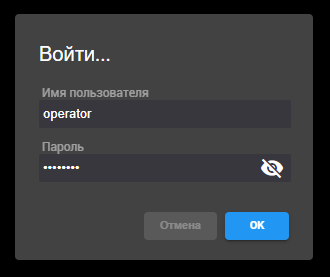


Рисунок 4.1 – Экранная форма авторизации

Форма входа предназначена для авторизации пользователей системы управления. Пользователи вводят свои учетные данные (имя пользователя и пароль) в соответствующие поля и нажимают кнопку "ОК" для подтверждения. Система проверяет введенные данные и при успешной авторизации пользователь получает доступ к функционалу системы (рисунок 4.2). В случае неправильного ввода данных система выводит соответствующее сообщение об ошибке, и пользователь может повторить попытку входа.

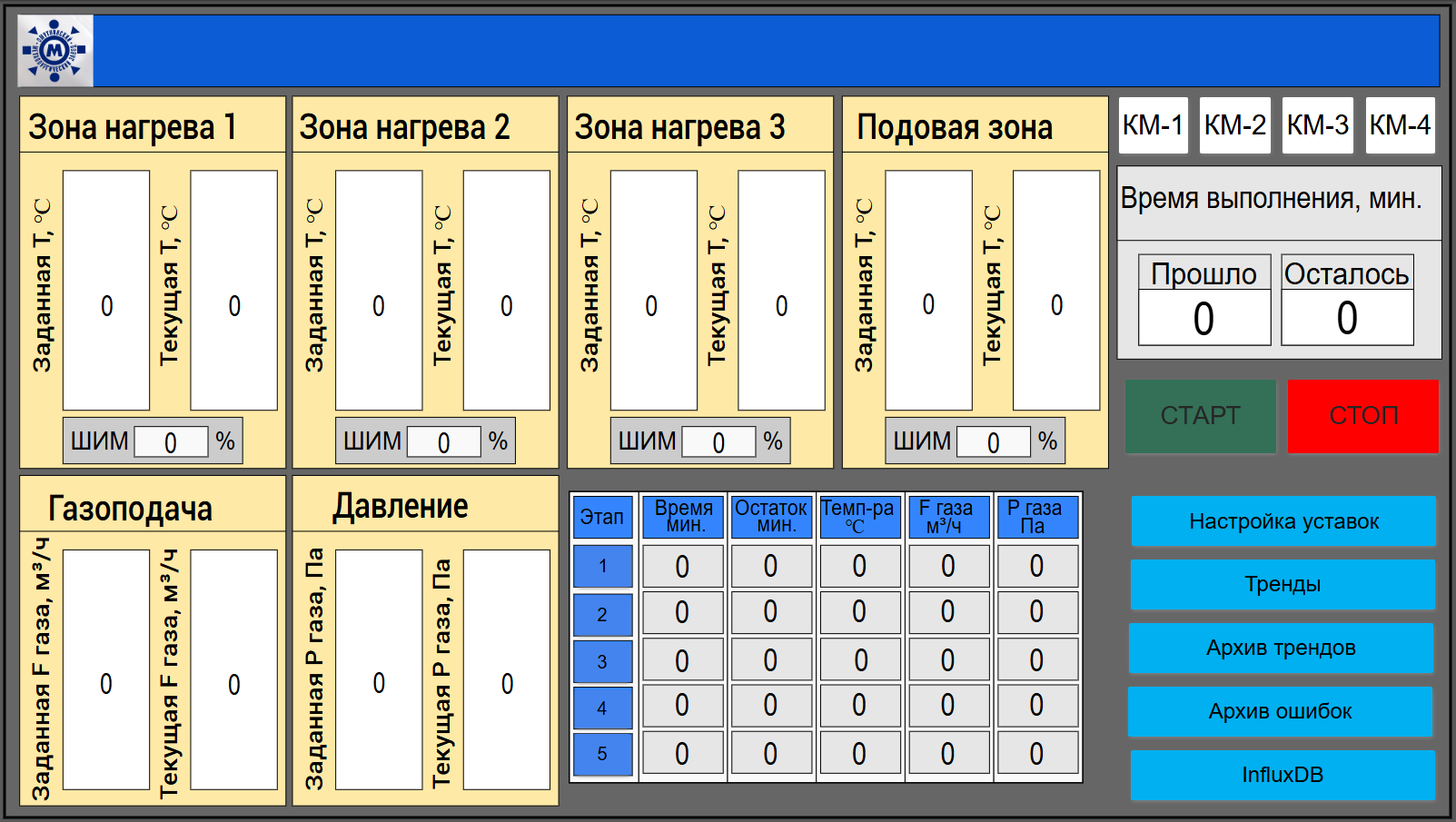


Рисунок 4.2 – Экранная форма пульт управления

Главный экран управления предоставляет оператору всю необходимую информацию для мониторинга и управления процессом отжига металла в режиме реального времени. Оператор может отслеживать текущие параметры каждой зоны нагрева, контролировать подачу газа и давление, а также управлять этапами выполнения процесса.

Зона нагрева 1, зона нагрева 2, зона нагрева 3, подовая зона: для каждой зоны нагрева отображаются заданная температура и текущая температура. Эти данные позволяют оператору контролировать и сравнивать установленные и реальные значения температур в каждой зоне печи.

Поле ШИМ показывает текущий уровень широтно-импульсной модуляции, что помогает контролировать интенсивность нагрева.

Газоподача: отображается заданный и текущий расход газа. Это позволяет контролировать подачу защитного газа в печь.

Давление: отображается заданное и текущее давление газа. Контроль давления важен для поддержания безопасных условий внутри печи.

Таблица этапов: содержит список этапов с отображением времени выполнения, остаточного времени, температуры, расхода газа и давления для каждого этапа. Это позволяет оператору следить за выполнением каждого этапа процесса отжига.

Время выполнения: отображаются прошедшее и оставшееся время выполнения текущего этапа. Это помогает оператору оценить прогресс и оставшееся время до завершения процесса.

Кнопки управления: кнопка СТАРТ используется для начала выполнения программы отжига. Кнопка СТОП позволяет прекратить выполнение процесса. Эти кнопки обеспечивают основное управление процессом.

КМ-1, КМ-2, КМ-3, КМ-4: позволяют оператору следить за включением и отключением контакторов для управления нагревом в различных зонах печи.

Настройка уставок: переход на экран настройки уставок, где оператор может задать и изменить параметры процесса отжига, с помощью ручного ввода или выборать рецепт. Экранная форма настройки уставок представлена на рисунке 4.3.

Тренды: доступ к графическим трендам, отображающим изменения параметров процесса во времени. Экранная форма представлена в приложении К.

Архив трендов: переход к архиву трендов для анализа исторических данных.

Архив ошибок: доступ к журналу ошибок для просмотра и анализа возникавших проблем.

InfluxDB: переход к панели InfluxDB для детального анализа данных и создания пользовательских отчетов. Экранная форма представлена в прило-жении Л.

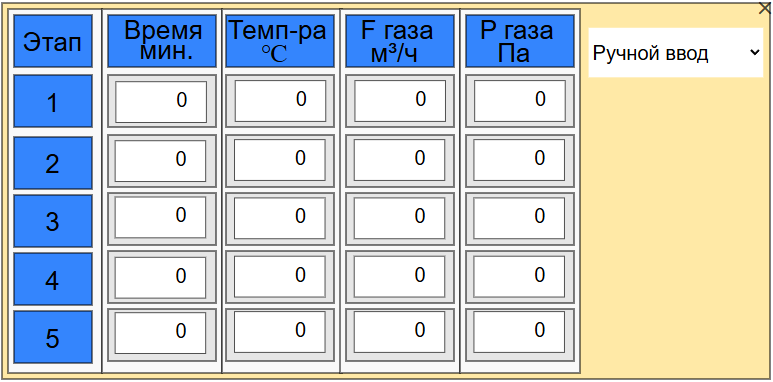


Рисунок 4.3 – Экранная форма настройки уставок

Экранная форма настройки уставок предоставляет оператору возможность задавать параметры каждого этапа процесса отжига металла, либо выбирать готовые рецепты, содержащие необходимые параметры. Оператор может вручную вводить значения времени, температуры, расхода газа и давления для каждого этапа, что позволяет гибко управлять процессом и адаптировать его под конкретные требования. Также оператор может выбрать один из заранее сохраненных рецептов, что ускоряет процесс настройки и минимизирует риск ошибок ввода.

Экранная форма трендов обеспечивает визуализацию данных, что помогает оператору контролировать и анализировать процесс отжига металла. Графики показывают текущие данные о температуре в каждой зоне нагрева, позволяя оператору быстро выявлять отклонения от нормы.

Экранная форма панели InfluxDB предоставляет мощные инструменты для визуализации и анализа данных, собранных системой управления. Графики позволяют отслеживать динамику ключевых параметров, таких как температура в различных зонах и расход газа, что важно для контроля и оптимизации технологического процесса. Возможность интерактивного анализа делает интерфейс удобным и информативным для операторов, технологов и инженеров.

## 4.3 Вывод к разделу 4

Разработка программного обеспечения для автоматизированной системы управления электрической печи отжига металла включает создание архитектуры, способной обеспечить управление технологическим процессом. Основные задачи разработки ПО включают реализацию алгоритмов управления, интеграцию с периферийными устройствами и системами хранения данных, а также создание удобного и интуитивно понятного интерфейса для операторов.

В ходе работы были разработаны ключевые компоненты программного обеспечения, такие как модули для управления нагревом, подачи защитного газа и поддержания давления. Эти модули обеспечивают регулирование параметров отжига, что критически важно для достижения показателей качества и механических свойств металла.

Использование средств разработки, таких как CODESYS для ПЛК и SCADA-система FUXA, позволило создать масштабируемую систему, способную адаптироваться к изменяющимся требованиям производства. Интеграция с базами данных InfluxDB и PostgreSQL обеспечивает хранение и анализ данных, что позволяет оперативно реагировать на изменения в процессе и проводить анализ эффективности работы системы.

Кроме того, была реализована система мониторинга и визуализации данных, которая позволяет операторам в реальном времени отслеживать состояние системы и принимать решения. Это значительно повышает удобство эксплуатации системы и снижает вероятность возникновения аварийных ситуаций.

# 5 Экономическое обоснование разработки программного обеспечения

В данном разделе производится расчёт затрат на создание программного обеспечения, прибыли и выручки от его реализации, расчёт затрат, связанных с покупкой, внедрением и использованием программного обеспечения.

## 5.1 Расчет затрат на создание ПО и его цены

В данном подразделе освещаются вопросы финансово-экономического обоснования программного обеспечения. Содержится экономическое обоснование необходимости разработки программного обеспечения, расчет затрат на создание программного обеспечения, цена и прибыль от его реализации.

Следует провести расчёт затрат на создание программного обеспечения. Для начала выделим состав разработчиков программного продукта (таблица 2).

Таблица 5.1 – Состав разработчиков программного обеспечения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование должности | Численность, чел. | Базовая ставка, руб. | Повышающий коэффициент | Месячный оклад, тыс. руб. |
| Инженер-программист | 1 | 55 000 | 1 | 55 000 |

Время участия в создании ПО каждого специалиста определяется на основе

перечня работ и трудоёмкости их выполнения.

Трудоемкость выполнения работ рассчитывается по формуле

|  |  |
| --- | --- |
| , | (5.1) |

где – минимальное время необходимое для выполнения работы;

– максимальное время необходимое для выполнения работы.

Расчет трудоемкости выполнения работы для каждого разработчика заполнен в таблице 5.2.

Таблица 5.2 – Состав разработчиков программного обеспечения

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование работы | , Ч. | , Ч. | , Ч. |
| Анализ предметной области и формирование требований | 110 | 120 | 114 |
| Проектирование автоматизированной системы | 70 | 85 | 76 |
| Проектирование информационно системы | 70 | 85 | 76 |
| Разработка ПО | 420 | 560 | 476 |
| Тестирование и наладка | 15 | 20 | 17 |
| Итого | 685 | 960 | 759 |

Общие затраты на создание ПО рассчитываются по формуле

|  |  |
| --- | --- |
| , | (5.2) |

где – затраты на разработку ПО;

– затраты на оформление программного продукта и подготовку его к продаже (их можно принять в размере 15 – 25 % от );

– затраты на маркетинговые исследования (их можно принять в размере 10 – 20 % от ).

Фонд оплаты труда для работников , которые участвуют в разработке ПО вычисляется по формуле

|  |  |
| --- | --- |
| , | (5.3) |

где – время участия специалиста определенной квалификации в создании ПО (часы);

– число специалистов определенной квалификации, участвующих в создании ПО;

– месячный оклад работника в соответствии с его категорией или тарифным разрядом;

– длительность смены (8 часов);

– среднее число рабочих дней в месяце (21 день);

– премия, предусмотренная для работников, участвующих в создании программного продукта (в размере 20 – 25 % от );

– выплаты по районному коэффициенту (установлены для г. Кирова в размере 15 % от ()).

Затраты на разработку ПО рассчитываются по формуле

|  |  |
| --- | --- |
| , | (5.4) |

где – затраты на создание математического обеспечения и написания программы;

– затраты, связанные с работой компьютера при разработке ПО.

– прочие затраты, связанные с разработкой ПО (25 – 50 % от ).

рассчитывается по формуле

|  |  |
| --- | --- |
| , | (5.5) |

где – общая ставка страховых взносов (30 % и ФСС НС 0,2 %);

– накладные расходы организации, где разрабатывается ПО (затраты на отопление, освещение, на содержание административно-управленческого персонала и др.), их можно принять в размере 100–200 % от .

Затраты на заработную плату работников, участвующих в создании математического обеспечения и написании программы , определяются по формуле

|  |  |
| --- | --- |
| , | (5.6) |

где – заработная плата программистов, участвующих в создании математического обеспечения и написании программы;

– заработная плата других работников временной творческой группы, принимающих участие в разработке математического обеспечения и написании программы.

Затраты на зарплату программистов рассчитываются по формуле

|  |  |
| --- | --- |
| , | (5.7) |

где – количество операторов (команд) в программе;

– средняя (стандартная) трудоемкость разработки одного оператора (команды) для используемого языка программирования;

– коэффициент новизны разрабатываемой программы;

– трудоемкость других видов работ, выполняемых программистами при создании математического обеспечения и написании программы (разработки общих принципов построения программы, ее структуры, входных и выходных форм и др.).

Затраты на выплату зарплаты других работников временной творческой группы , участвующих в разработке математического обеспечения и написании программы, определяются по формуле

|  |  |
| --- | --- |
| , | (5.8) |

где – число работников определенной квалификации, участвующих в разработке математического обеспечения и написании программы;

– время участия работника определенной квалификации в разработке математического обеспечения и написании программы, в час.

Затраты, связанные с работой компьютера при разработке ПО, можно рассчитать укрупненно по формуле

|  |  |
| --- | --- |
| , | (5.9) |

где – машинное время, необходимое для ввода программы в компьютер, ее трансляцию, редактирование, отладку, тестирование, корректировку и выполнение;

– стоимость одного часа эксплуатации компьютера определенной модели.

Стоимость одного часа эксплуатации компьютера можно определить укрупненно по формуле

|  |  |
| --- | --- |
| , | (5.10) |

где – потребляемая энергия компьютера киловатт-час;

– потребляемая энергия монитора киловатт-час;

– цена одного киловатт-час энергии на 1 июня 2024 года 4,8 рубля.

Расчеты по формулам (5.2) – (5.10) представлены в таблице 5.3.

Таблица 5.3 – Результаты расчетов

|  |  |  |
| --- | --- | --- |
| Наименование статьи затрат | Буквенное обозначение | Результат (руб.) |
| Стоимость одного часа эксплуатации компьютера |  | 0,624 |
| Затраты, связанные с работой компьютера при разработке ПО |  | 361,920 |
| Затраты на заработную плату работников |  | 471592,270 |
| Премии |  | 94318,460 |
| Фонд оплаты труда для работников |  | 663581,100 |
| Затраты на создание математического обеспечения и написания программы |  | 1322791,040 |
| Затраты на разработку ПО |  | 1464630,650 |
| Общие затраты на создание ПО |  | 1977261,370 |

После получения величины общих затратов на создание ПО необходимо определить проектные цены.

Цена создания ПО определяется по формуле

|  |  |
| --- | --- |
| , | (5.11) |

где – размер прибыли.

Размер прибыли рассчитывается по формуле

|  |  |
| --- | --- |
| , | (5.12) |

где – уровень рентабельности ПО (принимается равным в 0,25).

Розничная цена ПО определяется по формуле

|  |  |
| --- | --- |
| . | (5.13) |

Расчеты по формулам (5.11) – (5.13) представлены в таблице 5.4.

Таблица 5.4 – Результаты расчетов

|  |  |  |
| --- | --- | --- |
| Показатель | Буквенное обозначение | Сумма (руб.) |
| Прибыль |  | 494312,84 |
| Цена создания ПО |  | 2471564,21 |
| НДС |  | 484312,84 |
| Розничная цена |  | 2965877,05 |

Розничная цена ПО равна 2 965 877,05 рублей.

## 5.2 Расчет затрат, связанных с внедрением и использованием ПО

Затраты на то, чтобы внедрить ПО будут рассчитываться из стоимости программируемого логического контроллера и его обслуживания, также из затрат на обучение персонала использованием ПО и цены покупки ПО.

На обучение пользованием ПО предполагается отвести три лекции по часу в день. В обучении будут участвовать инженер-программист, инженер-технолог, оператор технологической установки. Заработные платы специалистов, за месяц, представлены в таблице 5.5.

Таблица 5.5 – Заработная плата специалистов

|  |  |
| --- | --- |
| Специалист | Заработная плата (руб.) |
| Инженер-программист | 55 000 |
| Инженер-технолог | 80 000 |
| Оператор технологической установки | 50 000 |

На основании заработной платы специалистов при стандартном рабочем дне в восемь часов и 21 рабочему дню в месяце, затраты на то, чтобы провести лекции инженер-программисту составит 949,32 руб., стоимость обучения инженер-технолога составит 1380,81 руб., стоимость обучения оператора технологической установки составит 863,01 руб. Затраты на обучение пользованием программного обеспечения трех операторов и одного технолога составит 4919,16 руб.

Для внедрения ПО требуется инженер контрольно-измерительных приборов и автоматики. Зарплата такого специалиста составляет 80 000 руб. в месяц. Его задача установить ПЛК и модули ввода-вывода. На установку требуется один рабочий день, что составит 3682,19 руб. Стоимость ПЛК Fastwel CPM723-01 со всеми необходимыми модулями составляет 317 000 рублей.

Прирост в доходности одной печи будет рассчитываться в абсолютных значениях, производственный цикл до попадания полуфабриката в печь учитываться не будет.

Полуфабрикат отжигается с помощью четырех нагревательных тэнов. Время выдержки заготовки один час на каждые два с половиной сантиметра толщины. Цена затрат на электроэнергию работы одной печи отжига определяется по формуле

|  |  |
| --- | --- |
| , | (5.14) |

где – потребляемая электроэнергия одного ТЭНа, кВт;

– время работы процесса, часов;

– количество ТЭНов в печи;

– стоимость одного киловатт-час.

Таким образом, цена затрат на электроэнергию работы одной печи отжига при , , , = 8 рублей, будет равна 107 520 рублей.

До модернизированной системы управления проводился только черный отжиг металла. В этом виде отжига, металл окисляется, что приводит к шлакованию печи. Это приводит к необходимости очистки металла и печи после каждого процесса отжига, что ведет к простоям между загрузками печи. Время простоя составляет одну рабочую смену.

Если учесть, что с одного процесса отжига получается 16 профилей толщиной 30 см., а стоимость одного такого профиля составляет 80 000 рублей, то выручка составит 1 280 000 рублей. С учетом простоев за месяц получается 4 391 000 рублей.

Модернизированная система управления позволяет использовать белый отжиг металла, подавая и регулируя защитную атмосферу, что значительно уменьшает окисление металла. Таким образом, можно избавиться от простоев, так как не нужно очищать металл и печь. С учетом этого выручка за месяц составит 5 120 000 рублей.

Экономия с использованием модернизированной системы управления составляет примерно 14,3 % в месяц.

Итоговые затраты на создание и внедрение ПО равны 3 291 478,41 рублей

Окупаемость рассчитывается как отношение итоговых затрат к годовой экономии и равна 0,06 года.

Расчет экономической эффективности с использованием автоматизированной системы управления электрической печью отжига металла является очень сложным, а иногда не отражает истинных данных, так как зависит от большого количества факторов, например, таких как полного производственного цикла металлической заготовки.

## 5.3 Вывод к разделу 5

В данном разделе произведен анализ экономической целесообразности разработки программного обеспечения для автоматизированной системы управления электрической печи отжига металла. Расчет затрат на создание программного обеспечения включал определение состава разработчиков, оценку трудоемкости выполнения работ и расчет связанных с этим финансовых затрат. Рассмотрены аспекты экономического обоснования, включая затраты на покупку, внедрение и использование программного обеспечения. Произведены расчеты, показывающие, что модернизация оправдана с экономической точки зрения, так как она обеспечит выгоду и прибыль от реализации, выгода составляет 14,3 % в месяц.

# Заключение

В настоящем проекте была разработана автоматизированная система управления электрической печи отжига металла. Основная цель проекта заключалась в повышении эффективности и качества технологического процесса отжига, что достигается путем автоматизации управления параметрами печи и интеграции системы мониторинга и визуализации.

На начальных этапах проекта был проведен анализ существующей системы, выявлены её недостатки и определены требования к новой системе.

Проектирование информационной системы включало разработку структурной схемы, которая отразила взаимодействие компонентов системы, включая датчики, исполнительные механизмы и интерфейсы связи. Также разработаны функциональные модели, с помощью которых были определены ключевые процессы, такие как сбор и обработка данных, управление параметрами печи и взаимодействие с оператором.

Разработанная функциональная схема автоматизации и схема информационных потоков позволили представить процессы, происходящие в системе. В результате был выбран оптимальный программируемый логический контроллер, обеспечивающий. Были разработаны алгоритмы управления процессом отжига.

Ключевым этапом проекта стала разработка программного обеспечения, которое включает алгоритмы управления процессом отжига, модули для работы с периферийными устройствами и системы хранения данных. Использование современных средств разработки, таких как CODESYS и SCADA-система FUXA, позволило создать гибкую и масштабируемую систему, легко адаптируемую к изменяющимся условиям производства.

Система мониторинга и визуализации данных, интегрированная с базами данных InfluxDB и PostgreSQL, обеспечивает надежное хранение и анализ данных о процессе отжига. Это позволяет операторам в реальном времени отслеживать состояние системы и оперативно принимать решения, что значительно повышает удобство эксплуатации и снижает вероятность аварийных ситуаций.

Важной частью проекта стало экономическое обоснование разработки программного обеспечения. Были проведены расчеты затрат на разработку, внедрение и эксплуатацию системы, а также оценены ожидаемые экономические эффекты.

Таким образом, проект по разработке и внедрению автоматизированной системы управления электрической печи отжига металла успешно достиг поставленных целей. Новая система продемонстрировала высокую эффективность и надежность в эксплуатации, что позволяет рекомендовать её для широкого применения в металлургической промышленности.

# Приложение А (обязательное) Программа main.cfc

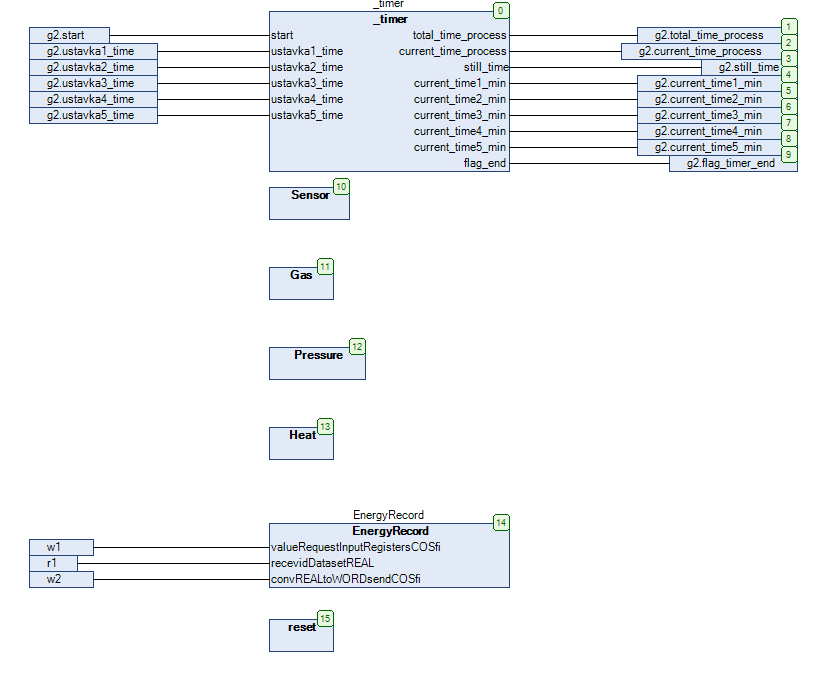
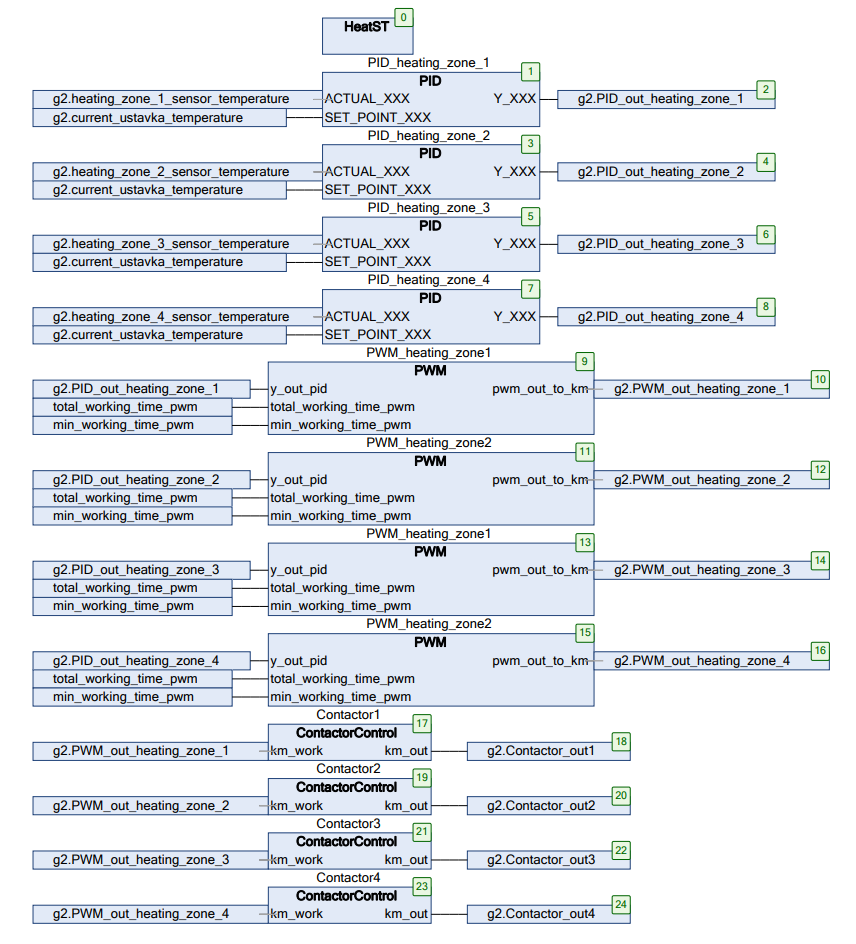


Рисунок А.1 – Программный код main.cfc

# Приложение Б (обязательное) Программа модуля нагрева

Рисунок Б.1 – Программный код файла Heat.cfc

Листинг Б.1 – Содержимое файла HeatST.st

PROGRAM HeatST

VAR

HeatingControl\_stage1: HeatingControl;

HeatingControl\_stage2: HeatingControl;

HeatingControl\_stage3: HeatingControl;

HeatingControl\_stage4: HeatingControl;

HeatingControl\_stage5: HeatingControl;

HeatingControl\_stage1\_start: BOOL;

HeatingControl\_stage2\_start: BOOL;

HeatingControl\_stage3\_start: BOOL;

HeatingControl\_stage4\_start: BOOL;

HeatingControl\_stage5\_start: BOOL;

stage1\_up: BOOL;

stage2\_up: BOOL;

stage3\_up: BOOL;

stage4\_up: BOOL;

stage5\_up: BOOL;

END\_VAR

HeatingControl\_stage1\_start := g2.start;

HeatingControl\_stage1(

start:= HeatingControl\_stage1\_start,

stop:= g2.stop,

ustavka1\_temperature:= 0,

ustavka2\_temperature:= g2.ustavka1\_temperature,

ustavka\_time:= g2.ustavka1\_time,

back\_stage\_temperature:= 0,

actual\_temperature=> g2.ustavka1\_actual\_temperature,

time\_stage\_min=> ,

time\_stage\_sec=> ,

stage\_up=> HeatingControl\_stage2\_start,

stage=> g2.stage1\_heat,

current\_ustavka\_temperature=> g2.current\_ustavka\_temperature);

HeatingControl\_stage2(

start:= HeatingControl\_stage2\_start,

stop:= g2.stop,

ustavka1\_temperature:= g2.ustavka1\_temperature,

ustavka2\_temperature:= g2.ustavka2\_temperature,

ustavka\_time:= g2.ustavka2\_time,

back\_stage\_temperature:= g2.ustavka1\_temperature,

actual\_temperature=> g2.ustavka2\_actual\_temperature,

time\_stage\_min=> ,

time\_stage\_sec=> ,

stage\_up=> HeatingControl\_stage3\_start,

stage=> g2.stage2\_heat,

current\_ustavka\_temperature=> g2.current\_ustavka\_temperature);

HeatingControl\_stage3(

start:= HeatingControl\_stage3\_start,

stop:= g2.stop,

ustavka1\_temperature:= g2.ustavka2\_temperature,

ustavka2\_temperature:= g2.ustavka3\_temperature,

ustavka\_time:= g2.ustavka3\_time,

back\_stage\_temperature:= g2.ustavka2\_temperature,

actual\_temperature=> g2.ustavka3\_actual\_temperature,

time\_stage\_min=> ,

time\_stage\_sec=> ,

stage\_up=> HeatingControl\_stage4\_start,

stage=> g2.stage3\_heat,

current\_ustavka\_temperature=> g2.current\_ustavka\_temperature);

HeatingControl\_stage4(

start:= HeatingControl\_stage4\_start,

stop:= g2.stop,

ustavka1\_temperature:= g2.ustavka3\_temperature,

ustavka2\_temperature:= g2.ustavka4\_temperature,

ustavka\_time:= g2.ustavka4\_time,

back\_stage\_temperature:= g2.ustavka3\_temperature,

actual\_temperature=> g2.ustavka4\_actual\_temperature,

time\_stage\_min=> ,

time\_stage\_sec=> ,

stage\_up=> HeatingControl\_stage5\_start,

stage=> g2.stage4\_heat,

current\_ustavka\_temperature=> g2.current\_ustavka\_temperature);

HeatingControl\_stage5(

start:= HeatingControl\_stage5\_start,

stop:= g2.stop,

ustavka1\_temperature:= g2.ustavka4\_temperature,

ustavka2\_temperature:= g2.ustavka5\_temperature,

ustavka\_time:= g2.ustavka5\_time,

back\_stage\_temperature:= g2.ustavka4\_temperature,

actual\_temperature=> g2.ustavka5\_actual\_temperature,

time\_stage\_min=> ,

time\_stage\_sec=> ,

stage\_up=> g2.heat\_over,

stage=> g2.stage5\_heat,

current\_ustavka\_temperature=> g2.current\_ustavka\_temperature);

IF HeatingControl\_stage5.stage\_up = TRUE THEN

HeatingControl\_stage1\_start := FALSE;

HeatingControl\_stage2\_start := FALSE;

HeatingControl\_stage3\_start := FALSE;

HeatingControl\_stage4\_start := FALSE;

HeatingControl\_stage5\_start := FALSE;

END\_IF;

IF HeatingControl\_stage1\_start = TRUE THEN g2.current\_ustavka\_temperature := HeatingControl\_stage1.current\_ustavka\_temperature; END\_IF;

IF HeatingControl\_stage2\_start = TRUE THEN g2.current\_ustavka\_temperature := HeatingControl\_stage2.current\_ustavka\_temperature; END\_IF;

IF HeatingControl\_stage3\_start = TRUE THEN g2.current\_ustavka\_temperature := HeatingControl\_stage3.current\_ustavka\_temperature; END\_IF;

IF HeatingControl\_stage4\_start = TRUE THEN g2.current\_ustavka\_temperature := HeatingControl\_stage4.current\_ustavka\_temperature; END\_IF;

IF HeatingControl\_stage5\_start = TRUE THEN g2.current\_ustavka\_temperature := HeatingControl\_stage5.current\_ustavka\_temperature; END\_IF;

Листинг Б.2 – Содержание файла HeatingControl.st

FUNCTION\_BLOCK HeatingControl

VAR\_INPUT

start: BOOL;

stop: BOOL;

ustavka1\_temperature: REAL;

ustavka2\_temperature: REAL;

ustavka\_time: REAL;

back\_stage\_temperature: REAL;

END\_VAR

VAR\_OUTPUT

actual\_temperature: REAL;

time\_stage\_min: REAL;

time\_stage\_sec: REAL;

stage\_up: BOOL;

stage: BOOL;

current\_ustavka\_temperature: REAL;

END\_VAR

VAR

delta\_temperature: REAL;

debugging\_time\_divider: REAL := 1;

blink: Util.BLINK;

ustavka\_time\_temp: REAL;

flag\_set\_temperature: BOOL := TRUE;

time\_elapsed: REAL;

tp: Standard.TP;

blink\_timer\_low : TON; // Таймер для низкого состояния

blink\_timer\_high : TON; // Таймер для высокого состояния

blink\_state : BOOL := FALSE; // Состояние мигания

time\_low : TIME := T#1S; // Время низкого состояния

time\_high : TIME := T#1MS; // Время высокого состояния

END\_VAR

IF NOT start THEN

stage\_up := FALSE;

stage := FALSE;

current\_ustavka\_temperature := 0.0;

delta\_temperature := 0.0;

ustavka\_time\_temp := 0.0;

actual\_temperature := 0.0;

time\_stage\_min := 0.0;

time\_stage\_sec := 0.0;

flag\_set\_temperature := TRUE;

blink.ENABLE := FALSE;

blink\_timer\_low.IN := FALSE;

blink\_timer\_high.IN := FALSE;

blink\_state := FALSE;

ELSIF start = TRUE THEN

IF flag\_set\_temperature = TRUE THEN actual\_temperature := back\_stage\_temperature; flag\_set\_temperature := FALSE; END\_IF;

IF ustavka\_time\_temp <> (ustavka\_time \* debugging\_time\_divider) THEN

stage := TRUE;

blink(ENABLE := TRUE, TIMELOW := T#1000MS, TIMEHIGH := T#1MS);

delta\_temperature := (ustavka2\_temperature - ustavka1\_temperature) / (ustavka\_time \* debugging\_time\_divider);

IF blink.OUT = TRUE THEN

actual\_temperature := actual\_temperature + delta\_temperature;

time\_stage\_sec := time\_stage\_sec + 1;

time\_stage\_min := time\_stage\_min - 1;

ustavka\_time\_temp := ustavka\_time\_temp + 1;

END\_IF;

current\_ustavka\_temperature := actual\_temperature;

ELSE

stage\_up := TRUE;

stage := FALSE;

blink.ENABLE := FALSE;

start := FALSE;

current\_ustavka\_temperature := 0.0;

delta\_temperature := 0.0;

//ustavka\_time\_temp := 0.0;

actual\_temperature := 0.0;

time\_stage\_min := 0.0;

time\_stage\_sec := 0.0;

flag\_set\_temperature := TRUE;

blink.ENABLE := FALSE;

END\_IF;

END\_IF;

# Приложение В (обязательное) Программа PWM.st

FUNCTION\_BLOCK PWM

VAR\_INPUT

y\_out\_pid : REAL;

total\_working\_time\_pwm : REAL;

min\_working\_time\_pwm : REAL;

END\_VAR

VAR\_OUTPUT

pwm\_out\_to\_km : BOOL;

END\_VAR

VAR

current\_temperature\_percent : REAL;

working\_time\_pwm : REAL;

time\_low\_working\_pwm : TIME;

time\_high\_working\_pwm : TIME;

blink\_pwm : Util.BLINK();

blink\_delay\_pwm : Util.BLINK();

END\_VAR

IF g2.start = TRUE THEN

current\_temperature\_percent := y\_out\_pid;

IF total\_working\_time\_pwm <= min\_working\_time\_pwm THEN

total\_working\_time\_pwm := 10000;

END\_IF

working\_time\_pwm := (current\_temperature\_percent / 100) \* total\_working\_time\_pwm;

blink\_delay\_pwm(ENABLE := TRUE, TIMELOW := T#1000MS, TIMEHIGH := T#1MS);

IF blink\_delay\_pwm.OUT = TRUE THEN

time\_low\_working\_pwm := REAL\_TO\_TIME(total\_working\_time\_pwm - working\_time\_pwm);

time\_high\_working\_pwm := REAL\_TO\_TIME(working\_time\_pwm);

END\_IF

blink\_pwm(ENABLE := TRUE, TIMELOW := time\_low\_working\_pwm, TIMEHIGH := time\_high\_working\_pwm, OUT => pwm\_out\_to\_km);

ELSE

blink\_pwm.ENABLE := FALSE;

blink\_delay\_pwm.ENABLE := FALSE;

pwm\_out\_to\_km := FALSE;

current\_temperature\_percent := 0;

END\_IF;

# Приложение Г (обязательное) Программа посредник между ПЛК и БД

Листинг Г.1 – содержимое файла main.py

from opcua\_client.opcua\_client\_driver import client, node\_path

from opcua\_client.get\_recipe\_name import get\_recipe\_name

from opcua\_client.send\_recipe import send\_recipe

from time import sleep

from repository.get\_recipe\_object import get\_recipe\_object

from repository.select\_recipe import select\_recipe

if \_\_name\_\_ == "\_\_main\_\_":

while True:

try:

client.connect()

recipe\_name = get\_recipe\_name(client, node\_path)

if recipe\_name != 0 and recipe\_name!= None:

recipe = select\_recipe(recipe\_name)

obj = get\_recipe\_object(recipe)

send\_recipe(obj)

finally:

client.disconnect()

sleep(1)

Листинг Г.2 – содержимое файла db\_driver.py

from sqlalchemy import create\_engine

from sqlalchemy.ext.declarative import declarative\_base

from sqlalchemy.orm import sessionmaker

import configparser

from sqlalchemy.exc import OperationalError

config = configparser.ConfigParser()

config.read('C:\\University\\DIPLOM\\metal-annealing-furnace\\opcua-client\\controller\\db\_config.ini')

config.sections()

DATABASE\_USER = config.get('database', 'user')

DATABASE\_PASSWORD = config.get('database', 'password')

DATABASE\_HOST = config.get('database', 'host')

DATABASE\_PORT = config.get('database', 'port')

DATABASE\_NAME = config.get('database', 'dbname')

DATABASE\_URL = f'postgresql://{DATABASE\_USER}:{DATABASE\_PASSWORD}@{DATABASE\_HOST}:{DATABASE\_PORT}/{DATABASE\_NAME}'

try:

engine = create\_engine(DATABASE\_URL)

Base = declarative\_base()

Session = sessionmaker(bind=engine)

session = Session()

print("Подключение к базе данных успешно!")

except OperationalError as e:

print(f"Ошибка подключения к базе данных: {e}")

Листинг Г.3 – содержимое файла recipe\_model.py

from sqlalchemy import Column, Integer, String, ForeignKey, Float

from sqlalchemy.orm import relationship

from controller.db\_driver import Base

class Stage(Base):

\_\_tablename\_\_ = 'stage'

id = Column(Integer, primary\_key=True)

time\_ustavka = Column(Float, nullable=False)

temperature\_ustavka = Column(Float, nullable=False)

gas = Column(Float, nullable=False)

pressure = Column(Float, nullable=False)

class Recipe(Base):

\_\_tablename\_\_ = 'recipe'

id = Column(Integer, primary\_key=True)

name\_recipe = Column(String(30), nullable=False)

stage\_1\_id = Column(Integer, ForeignKey('stage.id'), nullable=False)

stage\_2\_id = Column(Integer, ForeignKey('stage.id'), nullable=False)

stage\_3\_id = Column(Integer, ForeignKey('stage.id'), nullable=False)

stage\_4\_id = Column(Integer, ForeignKey('stage.id'), nullable=False)

stage\_5\_id = Column(Integer, ForeignKey('stage.id'), nullable=False)

stage\_1 = relationship("Stage", foreign\_keys=[stage\_1\_id])

stage\_2 = relationship("Stage", foreign\_keys=[stage\_2\_id])

stage\_3 = relationship("Stage", foreign\_keys=[stage\_3\_id])

stage\_4 = relationship("Stage", foreign\_keys=[stage\_4\_id])

stage\_5 = relationship("Stage", foreign\_keys=[stage\_5\_id])

Листинг Г.4 – содержимое файла opc\_client\_driver.py

from opcua import Client, ua

url = "opc.tcp://127.0.0.1:4840" # end point

node\_path = "ns=4;s=|var|CODESYS Control Win V3 x64.Application.g2"

client = Client(url)

Листинг Г.5 – содержимое файла select\_recipe.py

from controller.db\_driver import session

from models.recipe\_model import Recipe, Stage

def select\_recipe(recipe\_id):

return session.query(Recipe).filter(Recipe.id == recipe\_id).all()

Листинг Г.6 – содержимое файла get\_recipe\_name.py

def get\_recipe\_name(client, node\_path):

g2\_obj = client.get\_node(node\_path)

children = g2\_obj.get\_children()

for child in children:

child\_name = child.get\_browse\_name().Name

if child\_name == "recipe":

return child.get\_value()

Листинг Г.7 – содержимое файла get\_recipe\_object.py

def get\_recipe\_object(select\_recipe):

recipe\_info = []

for recipe in select\_recipe:

recipe\_data = {

"id": recipe.id,

"name": recipe.name\_recipe,

"stages": []

}

stages = [

recipe.stage\_1,

recipe.stage\_2,

recipe.stage\_3,

recipe.stage\_4,

recipe.stage\_5,

]

for i, stage in enumerate(stages, start=1):

stage\_info = {

"stage": i,

"time": stage.time\_ustavka,

"temp": stage.temperature\_ustavka,

"gas": stage.gas,

"pressure": stage.pressure

}

recipe\_data["stages"].append(stage\_info)

recipe\_info.append(recipe\_data)

return recipe\_info

Листинг Г.8 – содержимое файла send\_recipe.py

from opcua\_client.opcua\_client\_driver import client, node\_path, ua

def send\_recipe(recipe\_info):

for i in range(1,6):

node\_ustavka\_time = client.get\_node(node\_path + f".ustavka{i}\_time")

node\_ustavka\_temperature = client.get\_node(node\_path + f".ustavka{i}\_temperature")

node\_ustavka\_gas = client.get\_node(node\_path + f".ustavka{i}\_gas")

node\_ustavka\_pressure = client.get\_node(node\_path + f".ustavka{i}\_pressure")

stage\_time = recipe\_info[0]["stages"][i - 1]["time"]

variant\_time = ua.Variant(stage\_time, ua.VariantType.Float)

node\_ustavka\_time.set\_value(variant\_time)

stage\_temperature = recipe\_info[0]["stages"][i - 1]["temp"]

variant\_temperature = ua.Variant(stage\_temperature, ua.VariantType.Float)

node\_ustavka\_temperature.set\_value(variant\_temperature)

stage\_gas = recipe\_info[0]["stages"][i - 1]["gas"]

variant\_gas = ua.Variant(stage\_gas, ua.VariantType.Float)

node\_ustavka\_gas.set\_value(variant\_gas)

stage\_pressure = recipe\_info[0]["stages"][i - 1]["pressure"]

variant\_pressure = ua.Variant(stage\_pressure, ua.VariantType.Float)

node\_ustavka\_pressure.set\_value(variant\_pressure)

# Приложение Д (обязательное) Схема алгоритма нагрева

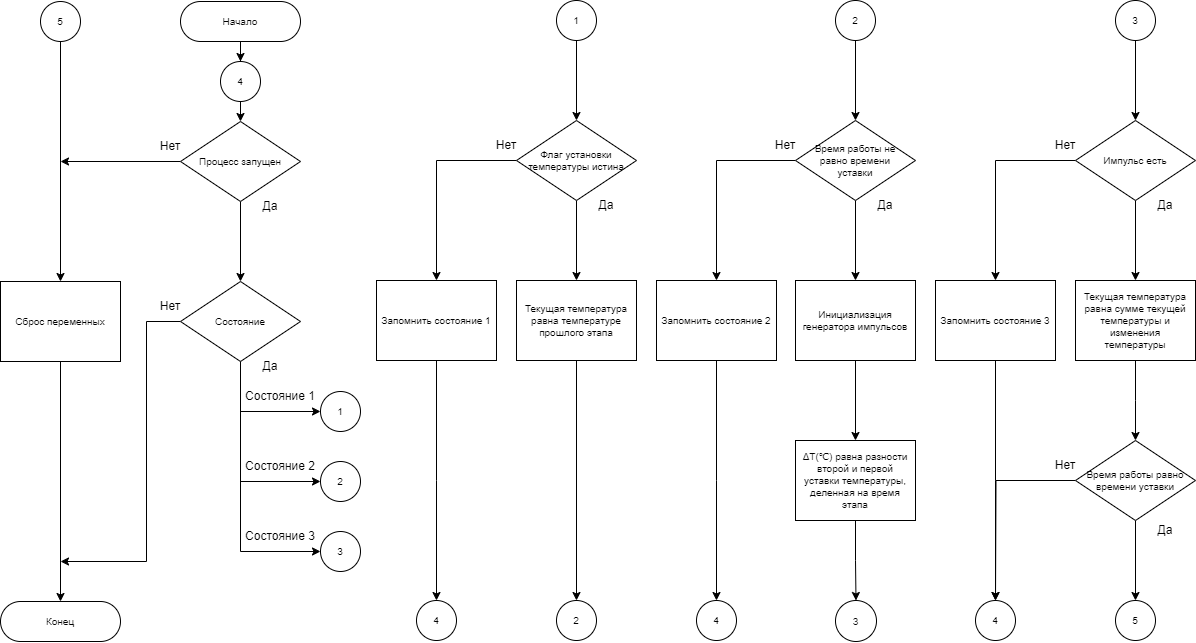


Рисунок Д.1 – Схема алгоритма нагрева

# Приложение Ж (обязательное) Функциональная схема автоматизации электрической печи отжига металла

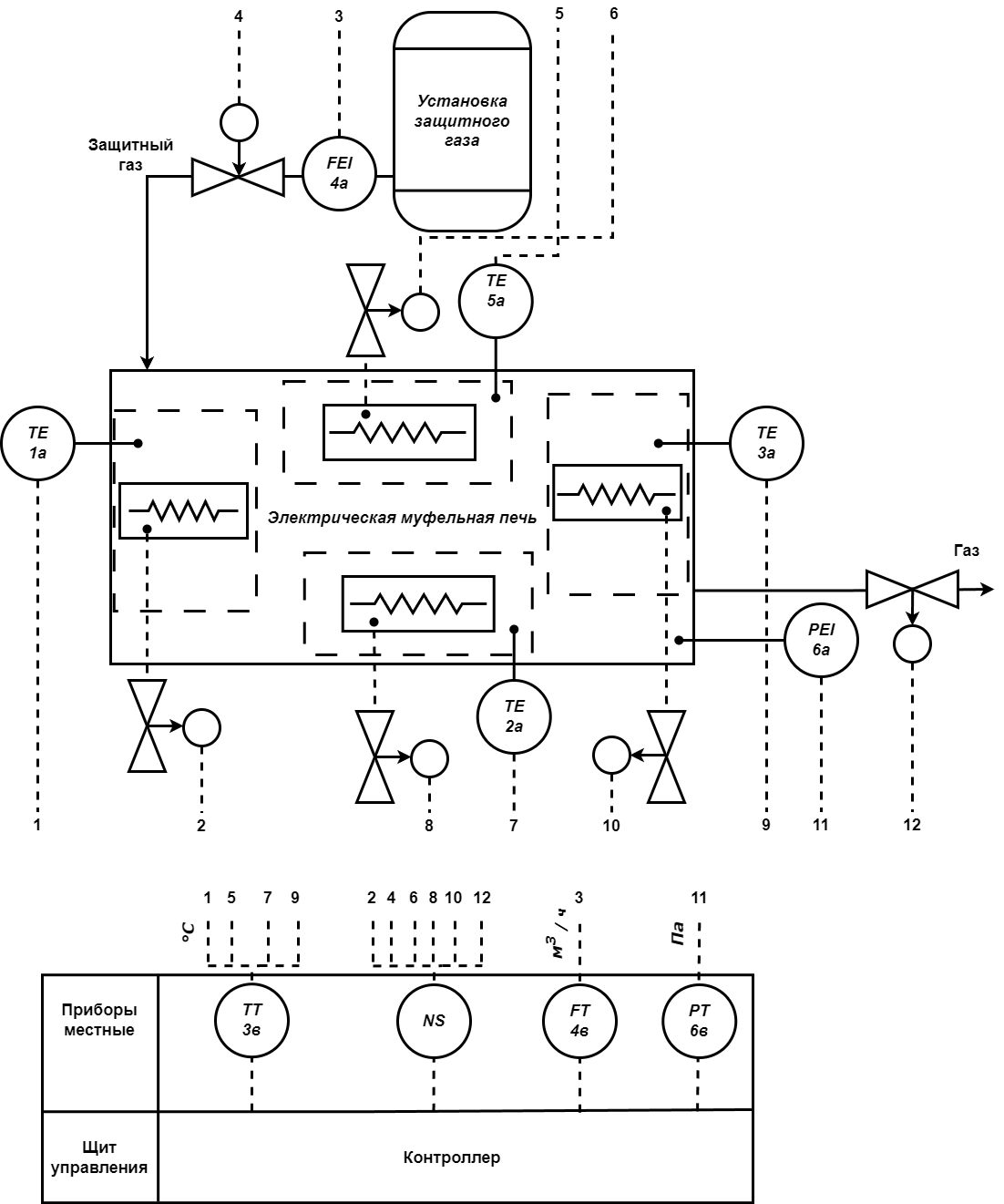


Рисунок Ж.1 *–* Функциональная схема автоматизации электрической печи отжига металла

# Приложение К (обязательное) Экранная форма панели трендов температуры

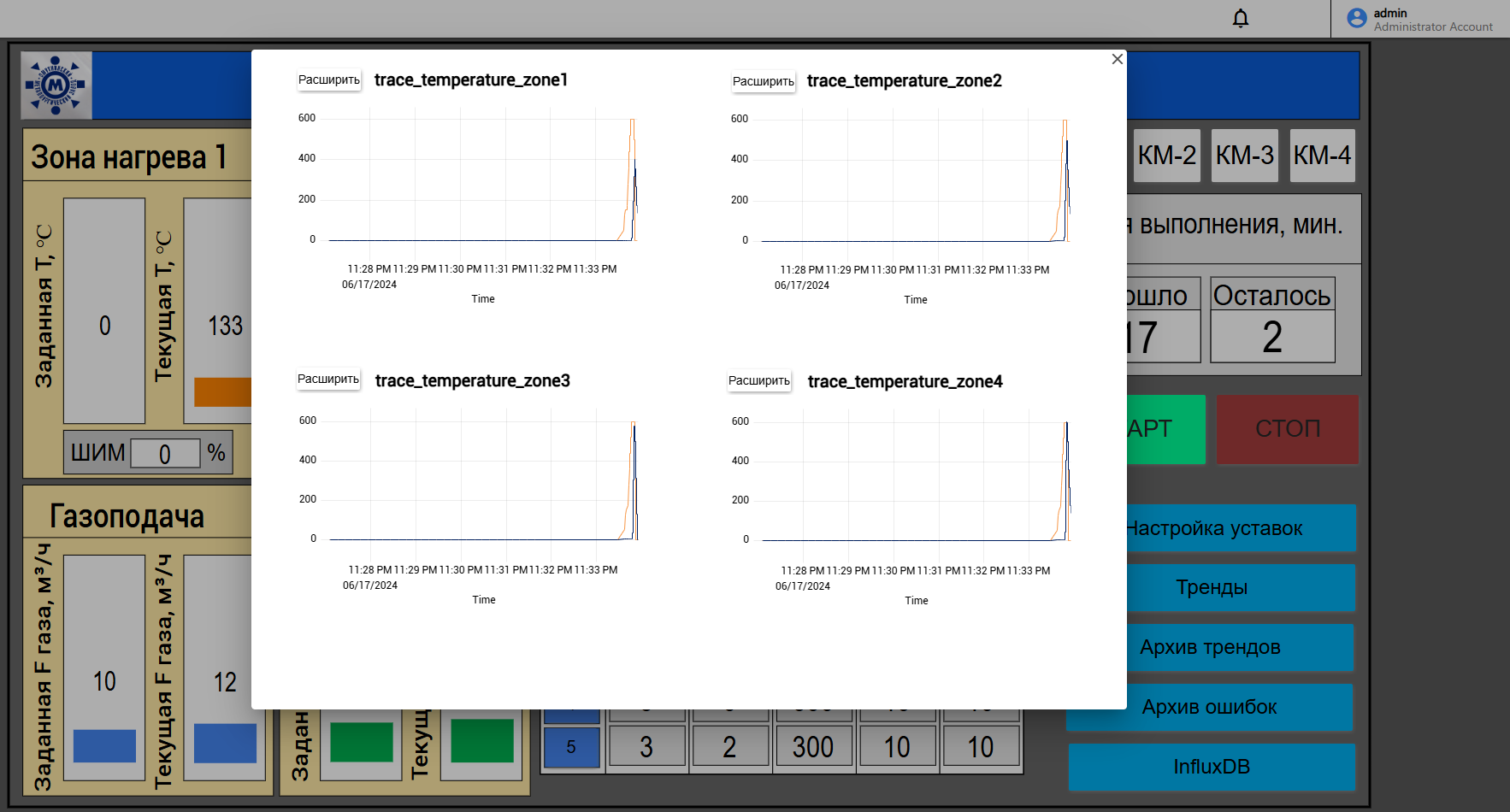


Рисунок К.1 *–* Экранная форма панели трендов

# Приложение Л (обязательное) Экранная форма панели InfluxDB

****

Рисунок Л.1 *–* Экранная форма InfluxDB

# Приложение М (обязательное) Авторская справка

Я, Уткин Никита Андреевич, автор выпускной квалификационной работы Модернизация системы управления электрической печью отжига металла, сообщаю, что мне известно о персональной ответственности автора за разглашение сведений, подлежащих защите законами Российской Федерации о защите объектов интеллектуальной собственности.

Одновременно сообщаю, что:

1. При подготовке к защите (опубликованию) выпускной квалификационной работы не использованы источники (документы, отчеты, диссертации, литература и т.п.), имеющие гриф секретности или «Для служебного пользования» ВятГУ или другой организации.
2. Данная работа не связана (связана) с незавершенными исследованиями или уже с завершенными, но еще официально не разрешенными к опубликованию ВятГУ или другими организациями.
3. Данная работа не содержит (содержит) коммерческую информацию, способную нанести ущерб интеллектуальной собственности ВятГУ или другой организации.
4. Данная работа является (не является) результатом НИР или ОКР, выполняемой по договору с организацией (указать согласие заказчика)                                                                .
5. В предлагаемом к опубликованию тексте нет данных по незащищенным объектам интеллектуальной собственности других авторов.
6. Согласен на использование результатов своей работы безвозмездно в ВятГУ для учебного процесса, а также на размещение своей работы в электронной информационно- образовательной среде ВятГУ.
7. Использование моей выпускной квалификационной работы в научных исследованиях оформляется в соответствии с законодательством Российской Федерации о защите интеллектуальной собственности.

Автор Уткин Н.А.

*(подпись) (Фамилия И.О.)*

« » 2024 г.

Сведения по авторской справке подтверждаю:

И.о. зав. каф. Семеновых В.И.

*(подпись) (Фамилия И.О.)*

« » 2024 г.

# Приложение Н (справочное) Список обозначений и сокращений

В настоящей выпускной квалификационной работе используются следующие обозначения и сокращения:

АСУТП – автоматизированная система управления технологическим процессом;

БД – база данных;

ИС – информационная система;

ПИД – пропорционально-интегрально-дифференцирующий;

ПЛК – программируемый логический контроллер;

ПО – программное обеспечение;

СУБД – система управления базами данных;

ФСА – функциональная схема автоматизации;

ШИМ – широтно-импульсная модуляция;

CFC – continuous flow chart;

DFD – data flow diagram;

FBD – function block diagram;

HMI – human-machine interface;

IDEF – integrated definition;

IDEF3 – integrated definition for process description capture method;

LD – ladder diagram;

OPC UA – open platform communications unified architecture;

SCADA – supervisory control and data acquisition;

SQL – structured query language;

ST – structured text;

TSDB – time series database.

# Приложение Р (справочное) Библиографический список

1. Автоматизация управления электрическими печами [Электронный ресурс] // Owen, 2024. URL: https://owen.ru/project/avtomatizacziya-upravleniya-elektricheskimi-pechami (дата обращения: 29.04.2024).
2. ФЩЛ-501, ФЩЛ-502 [Электронный ресурс] // ТеплоПрибор.рф, 2024. URL: https:// теплогазприбор.рф/catalog/187 (дата обращения: 29.04.2024).
3. ОТЖИГ СТАЛИ: ПРОЦЕСС, ВИДЫ, ПРЕИМУЩЕСТВА, НЕДОСТАТКИ И ОБЛАСТИ ПРИМЕНЕНИЯ [Электронный ресурс] // Stankoff.ru, 2024. URL: https://www.stankoff.ru/blog/post/772# (дата обращения: 25.04.2024).
4. Базовые знания по IDEF0 [Электронный ресурс] // Comindware, 2024. URL: https://www.comindware.ru/blog/ (дата обращения: 15.05.2024).
5. Использование DFD: как описать движение данных в бизнес-процессах [Электронный ресурс] // System Education, 2024. URL: https://systems.education/data-flow-diagrams (дата обращения: 15.05.2024).
6. Просто о стандартах OPC DA и OPC UA [Электронный ресурс] // IPC2U, 2024. URL: https://ipc2u.ru/articles/prostye-resheniya/prosto-o-standartakh-opc-da-i-opc-ua/ (дата обращения: 16.05.2024).
7. Знакомство с InfluxDB и базами данных временных рядов [Электронный ресурс] // Tproger, 2024. URL: https://tproger.ru/translations/influxdb-guide (дата обращения: 15.05.2024).
8. CODESYS [Электронный ресурс] // CODESYS Online Help, 2024. URL: https://help.codesys.com/ (дата обращения: 14.05.2024).
9. CODESYS V3.5 для enterprise-разработчика [Электронный ресурс] // Автоматизация производства, 2024. URL: https://aip.com.ru/article/codesys-v35-dlya-enterprise-razrabotchika (дата обращения: 14.05.2024).
10. Python OPC-UA Documentation [Электронный ресурс] // Python OPC-UA, 2024. URL: https://help.codesys.com/ (дата обращения: 17.05.2024).
11. FUXA Wiki [Электронный ресурс] // GitHub, 2024. URL: https://github.com/frangoteam/FUXA/wiki (дата обращения: 08.05.2024).
12. Функциональная схема автоматизации [Электронный ресурс] // Электронный фонд правовых и нормативно- технических документов, 2024. URL: https://docs.cntd.ru/document/1200108003 (дата обращения: 11.05.2024).