

## АННОТАЦИЯ

Целью данного дипломного проекта является модернизация автоматизированной системы диспетчерского управления.

При анализе существующей системы диспетчерского управления были выявлены недостатки комплекса, обозначены пути и глубина модернизации, был произведен выбор технических решений, разработан обратный канал связи для реализации удаленного управления параметрами подстанции, разработан шкаф управления.

Была спроектирована структура программных средств оперативно-информационной подсистемы комплекса и системы управления подстанцией. Составлены блок-схемы алгоритмов работы программы диспетчерского управления и программы управления технологическим оборудованием. Произведено моделирование работы энергосистемы, рассмотрены алгоритмы оптимизации режима и планирования потребления. Реализована программа диспетчерского управления подстанции в виде интерактивного приложения с графическим интерфейсом пользователя.

Произведен функционально-стоимостной и экономический анализ проекта и анализ с точки зрения безопасности и влияния на окружающую среду.

Объем расчетно-пояснительной записки составляет 160 страниц, графической части – 11 листов формата А1, наименований используемых источников – 31.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ . . . . .	5
1 Обоснование модернизации автоматизированной системы диспетчерского управления . . . . .	8
1.1 Характеристика оборудования подстанции . . . . .	8
1.2 Анализ задач диспетчерского управления . . . . .	23
1.3 Анализ тенденций развития систем диспетчерского управления	25
1.4 Функционально-стоимостной анализ базового варианта . . . . .	27
1.5 Техническое задание на проект . . . . .	35
2 Проектирование, конструирование и моделирование технических средств . . . . .	38
2.1 Системный анализ проектируемой системы на основе методов декомпозиции . . . . .	39
2.2 Разработка структурной схемы системы диспетчерского управления . . . . .	47
2.3 Разработка структурной схемы системы телемеханизации подстанции . . . . .	51
2.4 Выбор основных технических средств . . . . .	53
2.4.1 Требования к техническим средствам . . . . .	53
2.4.2 Электрошкаф . . . . .	54
2.4.3 Программируемый логический контроллер Овен . . . . .	55
2.4.4 Блок питания контроллера Овен . . . . .	57
2.4.5 Вспомогательный контроллер Raspberry Pi . . . . .	58
2.4.6. Модуль телеуправления ТУ430 для «КП Исеть» . . . . .	59
2.4.7 Регулятор микропроцессорный РНМ-1 . . . . .	61
2.5 Разработка шкафа управления . . . . .	65
2.6. Разработка электрической схемы подключения программируемого контроллера . . . . .	66
2.7 Моделирование потребления энергии с помощью программы «Консоль ПП» . . . . .	67

2.8. Разработка схемы электрической принципиальной шкафа управления . . . . .	70
3 Информационное и программное обеспечение системы управления . . . . .	74
3.1 Информационная структура, ведомости входных и выходных сигналов . . . . .	74
3.1.1 Информационная структура . . . . .	74
3.1.2 Ведомости входных и выходных сигналов . . . . .	76
3.2 Разработка алгоритмов диспетчерского управления . . . . .	78
3.2.1 Разработка алгоритма программы диспетчерского управления . . . . .	78
3.2.2 Разработка алгоритма программируемого логического контроллера . . . . .	80
3.3 Разработка экранных форм диспетчера . . . . .	81
3.3.1 Разработка мнемосхемы подстанции «Левашово» для системы Тораз . . . . .	81
3.3.2 Разработка графического интерфейса программы диспетчерского управления подстанции . . . . .	82
3.4 Разработка программы диспетчерского управления . . . . .	83
3.4.1 Программный интерфейс расширения функциональности ОИК СК-2007 . . . . .	83
3.4.2 Разработка программы контроля параметров подстанции ДиспетчерПро . . . . .	86
4 Эксплуатационная документация . . . . .	92
4.1 Инструкция по эксплуатации . . . . .	92
4.1.1 Инструкция по установке программы контроля параметров ДиспетчерПро . . . . .	92
4.1.2 Инструкция по подключению микропроцессорного регулятора РНМ-1 . . . . .	94
4.1.3 Инструкция по программированию и настройке контроллера	

Овен ПЛК . . . . .	95
4.1.4 Указания по монтажу . . . . .	98
5 Функционально-стоимостной и экономический анализ проекта	100
5.1 Функционально–стоимостной анализ проектируемой информационной системы . . . . .	100
5.2 Экономическая оценка проекта . . . . .	102
6 Безопасность и экологичность проекта . . . . .	106
6.1 Анализ безопасности труда диспетчера . . . . .	106
6.1.1 Специальная оценка условий труда . . . . .	110
6.2 Экологическая безопасность и охрана окружающей среды . . . . .	112
6.2.1 Влияние подстанции «Левашово» на окружающую среду . . . . .	112
6.2.2 Влияние электромагнитного поля на население и животный мир . . . . .	116
6.3 Предупреждение техногенных аварий и защита в ЧС . . . . .	118
6.3.1 Анализ возможных ЧС в районе расположения подстанции «Левашово» . . . . .	118
6.3.2 Расчёт молниезащиты подстанции . . . . .	121
ЗАКЛЮЧЕНИЕ . . . . .	126
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ . . . . .	127
ПРИЛОЖЕНИЯ . . . . .	130

## ВВЕДЕНИЕ

В настоящее время все более широкое распространение получают глобальные автоматизированные производства, оснащенные микропроцессорной техникой. Тенденция развития производства постепенно идет по пути уменьшения требований к алгоритмическо-языковым и программно-техническим знаниям оператора. На базе микроконтроллерных систем создаются гибкие структуры, позволяющие наиболее быстро и эффективно, с требуемой надежностью обеспечить автоматическое управление каким-либо производственным процессом; а также имеющие возможности как быстрого перепрограммирования на новую технологию, так и встроенные аппаратные и программные возможности тестирования и самонастройки математической и алгоритмической части под новый, иногда неизвестный заранее объект регулирования. Новые требования предъявляются и к возможности связи микропроцессорных систем с персональным компьютером, для обеспечения более наглядного и формализованного представления производственного процесса, а также возможность создания локальных систем управления на основе нескольких микроконтроллеров.

Данные тенденции не обходят стороной и сферу электроэнергетики. Электроэнергетика является наиболее важной отраслью энергетики, что объясняется такими преимуществами электроэнергии перед энергией других видов, как относительная лёгкость передачи на большие расстояния, распределения между потребителями, а также преобразования в другие виды энергии (механическую, тепловую, химическую, световую и др.). Отличительной чертой электрической энергии является практическая одновременность её генерирования и потребления, т. к. электрический ток распространяется по сетям со скоростью, близкой к скорости света. Передача же электрической энергии от электрических станций до потребителей осуществляется по электрическим сетям. Электросетевое хозяйство — естественно-монопольный сектор электроэнергетики: потребитель может

выбирать, у кого покупать электроэнергию (т.е. энергосбытовую компанию), энергосбытовая компания может выбирать среди оптовых поставщиков (производителей электроэнергии), однако сеть, по которой поставляется электроэнергия, как правило, одна, и потребитель технически не может выбирать электросетевую компанию. С технической точки зрения, электрическая сеть представляет собой совокупность линий электропередачи (ЛЭП) и трансформаторов, находящихся на подстанциях.

Таким образом подобная сложная структура с множеством задач, требующих незамедлительного решения и контроля за выполнением, нуждается в глубоком и качественном управлении. Именно с этой целью в энергосбытовых компаниях используются мощные автоматизированные программно-аппаратные комплексы контроля параметров энергосистемы, предназначенные для приёма, обработки, передачи и хранения телеметрической информации о режиме работы энергетической системы (уровень АО-энерго, РДУ и ОДУ), поступающей в реальном времени от объектов генерации и трансформации электроэнергии а также других программно-аппаратных комплексов, и предоставления оперативно-диспетчерскому персоналу доступа к ней. Программная и техническая реализация данного комплекса является задачей важной, сложной и актуальной.

В данном дипломном проекте решаются задачи модернизации программно-аппаратного комплекса, эксплуатируемого в Курганском диспетчерском управлении энергосистемы области, ставится задача с помощью методов декомпозиционного анализа и синтеза создать на базе имеющихся программно-аппаратных средств SCADA-систему, способную собирать информацию и предоставлять управление энергообъектами.

Целью модернизации системы диспетчерского управления является:

- 1) обеспечение обратной связи с энергообъектами введением управляющих сигналов;
- 2) повышение эффективности управления подстанцией с целью

ведения заданного режима;

3) предотвращения отказов оборудования, локализации и устранения последствий отказов оборудования;

4) локализации и устранения последствий отказов оборудования;

5) повышение эффективности и комфортности работы оперативного и обслуживающего персонала;

6) получение качественной оперативной информации на соответствующих автоматизированных рабочих местах;

7) обеспечение возможности подробного ретроспективного анализа режимов работы основного электрооборудования.

# 1 Обоснование модернизации автоматизированной системы диспетчерского управления

## 1.1 Характеристика оборудования подстанции

Оперативно-информационная подсистема (ОИП), представляет собой программно-аппаратный комплекс и выполнена в сетевой распределенной архитектуре. Она может работать как на одиночном сервере, так и в виде сложных распределенных многомашинных комплексов. Клиенты, пользующиеся информацией подсистемы, работают на отдельных рабочих станциях и могут подключаться к серверам комплекса при помощи вычислительных сетей различного рода. Принципиально возможна работа клиента и сервера на одном компьютере.

В случае многомашинных комплексов появляется возможность объединять их в домены и группы комплексов, использовать функцию автоматического горячего резервирования, производить репликацию данных в реальном времени между всеми серверами домена комплекса. Количество групп в каждом домене и серверов в группе ограничено лицензионным соглашением и соответствующим ему лицензионным ключом, а также технической целесообразностью и здравым смыслом.

Доменом ОИП называется совокупность серверов, каждый из которых выполняет все или часть функций приема, обработки, архивирования информации о режиме работы и выработки управляющих воздействий для одного объекта управления, т.е., служащих одной цели. В одном центре управления может быть организовано любое число доменов ОИП, каждый из которых работает автономно и предназначен для выполнения своей функции: основной домен, тренажерный домен, полигон и т.д. Пользователь самостоятельно или при помощи администратора настраивает свой компьютер на работу с нужным доменом ОИП. Впоследствии он автоматически подключается к одному из серверов этого домена.

Группой ОИП называется совокупность серверов, выполняющих одни и



те же функции (быть может, в разные моменты времени) и обладающие в нормальном режиме работы идентичным содержимым баз данных и автоматически выравнивающие это содержимое после устранения аварийных ситуаций по основному серверу в группе.

Любые серверы в домене ОИП могут быть объединены в группу, в рамках которой осуществляется централизованная обработка и параллельное архивирование информации.

Общее число серверов в группе и групп в домене программным обеспечением специально не ограничивается.

В любой момент времени только один из серверов группы может являться основным. Остальные серверы могут быть резервными или временно отключенными.

Смена ролей серверов внутри группы и групп внутри домена производится автоматически или по команде оператора. Процесс смены ролей протекает незаметно для конечных пользователей. Пакеты серверных программ автоматически запускаются и «на лету» подхватывают информацию в соответствии с настройками при каждой смене конфигурации в рамках домена ОИП.

Основные компоненты комплекса ОИП приведены на рисунке 1.1.

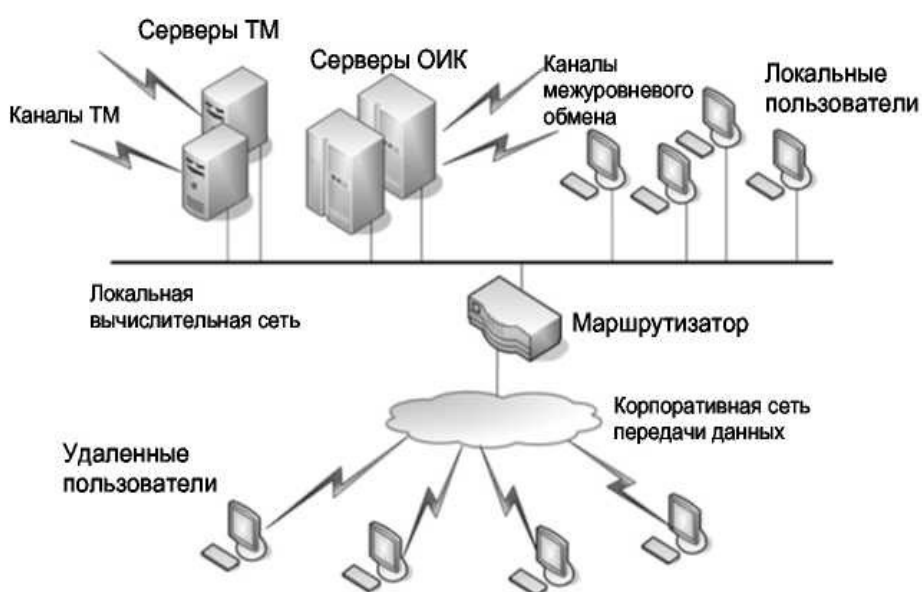


Рисунок 1.1 – Архитектура ОИП

ОИП в общем случае не занимается приемом данных непосредственно из каналов телемеханики и обработкой соответствующих телемеханических протоколов. Обмен данными с системой сбора информации осуществляется двумя способами — через серверы телемеханики и по каналам межуровневого (межмашинного) обмена. И в том и в другом случае используются соответствующие протоколы обмена. При использовании в устройствах телемеханики (УТМ) протокола МЭК 870-5-104, работающего поверх стека TCP/IP, ОИП может подключаться к таким устройствам напрямую или через соответствующие сетевые устройства, такие как маршрутизаторы, брандмауэры и т.д., играя роль клиента протокола МЭК 870-5-104, и получать телеметрическую информацию прямо из УТМ.

Подключение серверов телемеханики (центральных приемо-передающих станций) к ОИП может осуществляться как по вычислительной сети (обычно по локальной), так и по последовательному интерфейсу для некоторых унаследованных типов серверов ТМ.

Клиентскую часть комплекса, также как и серверную, можно разделить на две части: системное и прикладное ПО.

В состав системного ПО входят операционная система Microsoft Windows XP или 2000. .NET Framework 2.0 (для программы обработки макетированной информации MaketPLexe), программа MS Excel 2002/2003, используемая как генератор отчетов.

Прикладное ПО состоит из нескольких групп программ: библиотеки доступа к функциям БД РВ и сервиса ОИП rtdbcon.dll, Монитора отображения СК-2007 и связанных с ним библиотек и программ, студии редакторов НСИ, редакторов форм отображения и программы управления печатью отчетов, набора служебных программ.

Отдельного упоминания заслуживает программа Загрузчик ПО, предназначенная для автоматической загрузки и установки программного обеспечения на рабочих станциях клиентов из базы нормативно-справочной информации. Эта программа определяет, установлено ли затребованное

пользователем ПО на клиентской машине, если установлено, то определяется его версия, которая сравнивается с описанием, находящимся в НСИ. Если на сервере опубликована более свежая версия, то Загрузчик ПО считывает ее (программа может состоять из любого количества частей, включая exe, dll, reg, hip и другие модули), производит необходимые действия по установке и регистрации компонентов (например, для СОМ-объектов) и запускает программу с указанными в командной строке параметрами.

Данный подход позволяет существенно снизить затраты на установку программного обеспечения у клиентов, за исключением первоначальной настройки, а также изменения состава клиентского ПО, в связи с изменением роли пользователя в организации или смене версии ОИП, в результате которой появились новые программы, которые требуются тем или иным пользователям.

Все настройки, хранящие информацию о конфигурации программного обеспечения, а также те, которые пользователь производит на своей машине с ПО комплекса, запоминаются в системном реестре.

Для связи с базами данных, так же, как и на серверной стороне, используется Клиент БД РВ и технология ADO.

Каждая клиентская программа комплекса, за исключением некоторых служебных программ, в начале своей работы обращается к системе санкционирования доступа и проверяет, разрешено ли текущему пользователю (напрямую или за счет его включения в группы пользователей с необходимыми правами) использовать эту программу. Если права на работу с программой у пользователя есть, то программа начинает свою работу, иначе отображается соответствующее сообщение.

Лицом комплекса, с которым чаще всего сталкивается любой пользователь, является система отображения, состоящая из Монитора отображения, gol-файлов (динамических библиотек графической системы Graphic Object Library), дополнительных специализированных библиотек, исполняемого модуля векторной графики ТОПАЗ, набора форм, разработанных для каждого центра управления.

Все многообразие форм, стилей и внешних программ отображения опирается на фундамент общей панели управления Монитора отображения. Панель управления выполнена в виде всплывающего окна по аналогии со стандартной панелью задач MS Windows и отвечает за управление формами отображения, обеспечивает единообразный интерфейс пользователя, предоставляет формам доступ к базам данных ОИП через промежуточный слой доступа к БД РВ и ADO.

Монитор отображения строит дерево доступных ОИП-ов (получая информацию от сервиса ОИП, к которому подключен клиент), дерево форм на основании соответствующих таблиц НСИ и прав доступа к формам, предоставленных текущему пользователю, предоставляет возможность вызова форм за указанное время. Монитор отображения определяет необходимость загрузки каждой формы из базы данных. В случае если на диске клиентской машины этой формы еще нет, либо она уже устарела, Монитор отображения производит загрузку и, при необходимости, распаковку формы, проверяет тип и наличие свежей версии gol-файла, который умеет интерпретировать формы данного типа, также загружает его из базы и передает путь к файлу, содержащему запрошенную форму, соответствующей функции gol-файла.

В процессе работы с формой gol-файл общается с панелью управления посредством сообщений Windows. Панель управления предоставляет gol-файлам возможность для каждой открытой формы создавать собственную закладку на панели управления и размещать там необходимые органы управления формой, gol-файлы, разбирая структуру формы отображения, определяют список информации, которую следует показать, и обращаются за ней к панели управления. Панель управления обеспечивает установление и поддержание соединения с БДРВ. Посылку запросов и передачу ответов осуществляют соответствующие функции, размещенные в gol-файлах. Панель управления осуществляет коммутацию сообщений между экземплярами открытых форм, обрабатываемых любыми типами gol-файлов. При помощи

данного механизма производится выполнение таких функций, как навигация между формами, создание динамических наборов параметров с форм отображения, создание списков переключений ТС, переход к схемам объектов из списков событий и т.д.

В клиентское программное обеспечение входят средства создания форм, публикации форм и документов, различные редакторы НСИ, позволяющие настраивать комплекс. Состав клиентского ПО для конкретного пользователя определяется при установке клиента программного обеспечения комплекса в процессе выбора необходимых для работы компонент.

Редактор форм — средство для создания, редактирования и просмотра графических форм. Он позволяет использовать и отображать все типы оперативной информации, данные из нормативно-справочной базы ОИП, графические примитивы, и представлять информацию в виде графиков, схем, таблиц в различных комбинациях.

Публикатор — программа, выполняющая функцию записи подготовленных форм, графических примитивов и подложек, программного обеспечения и документов любого формата (MS Word, MS Excel, pdf, html и т.д.) в таблицы НСИ. После записи в базу данных эта информация становится доступной пользователям ОИП в соответствии с их уровнем доступа.

Графические подложки и примитивы создаются в любых имеющихся редакторах. В качестве примитивов (условных обозначений трансформаторов, реакторов, генераторов, выключателей и т.п.) могут использоваться рисунки только в формате bmp, подложки (географические, схемные) могут быть представлены в виде bmp-, gif-, jpeg-, emf-форматов.

Для формирования отчетов в комплексе используется средство работы с электронными таблицами MS Excel. Возможно применение данной программы, начиная с версии MS Excel 2002. Отчетные документы могут использовать весь богатый арсенал изобразительных и вычислительных средств, которыми обладает MS Excel, включая средства анализа и построения графиков. Данные для отчетов MS Excel получает из SQL-архива, хранящегося

в таблицах MS SQL Server-a. Для подключения к реляционной базе применяется технология ADO. Она не требует сложной настройки, и показала достаточную производительность при обработке запросов.

В архивной базе находится набор процедур, предоставляющий простой доступ к данным. В шаблонной таблице имеются функции, которые устанавливают связь с хранимыми процедурами в реляционной базе, обращаются с различными запросами и формируют отчеты.

Управление печатью отчетов оператор осуществляет через специальную программу, которая позволяет задать набор подготавливаемых отчетов, количество экземпляров и дату каждого документа. Перед печатью оператор может просмотреть каждый сформированный документ.

Настройка комплекса производится при помощи Студии НСИ, представляющей собой управляющий модуль, обеспечивающий единую точку входа и выбора редактируемой базы данных для большого набора редакторов НСИ. Эти редакторы позволяют в доступной для технологов форме вносить изменения в описания различных аспектов комплекса и записывать их в таблицы базы данных.

Служебные программы.

Для наладчиков, администратора ОИП и операторов, управляющих работой комплекса, предназначен набор сервисных программ, которые недоступны и не нужны большинству обычных пользователей. К этим программам относятся: Управление доступом, Управление задачами, Монитор БД РВ, Анализ сети ОИП, Программа генерации групповых пакетов и другие.

Управление доступом пользователей базируется на информации о пользователях и их членстве в группах Active Directory. Права предоставляются как пользователям, так и группам, которые являются представлением выполняемых функций или ролей. Любой пользователь может исполнять несколько ролей, т.е. одновременно входить в несколько групп. Его права определяются суперпозицией собственных прав и прав всех групп, в которые он входит.

Комплекс позволяет определить доступ субъектов к трем крупным типам классов защиты: Базы данных SQL-сервера, Объекты РБД (таблицы, представления, процедуры, функции), Функции и информация ОИП. Конкретные классы защиты относятся к одному из этих трех типов. Субъектами системы доступа являются пользователи и группы пользователей. Каждый субъект защиты может выполнять одну или несколько ролей. Для каждой роли могут быть определены права, такие как запуск клиентских приложений, добавление записей, редактирование записей, удаление записей, выполнение хранимых процедур, просмотр форм отображения, инициирование запуска серверных задач, право записи определенных категорий оперативной информации, генерация событий, установка и снятие диспетчерских пометок и т.д.

Программа «Управление задачами» позволяет осуществлять мониторинг загрузки серверов ОИП и состояния серверных задач, просматривать журнал работы любого сервера, управлять работой пакетов программ, изменять состав программ и пакетов, изменять параметры их запуска, менять «на лету» уровень подробности журнала работы серверных программ (не все программы поддерживают данную функцию), управлять внешними службами Windows, менять роли серверов внутри групп и состав групп ОИП, останавливать/запускать работу ОИП. Предоставляется три уровня доступа: просмотр, управление, администрирование. Система управления специально спроектирована для работы с сетью ОИП. Возможно дистанционное управление из единого центра по каналам передачи данных, предоставляющим сервис IP-соединения.

Для облегчения работы оператора предназначены программы мониторинга работы БД РВ. Одни из них позволяют в реальном времени просматривать поток поступающей и записанной информации, применяя различные фильтры и настройки. Другие предназначены для автоматического контроля доступности указанных БД РВ. В случае невозможности осуществить подключение к некоторой БД РВ в сети ОИП за заданное

оператором время начинает звучать сигнал тревоги.

Большой объем информации, хранящийся в базах ОИП, сложные взаимосвязи между отдельными элементами данных, наличие целой иерархии расчетных формул может вызвать трудность в определении причин появления некорректных значений. Для выполнения анализа и корректировки архивных данных предназначена программа «Анализ данных» (DECK.exe). Эта программа предназначена как для администраторов и операторов ОИП, так и для других пользователей, таких как специалисты по электрическим режимам, планированию графиков работы и т.д. Она обладает широкими возможностями по просмотру и выборке данных, анализу иерархии формул и входящих в них параметров, фильтрации по признакам, запросам данных из архива, просмотру любых таблиц НСИ через SQL-запросы, получения подробного паспорта любого параметра. С ее помощью можно:

- 1) скорректировать данные в архиве (за исключением архивов ТИ, ТС и событий);
- 2) запустить повторный перерасчет или заново сформировать выбранные типы отчетных данных за указанное время;
- 3) поставить ТИ или ТС на ручной ввод или объявить недоверие телемеханике для выбранного параметра;
- 4) найти любой параметр по идентификатору или подстроке в его имени;
- 5) выполнить достаточно сложный анализ существующей нормативно-справочной информации при помощи набора из более чем 20 шаблонов аналитических SQL-запросов и произвести экспорт результатов анализа в xml, html, xls-форматы;
- 6) проанализировать скачки и экстремумы в заданном временном интервале;
- 7) произвести статистический экспресс-анализ поведения выбранного параметра за указанный интервал времени: сумма, среднее, минимум, максимум, среднеквадратичное отклонение, дисперсия, эксцесс, коэффициент



асимметрии;

8) произвести экспорт в MS Excel и при помощи его средств построить гистограммы изменений;

9) произвести восстановление или сравнение архивов БД РВ на разных серверах (функция доступна только администратору ОИП);

10) анализировать состав и управлять импульс-архивами.

Данная задача — необходимый инструмент администратора ОИП, дежурного оператора и специалистов, занятых анализом режимов.

В комплексе есть и другие вспомогательные сервисные средства: программа анализа отладочных сообщений, имитатор и анализатор работы сети ОИП, генератор тестовых сигналов, преобразование времени и т.д.

Одной из наиболее глубоко интегрированных в комплекс задач является графическая система Toraz Graphics. Эта система расширяет возможности отображения графической информации комплекса и предоставляет средства для создания и работы со сверхбольшими схемами (в том числе на проекционных экранах), плавное масштабирование, управление слоями и яркостью, отображение переключений, произошедших на объектах нижнего уровня, на их условных обозначениях, представленных на верхних уровнях, предоставление интерфейса для ведения диспетчерских пометок на схемах, создание геоинформационного представления электрической сети. Система отображения Toraz работает под управлением панели управления программного обеспечения комплекса, так же как и стандартные формы, может обращаться к любым данным ОИП, и практически неотличима для конечного пользователя от других форм отображения.

Важной функцией любого ОИП является оценка состояния (получение сбалансированного установившегося режима) по данным телемеханики.

Программное обеспечение комплекса сопряжено с программой оценки КОСМОС-2004. Имеется возможность как циклических расчетов с получением вектора оцененных параметров режима практически в реальном времени (на реальных схемах объемом 400-600 узлов успешно проводились расчеты с

циклом в 10 секунд), так и экспресс-расчетов в интерактивном режиме по заданию диспетчера, с последующим проигрыванием ситуации по включению/отключению оборудования и анализом ситуации.

Режимный диспетчерский тренажер ФЕНИКС также сопряжен с программным обеспечением ОИП. Он предназначен для проведения противоаварийных тренировочных учений оперативно-диспетчерского персонала энергосистем и энергообъединений и позволяет моделировать основные оперативные состояния электрической сети в нормальных и послеаварийных режимах.

Тренажер получает срезы телеинформации текущего или архивного режима из базы данных реального времени и использует их для подготовки тренировок. Во время проведения тренировки осуществляется передача в ОИП потока данных, имитирующих поступление телеинформации, отражающей моделируемый режим и состояние оборудования энергосистемы. Диспетчер со своего рабочего места или с любого компьютера в локальной сети по выбору может подключаться, как к основному (подключение по умолчанию), так и к тренажерному ОИП.

ОИП осуществляет прием телемеханики от предвключенных устройств — серверов ТМ и других внешних систем. В текущей версии реализован обмен с устройствами типа SmartFER, Motorola (программное обеспечение ДИСПЕТЧЕР-4). RS-6000 (программное обеспечение ДИСПЕТЧЕР-5). РПТ-80 (обмен по интерфейсу RS-232 в протоколе RPT-ЭВМ), Систел, Интерфейс, КОТМИ.

Имеются модули связи по следующим протоколам: МЭК 870-5-104, АИСТ (по последовательному порту), расширенный протокол RPT-ЭВМ (RPT-ЭВМ+) поверх UDP, FDST, FER1/2, СИСТЕЛ. Серверы телемеханики и иные узлы обмена информацией, с которыми обменивается ОИП, могут быть дублированы. В случае нарушения приема информации от одного из устройств группы автоматически происходит переключение на прием от резервного устройства.

ОИП способен принимать телеизмерения (ТИ), закодированные в 7, 8, 12 и 16 двоичных разрядах, в том числе с учетом знака для квантов; телесигналы (ТС), в том числе многопозиционные (вкл, откл., промежуточное состояние), включая информацию о сигналах повреждения устройств и каналов телемеханики, положениях анцапф переключателей ответвлений обмоток трансформаторов и автотрансформаторов с регулированием коэффициента трансформации под нагрузкой, отчетную цифробуквенную информацию, представленную в виде макетов, состав информации в которых описан в нормативной базе, а средствами распаковки и упаковки в требуемый формат могут служить специальные дополнительные VB-скрипты, которые при необходимости могут разработать специалисты по информационным технологиям в каждом центре управления.

Программы приема телемеханики следят за состоянием каналов и устройств ТМ и узлов обмена информацией, в случае изменения их состояния генерируют соответствующие события и сигналы, позволяющие отслеживать состояние сети телемеханики практически любой сложности и конфигурации.

Помимо функции приема информации в ОИП предусмотрена возможность передачи (ретрансляции) любых числовых данных в заданные узлы обмена информацией. Эта возможность может применяться для передачи по каналам межмашинного обмена или по каналам ТМ оперативных, отчетных и плановых данных. Эти данные могут быть рассчитаны в ОИП, могут попасть в базу данных ОИП из внешних приложений, могут быть получены по каналам телемеханики/ММО с объектов управления и пройти необходимые процедуры достоверизации и восстановления. Для принимающей стороны такие данные выглядят как телеметрическая, отчетная или плановая информация. Механизм ретрансляции реализован через передачу данных из ОИП в назначенный узел обмена по тому протоколу, который указан для этого узла, с тем регламентом (по изменению, циклически, адаптивно-циклически, по расписанию и т.д.), который указан для конкретного набора обмена.

Для прямого обмена данными между ОИП-ами смежных уровней

диспетчерского управления или смежными энергосистемами применяется механизм межмашинного обмена (ММО). В отличие от ретрансляции, производимой по каналам телемеханики, ММО осуществляется по каналам передачи данных. При этом в качестве сетевых и транспортных протоколов используется стек протоколов TCP/IP. Применяются различные варианты организации каналов ММО — работа в общих каналах передачи данных с или без поддержки механизма QoS (Quality of Service), организация отдельных или виртуальных каналов для ММО. В качестве протокола прикладного уровня для ММО в ОИП реализован протокол FDST. Межмашинный обмен не имеет жестких ограничений по скорости и объемам передачи данных. На высших уровнях диспетчерского управления ММО является главным каналом передачи данных между ОИП-ами этих уровней. Телемеханические каналы выступают в качестве резерва.

При использовании протокола FDST, в отличие от телеметрических протоколов, имеется возможность передачи временных рядов наборов данных, которая позволяет за одну посылку передать в другой центр управления диспетчерский график по всем групповым объектам управления на следующие сутки, новый план балансирующего рынка.

Для обеспечения функции обмена планами и диспетчерскими графиками в ОИП предусмотрена специальная настройка и обработка. Наборы плановых данных, относящихся к планам балансирующего рынка, должны быть специальным образом отмечены при настройке, после чего их отправка может осуществляться автоматически, например, из ОИП уровня объединенного диспетчерского управления (ОДУ) во все подчиненные региональные диспетчерские управления (РДУ). Триггером для отправки служит событие о том, что получен новый план балансирующего рынка (ПБР) или о том, что в силе остается предыдущий ПБР. Эти события генерируются специальной программой-загрузчиком ПБР или сервером FDST при получении нового ПБР по каналам ММО. Наличие данного механизма предоставляет возможность, например, автоматически распространить ПБР из центрального

диспетчерского управления (ЦДУ) во все ОДУ и далее во все РДУ.

Сеть обмена информацией с точки зрения ОИП состоит из устройств и каналов передачи телеметрической информации. К устройствам относятся, в первую очередь, разнообразные устройства телемеханики, расположенные на энергетических объектах. Кроме того, к устройствам, образующим сеть телемеханики, можно отнести серверы телемеханики (ЦППС), ОИП (если они умеют обмениваться друг с другом телеметрической информацией в реальном времени), серверы и устройства АИИС КУЭ (с определенной долей условности), системы АСУ ТП на энергетических объектах.

Сеть обмена информацией может иметь произвольную структуру — в виде звезды, иерархическую, магистральную, сложную сетевую.

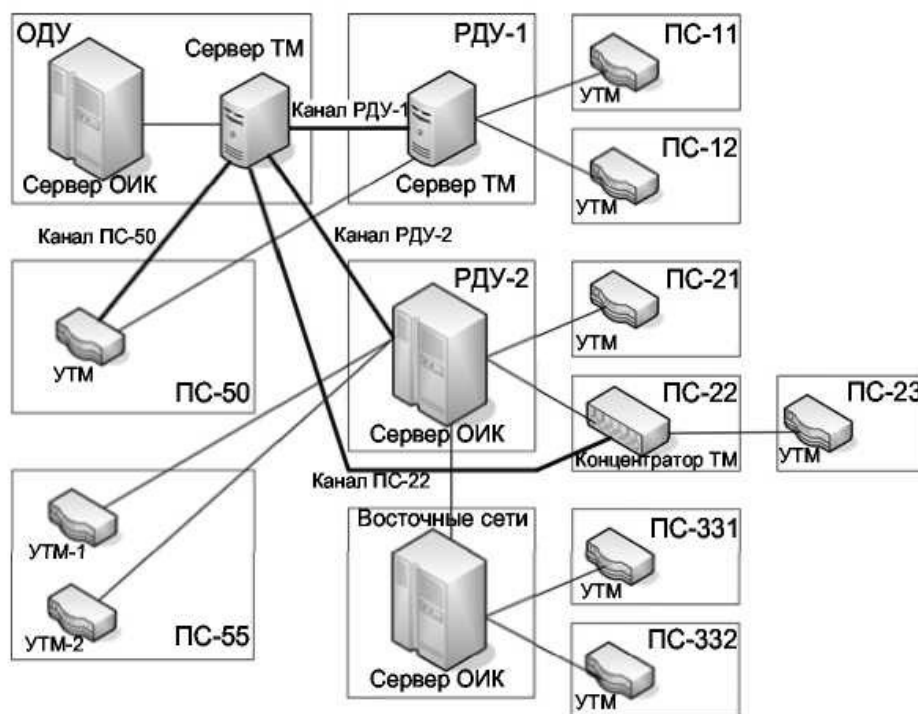


Рисунок 1.2 – Фрагмент сети обмена оперативной информацией

Приведенный фрагмент сети состоит из нескольких блоков. На верхнем уровне, в данном случае — ОДУ, показаны серверы ОИП и телемеханики (в действительности могут работать по несколько экземпляров каждого вида).

Телеметрическая информация, состоящая в данном контексте из телеизмерений (разного рода - мгновенных, интегральных) и телесигналов, порождается на энергетических объектах и передается в управляющие или

информационные системы при помощи устройств телемеханики. Поэтому каждый ТИ и каждый ТС с одной стороны, принадлежит некоторому энергетическому объекту, а с другой - некоторому устройству телемеханики. Дерево (сеть) обмена информацией не может повторять иерархию энергообъектов, так как последняя отражает хозяйственную или территориальную принадлежность, а сеть обмена - маршруты и способы доставки информации [1].

Структурная схема системы диспетчерского управления подстанции представлена на листе графической части дипломного проекта.

Система телемеханики ПС «Левашово» представляет собой иерархическую систему оборудования телемеханики, устанавливаемого в шкаф основного контроллера телемеханики «1ТМ» марки «КП-Исеть» производства «НТК Интерфейс», связанного со шкафами технологического оборудования электрической подстанции ПС «Левашово» 110 кВ и 10 кВ.

Основные функции проектируемой системы телемеханики следующие:

- 1) сбор и первичная обработка аналоговой и дискретной информации от электротехнического оборудования и систем, обеспечивающих функционирование подстанции;
- 2) управление коммутационными аппаратами;
- 3) регистрация событий;
- 4) оперативная блокировка разъединителей;
- 5) контроль качества и технический учет электроэнергии;
- 6) интеграция смежных систем.

Соединение с управляющими контроллерами в каждом технологическом шкафу со шкафом «1ТМ» осуществляется в основном при помощи интерфейса RS-485. В качестве кабелей связи применяются экранированные кабели, прокладываемые в помещении под фальшполом в металлических коробах. Проектируемый шкаф телемеханики устанавливается совместно с технологическими шкафами в помещении релейных панелей.

Передача данных телемеханики осуществляется в Курганское РДУ по

протоколу МЭК 60870-5-104. В качестве транспортного протокола для передачи данных применяется технология Ethernet. Контроллер телемеханики подключается по интерфейсу RJ-45 к маршрутизатору Cisco 1921 и далее передается по каналам ВОЛС в помещение связи в Курганском РДУ.

Подстанция оснащена локальной системой управления, реализующей функции контроля напряжения вторичной обмотки автотрансформаторов, переключения секций на базе регулятора РНМ-1. Существующая система не имеет возможности удаленного управления.

Структурная схема системы телемеханизации подстанции представлена на листе графической части дипломного проекта.

## 1.2 Анализ задач диспетчерского управления

Состав и объёмы функций, реализуемых АСУД, существенно различаются для разных уровней оперативно-технологического управления. Курганское РДУ реализует следующие функции:

- 1) сбор и обработка информации (SCADA) – формирование базы данных АСДУ, диалог персонала с ЭВМ, сбор информации, телеуправления, документирование;
- 2) управление основной электрической сетью – формирование схемы контролируемой сети, эквивалентирование, оценивание состояния, анализ возможных аварийных ситуаций, расчет установившегося режима в темпе процесса, расчет токов КЗ в темпе процесса, расчет штрафных коэффициентов, оптимизация режима электрической сети, исследование режимов, разработка графиков напряжений и реактивных мощностей, оптимизация режима с учетом ограничений по надежности, переключения с учетом требований надежности;
- 3) управление распределительными электрическими сетями – формирование схемы контролируемой электрической сети, распределение нагрузки по узлам электрической сети, определение места повреждения и отделение его от остальной части, неповрежденной части ЭЭС, расчеты

потокораспределения, расчеты токов КЗ, управление ремонтными работами по нарядам, оптимизация потокораспределения, управление трансформаторами и синхронными компенсаторами, оптимизация точек раздела электрической сети;

4) управление энергетическими режимами – АРЧМ, контроль за резервами, краткосрочный прогноз нагрузки, расчеты межсистемных обменов электроэнергией, оптимизация режима по активной мощности, управление нагрузкой, отключения нагрузки, выбор состава оборудования, координация работы ТЭС и ГЭС;

5) обучение и тренировка диспетчерского персонала – функции инструктора, функции тренируемого, моделирование процессов в ЭЭС;

6) применение экспертных систем – интеллектуальная обработка аварийной информации, анализ аварийных нарушений, советы диспетчеру по восстановлению погашенного участка электрической сети, диагностика ОИУК.

Ниже приведены данные, характеризующие состав функций, реализуемых современными ОИУК [2]:

- автоматическое регулирование частоты и активной мощности, оперативная оптимизация режима по активной мощности без учета и с учетом ограничений по электрической сети;

- выбор состава работающих агрегатов;

- оценивание состояния;

- телеуправление выключателями, РПН трансформаторов, идентификация режима ЭЭС;

- фиксация нарушений установленных пределов значений параметров режима;

- оперативный расчёт потокораспределения;

- оперативная оптимизация режима электрической сети по напряжениям и реактивным мощностям;

- анализ возможных аварийных ситуаций;

- расчёты за поставку и обмен электроэнергией.



### 1.3 Анализ тенденций развития систем диспетчерского управления

Диспетчерское управление и сбор данных (SCADA Supervisory Control And Data Acquisition) является основным и в настоящее время остается наиболее перспективным методом автоматизированного управления сложными динамическими системами (процессами) в жизненно важных и критичных с точки зрения безопасности и надежности областях. Это связано со значительным прогрессом в области вычислительной техники, программного обеспечения и телекоммуникаций.

Тем не менее, количество аварий существенно не сократилось. Это связано с тем, что развитие информационных технологий, повышение степени автоматизации и перераспределение функций между человеком и аппаратурой обострило проблему взаимодействия человека-оператора с системой управления. Последнее отмечается в ряде работ. Так например, в [2] указывается, что в 60-х годах ошибка человека являлась первоначальной причиной лишь 20% аварий (80%, соответственно, за технологическими неисправностями и отказами), то в 90-х годах доля человеческого фактора возросла до 80%, причем, в связи с постоянным совершенствованием технологий и повышением надежности электронного оборудования и машин, доля эта может еще возрасти (рис. 1.3).

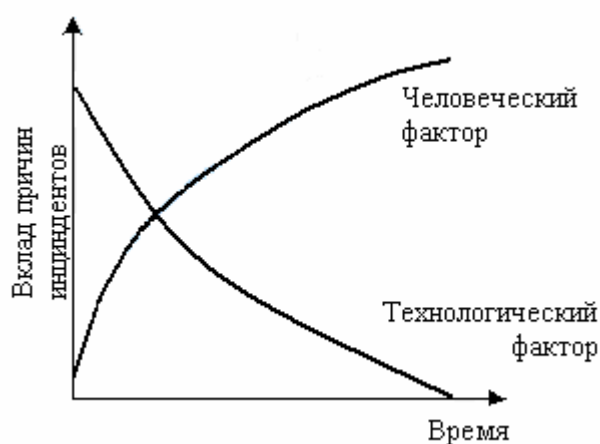


Рисунок 1.3 – Тенденции причин аварий в сложных автоматизированных системах

Основной причиной таких тенденций является старый традиционный подход к построению сложных автоматизированных систем управления, который применяется часто и в настоящее время: ориентация в первую очередь на применение новейших технических (технологических) достижений, стремление повысить степень автоматизации и функциональные возможности системы и, в то же время, недооценка необходимости построения эффективного человеко-машинного интерфейса.

Прогресс в области информационных технологий обусловил развитие всех 3-х основных структурных компонентов систем диспетчерского управления и сбора данных – диспетчерских пунктов управления, удаленных терминалов и каналов связи, что позволило значительно увеличить их возможности.

Основная тенденция развития технических средств (аппаратного и программного обеспечения) SCADA миграция в сторону полностью открытых систем. Открытая архитектура позволяет независимо выбирать различные компоненты системы от различных производителей; в результате расширение функциональных возможностей, облегчение обслуживания и снижение стоимости SCADA-систем.

Главной тенденцией развития диспетчерских пунктов управления является переход большинства разработчиков SCADA-систем на архитектуру клиент-сервер, состоящую из 4-х функциональных компонентов [3]:

1) интерфейс пользователя – исключительно важная составляющая систем SCADA. Для нее характерны:

а) стандартизация интерфейса пользователя вокруг нескольких платформ;

б) все более возрастающее влияние Windows NT;

в) использование стандартного графического интерфейса пользователя (GUI);

г) технологии объектно-ориентированного программирования: DDE, OLE, Active X, OPC (OLE for Process Control), DCOM;

д) стандартные средства разработки приложений, наиболее популярные среди которых, Visual Basic for Applications (VBA), Visual C++.

2) управление данными – отход от узкоспециализированных баз данных в сторону поддержки большинства корпоративных реляционных баз данных (Microsoft SQL, Oracle). Функции управления данными и генерации отчетов осуществляются стандартными средствами SQL; эта независимость данных изолирует функции доступа и управления данными от целевых задач SCADA, что позволяет легко разрабатывать дополнительные приложения по анализу и управлению данными;

3) сети и службы – переход к использованию стандартных сетевых технологий и протоколов. Службы сетевого управления, защиты и управления доступом, мониторинга транзакций, передачи почтовых сообщений, сканирования доступных ресурсов (процессов) могут выполняться независимо от кода целевой программы SCADA, разработанной другим вендором;

4) службы реального времени – освобождение диспетчерских пунктов от нагрузки перечисленных выше компонентов дает возможность сконцентрироваться на требованиях производительности для задач реального и квази-реального времени. Данные службы представляют собой быстродействующие процессоры, которые управляют обменом информацией с терминалами и SCADA-процессами, осуществляют управление резидентной частью базы данных, оповещение о событиях, выполняют действия по управлению системой, передачу информации о событиях на интерфейс пользователя (оператора).

Системы диспетчерского управления имеют тенденцию развиваться в направлении интеграции компонентов, развивая идеи интегрированных систем управления.

#### 1.4 Функционально-стоимостной анализ базового варианта

Функционально-стоимостная модель объекта пригодна для выявления

ненужных функций и элементов в объекте (бесполезных и вредных); определения функциональной достаточности и полезности материальных элементов объекта; распределения затрат по функциям; оценки качества исполнения функций; выявления дефектных функциональных зон в объекте; определения уровня функционально-структурной организации изделия. Построение функционально-стоимостной модели осуществляется путем суперпозиции функциональной и структурной моделей объекта.

Оценка значимости функции ведется последовательно по уровням функциональной модели (сверху вниз), начиная с первого. Для главной и второстепенной, т.е. внешних функций объекта при оценке их значимости исходным является распределение требований потребителей (показателей качества, параметров, свойств) по значимости (важности).

Нормирующим условием для функции является следующее:

$$\sum_{j=1}^n r_{ij} = 1 \quad , \quad (1.1)$$

где  $r_{ij}$  - значимость  $j$ -ой функции, принадлежащей данному  $i$ -ому уровню функциональной модели;

$j=1,2,...,n$ ;

$n$  - количество функций, расположенных на одном уровне функциональной модели и относящихся к общему узлу вышестоящего уровня.

Для внутренних функций определение значимости ведется исходя из их роли в обеспечении функций вышестоящего уровня.

Для функционально-стоимостного анализа построим структурную и функциональную модели базового варианта. Структурная модель (СМ) - это упорядоченное представление элементов объекта и отношений между ними, дающее представление о составе материальных составляющих объекта, их основных взаимосвязях и уровнях иерархии. Базовым вариантом примем регулятор РНМ-1 с электрическими аппаратами, требуемыми для его подключения к приводу.

Структурная и функциональная модели базового варианта объекта представлена на рисунках 1.4 и 1.5.

Функциональная модель (ФМ) – это логико-графическое изображение состава и взаимосвязей функций объекта, получаемое посредством их формулировки и установления порядка подчинения.

После построения структурной модели на ее основании строим функциональную модель. При этом используем экспертные оценки значимости функций структурных составляющих объекта.

Проведем функционально-стоимостный анализ для разрабатываемой системы управления параметрами подстанции, как важного структурного элемента программно-аппаратного комплекса.

Построение совмещенной функционально-стоимостной модели:

ФСМ объекта пригодна для выявления ненужных функций и элементов в объекте (бесполезных и вредных); определения функциональной достаточности и полезности материальных элементов объекта; распределения затрат по функциям; оценки качества исполнения функций; определения уровня функционально-структурной организации изделия.

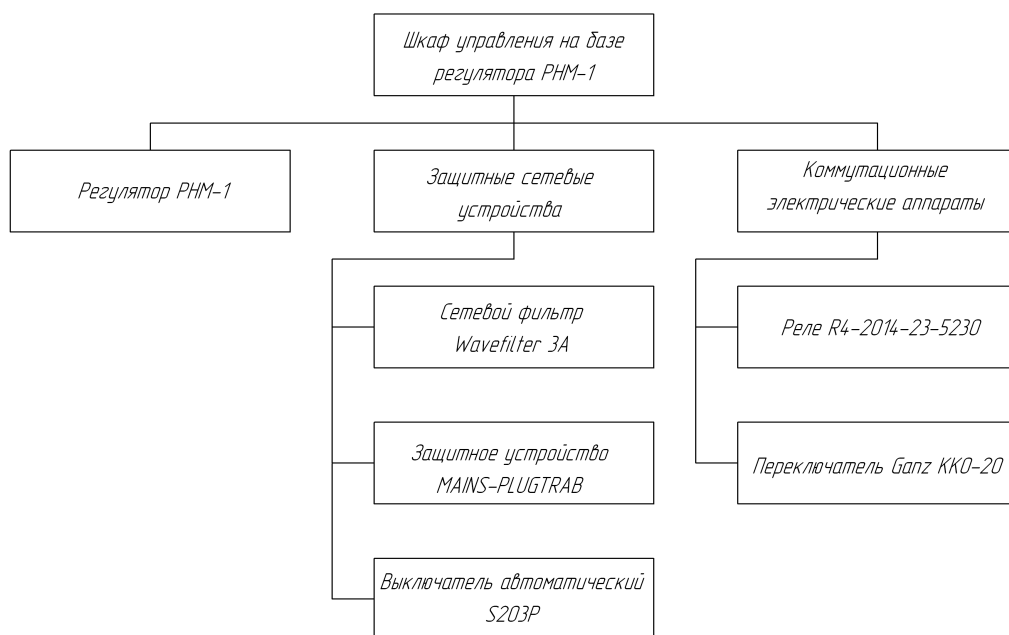


Рисунок 1.4 – Структурная модель базового варианта

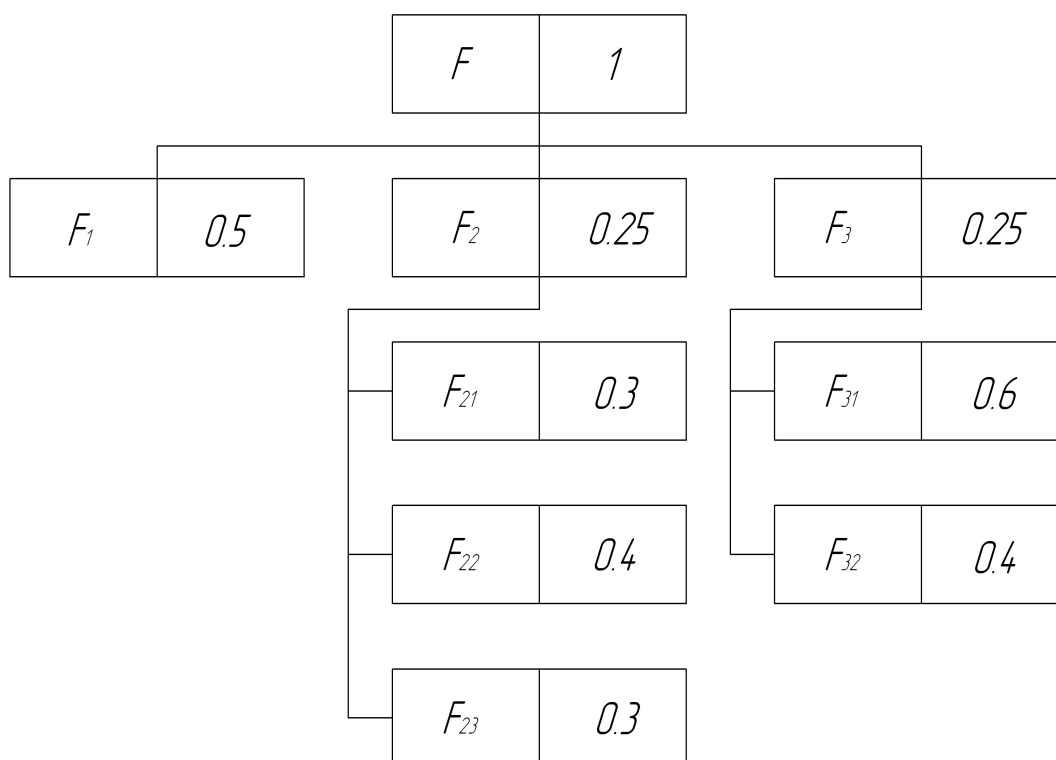


Рисунок 1.5 – Функциональная модель базового варианта

Построение ФСМ осуществляется путем совмещения ФМ и СМ объекта.

ФСМ базового варианта системы управления в таблице 1, где  $r$  – значимость функции;  $R$  – относительная важность функции;  $Q$  – качество исполнения функции;  $S_{\text{абс}}$  – абсолютная стоимость реализации функции;  $S_{\text{отн}}$  – относительная стоимость реализации функции

Учитывая многоступенчатую структуру функциональной модели, наряду с оценкой значимости функций по отношению к ближайшей вышестоящей определяется показатель относительной важности функции любого  $i$ -го уровня  $R_{ij}$  по отношению к изделию в целом:

$$R_{ij} = \prod_i^{G-1} r_{ij}, \quad (1.2)$$

где  $G$  - количество уровней функциональной модели.

В случае, если одна функция участвует одновременно в обеспечении нескольких функций верхнего уровня функциональной модели, ее значимость

определяется для каждой из них отдельно, а относительная важность функции для объекта в целом рассчитывается как сумма значений  $R_{ij}$  по каждой ветви функциональной модели (от  $i$ -го уровня до первого), проходящей через эту функцию.

Обобщенный (комплексный) показатель качества варианта исполнения функций оценивается по формуле:

$$Q_v = \sum_{n=1}^m \beta_n * P_{nv}, \quad (1.3)$$

где  $\beta_n$  - значимость  $n^{го}$  потребительского свойства;

$P_{nv}$  - степень удовлетворения  $n^{го}$  свойства в  $v^{ом}$  варианте;

$m$  - количество свойств.

Важным элементом качества исполнения функций является функциональная организованность изделий. Показатель актуализации функций определяется коэффициентом актуализации:

$$k_{aF} = \frac{F_{\Pi}}{F_{об}}, \quad (1.4)$$

где  $F_{\Pi}$  - необходимые функции;  $F_{об}$  - общее количество действительных функций;

$$K_{aF} = 5/7 = 0,71$$

— показатель сосредоточения функций, определяется коэффициентом сосредоточения:

$$k_{cF} = \frac{F_{осн}}{F_{об}}, \quad (1.5)$$

где  $F_{осн}$  - количество основных функций;

$F_{об}$  - общее количество функций;

$$k_{cF} = 6/7 = 0,85$$

– показатель совместимости функций, определяется коэффициентом совместимости:

$$k_{\text{совмF}} = \frac{F_c}{F_{\text{об}}} , \quad (1.6)$$

где  $F_c$  - функции согласования;

$F_{\text{об}}$  - общее количество функций;

$$k_{\text{совмF}} = 3/7 = 0,43$$

– показатель гибкости функций, определяется коэффициентом гибкости:

$$k_{\text{гF}} = \frac{F_p}{F_{\text{п}} + F_p} , \quad (1.7)$$

где  $F_p$  - количество потенциальных функции;

$F_{\text{п}}$  - количество необходимых функций.

$$k_{\text{гF}} = 2/(7+2) = 0,22$$

Учитывая (1.4), (1.5), (1.6), (1.7), выражение качества выполнения функций будет иметь вид:

$$Q_{\text{Fv}} = k_{\text{аF}} * k_{\text{сF}} * k_{\text{совмF}} * k_{\text{гF}} * \left( \sum_{n=1}^m \beta_n * P_{\text{nv}} \right) \quad (1.8)$$

Функционально необходимые затраты - минимально возможные затраты на реализацию комплекса функций объекта при соблюдении заданных требований потребителей (параметров качества) в условиях производства и применения (эксплуатации).

Абсолютная стоимость реализации функций  $S_{\text{абс}}$  определяется по формуле:

$$S_{\text{абс}} = S_{\text{изг}} + S_{\text{экспл}} + S_{\text{тр}} + S_{\text{эн}} + S_{\text{проч}} , \quad (1.9)$$

где  $S_{\text{изг}}$  - затраты, связанные с изготовлением (приобретением) материального носителя(-ей) функции. В состав этих затрат входят: затраты на проектирование, изготовление (модернизацию), пуско-наладочные



работы, обучение персонала.

$S_{\text{экспл}}$  - эксплуатационные затраты;

$S_{\text{тр}}$  - затраты, связанные с трудоемкостью реализации функции;

$S_{\text{эн}}$  - энергозатраты на реализацию функции;

$S_{\text{проч}}$  - прочие затраты на реализацию функции (отвод земли, изыскания, плата за загрязнение и пр.).

Относительная стоимость реализации функций  $S_{\text{отнF}}$  определяется по формуле:

$$S_{\text{отнFi}} = \frac{S_{\text{абсFij}}}{\sum_1^n S_{\text{абс}}}, \quad (1.10)$$

где  $\sum S_{\text{абс}}$  - суммарная абсолютная стоимость функционирования объекта, определяется путем суммирования значений абсолютных стоимостей реализации функций;

$S_{\text{абсFij}}$  - абсолютная стоимость реализации  $j^{\text{ой}}$  функции  $i^{\text{го}}$  уровня функциональной модели.

Таблица 1.1 – Функционально-стоимостная модель для базового варианта.

Индекс функции	Наименование функции	Материальный носитель ф-и	r	R	Q	$S_{\text{абс}}$	$S_{\text{отн}}$
1	2	3	4	5	6	7	8
F <sub>2.1</sub>	Фильтрация напряжения	Сетевой фильтр Wavefilter 3A	0.3	0.075	0.1	3100	0.056
F <sub>2.2</sub>	Защита от перенапряжения	Защитное устройство MAINS	0.4	0.1	0.1	2150	0.04
F <sub>2.3</sub>	Защита от КЗ	Автоматический выключатель S203P	0.3	0.075	0.1	720	0.013
F <sub>3.1</sub>	Коммутация сигналов	Реле R4-2014-23-5230-W TL	0.6	0.15	0.2	1800	0.033
F <sub>3.2</sub>	Переключение режимов	Переключатель Ganz KK0	0.4	0.1	0.2	1800	0.033
F1	Регулирование привода	Регулятор PHM-1	0.5	0.5	0.5	45000	0.82
F2	Питание комплекса	Сетевой фильтр, защитное устройство	0.25	0.25	0.2	6320	0.11
F3	Коммутация сигналов	Реле и переключатели	0.25	0.25	0.2	3600	0.066

$$\sum S_{\text{абс}} = 54920$$

Диаграммы ФСД и КИФ строятся для базового варианта до принятия проектного решения.

Построение функционально-стоимостных диаграмм (ФСД) и диаграмм качества исполнения функций (КИФ):

Данные диаграммы строятся для базового (на основании таблицы 1.1) варианта исследуемого объекта. Они имеют целью выявление зон диспропорции, т.е. зон избыточной затратности реализации функции, а также определение зон функциональной недостаточности (низкого качества исполнения функций).

При анализе диаграмм видно, что зоны диспропорции устраняются, а качество исполнения функций возрастает (рис. 1.6, рис. 1.7).

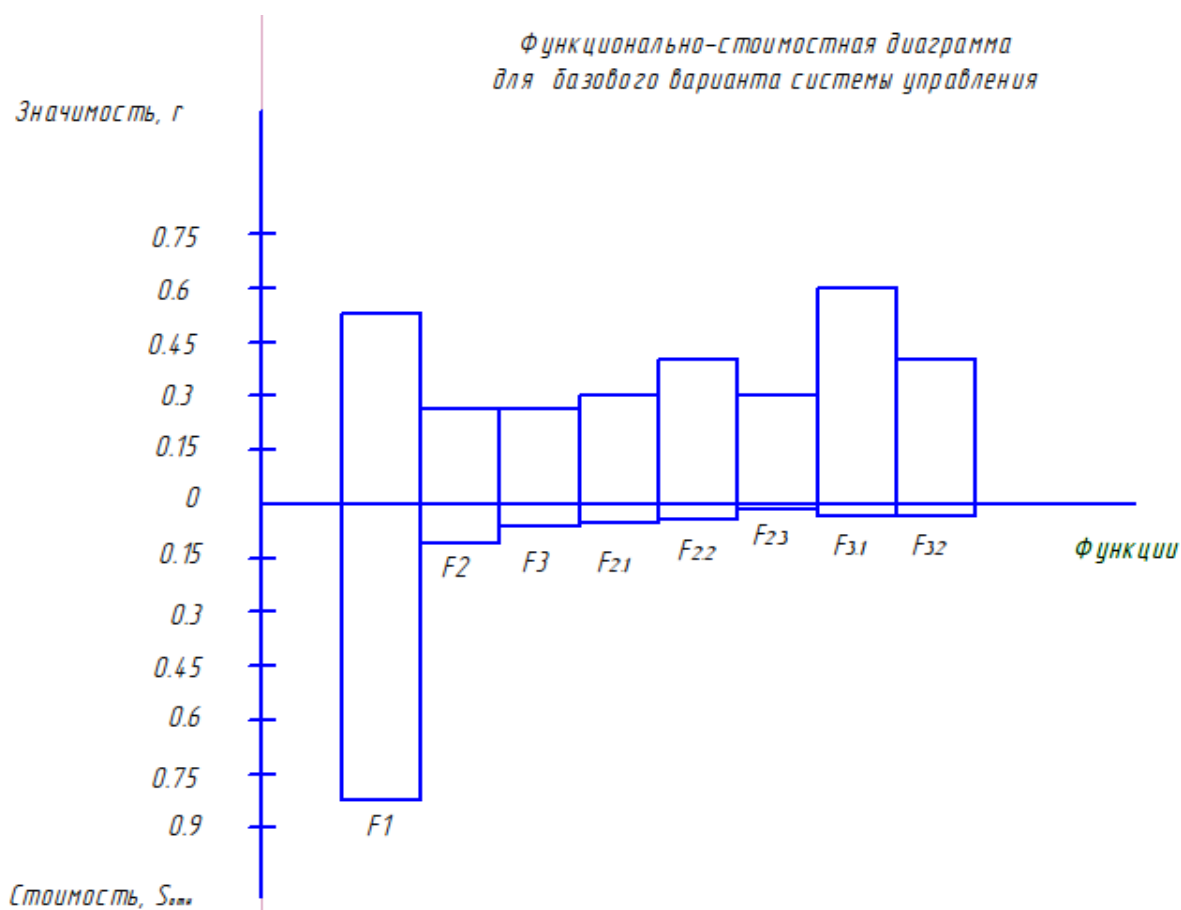


Рисунок 1.6 - Функционально-стоимостная диаграмма для базового варианта

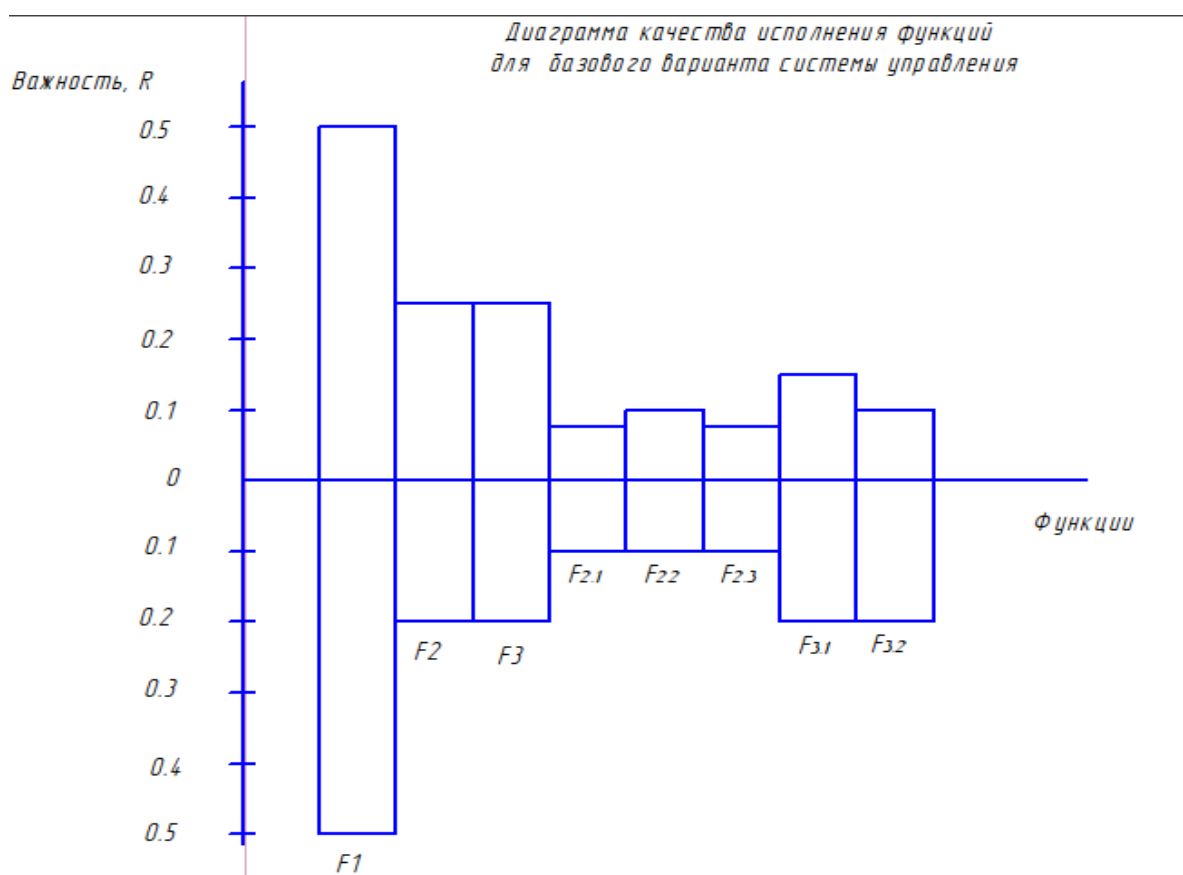


Рисунок 1.7 - Диаграмма качества исполнения для базового варианта

### 1.5 Техническое задание на проект

Наименование и область применения:

Проект модернизации автоматизированной системы диспетчерского управления подстанции «Левашово» предназначен для Курганского регионального диспетчерского управления энергосистемы области (Курганское РДУ) с целью улучшения эксплуатационных характеристик и расширением функциональных возможностей существующей SCADA системы.

Основание для разработки:

Проект модернизации автоматизированной системы диспетчерского управления подстанции «Левашово» разработан на основании задания на дипломный проект по специальности 220301 «Автоматизация технологических процессов и производств», полученного в соответствии с учебным планом Курганского Государственного Университета.

Цель и назначение разработки:

Целью модернизации автоматизированной системы диспетчерского управления является улучшение эксплуатационных характеристик и расширение функциональных возможностей существующей SCADA системы, эксплуатируемой в Курганском РДУ, а именно: обеспечение оперативного телеуправления технологическими параметрами подстанции введением управляющего воздействия и подключением локальной системы управления подстанции к диспетчерской системе.

Источники разработки:

Основными источниками для разработки послужили литература по автоматизации диспетчерского управления, документация на программно-аппаратный комплекс диспетчерского пункта, проектная документация подстанции «Левашово», техническая документация программируемого контроллера Овен, документация регулятора РНМ, а также справочная информация к программному пакету разработки QT Creator (C++).

Технические требования:

Система диспетчерского управления должна:

- 1) обеспечивать АРМ диспетчера программой контроля;
- 2) обеспечивать прием данных по протоколам МЭК 101/104;
- 3) обеспечить возможность регулирования напряжения автотрансформаторов и управлять переключателями;
- 4) обеспечивать достоверизацию принимаемой информации;
- 5) анализировать возможные ошибки управления и обрабатывать их согласно алгоритму.

Условия эксплуатации:

- температура окружающего воздуха в диапазоне (от -40 С до 45 С);
- относительная влажность от 10 до 99 %;
- атмосферное давление от 84 до 107 кПа;
- запыленность не более 5 мг/м<sup>3</sup>;
- наличие помех от магнитных полей не должны превышать значения

250 мВб/м<sup>2</sup>;напряжение питания трехфазным переменным током должно быть в пределах:  $200\text{В}^{+15\%}_{-10\%}$  ;

– частота питающей сети должна быть 50..60 Гц.

Стадии и этапы разработки:

- 1) Получение задания на модернизацию автоматизированной системы диспетчерского управления подстанции «Левашово»;
- 2) сбор и анализ документации;
- 3) Разработка структуры программно-аппаратных средств диспетчерского пункта и подстанции;
- 4) Разработка шкафа управления подстанции;
- 5) Разработка электрической принципиальной схемы и схемы электрической подключения;
- 6) Разработка графического приложения диспетчера;
- 7) Оформление расчетно-пояснительной записки;
- 8) Оформление графической части дипломного проекта.

## 2 Проектирование, конструирование и моделирование технических средств

Целью модернизации автоматизированной системы диспетчерского управления подстанции «Левашово» является расширение функциональных возможностей и улучшения эксплуатационных характеристик внедрением управляющих сигналов из диспетчерского пункта. Наиболее рациональным и эффективным способом решения будет масштабирование существующей системы и каналов связи путем написания программного обеспечения для диспетчерского пункта и оснастки подстанции программируемым логическим контроллером, включенным в схему регулятора. Так как используемая SCADA система предоставляет программный интерфейс для написания и интеграции собственного программного обеспечения, то разработку можно осуществлять на любом языке программирования, отдается, однако, предпочтение графическому приложению. Используемая на подстанции система управления автотрансформаторами и контролем переключателей также может быть легко расширена, так как регулятор имеет возможность удаленного управления.

Для обеспечения необходимой функциональности новой системы управления добавим к уже существующим решениям на стороне РДУ и подстанции свои устройства, используя существующие каналы связи.

Однако, поскольку приемо-передающее устройство не имеет конверторов для протокола Modbus, на стороне диспетчерского управления также необходим дополнительный контроллер, реализующий конвертацию протоколов и транслирующий на приемо-передающее устройство готовый сигнал для дальнейшей передачи на объект управления.

Новая система управления позволит диспетчеру, работая в привычном окружении, осуществлять оперативное управление параметрами подстанции, а также обеспечит дополнительный резервный канал связи.

## 2.1 Системный анализ проектируемой системы на основе методов декомпозиции

Потребности и цели создания объекта, структуризация целей и предполагаемых задач, решение которых ведет к достижению этих целей, формирует общее представление об объекте. Анализ объекта на втором этапе осуществляется путем декомпозиции поставленных задач в пространстве, в основном, неметрических структурных характеристик, обусловленных требованиями к объекту. Результатом такого анализа является декомпозиционная схема, при построении которой исходят из следующих двух положений.

Во-первых, декомпозиция осуществляется неформально и на первом уровне разбиения задачи, исходя из назначения и потребности объекта, определяются основные направления, формирующих концепцию его строения в виде множества структурных характеристик  $X_\xi$ , образующих  $n$  блоков 1-го уровня декомпозиции.

Во-вторых, для каждой области (свойства)  $X_\xi$  первого уровня на втором уровне декомпозиции формируются множества альтернатив  $X_\xi^\beta \in X_\xi$ , выбираемым на основе анализа известных решений [4].

Декомпозиционная схема является  $n$  – мерным поисковым пространством  $R^n = \{ X_\xi \}$ , в котором каждое решение представлено вектором  $x \in R^n$ , являющимся множеством из набора альтернатив  $X_\xi^\beta$  вида:

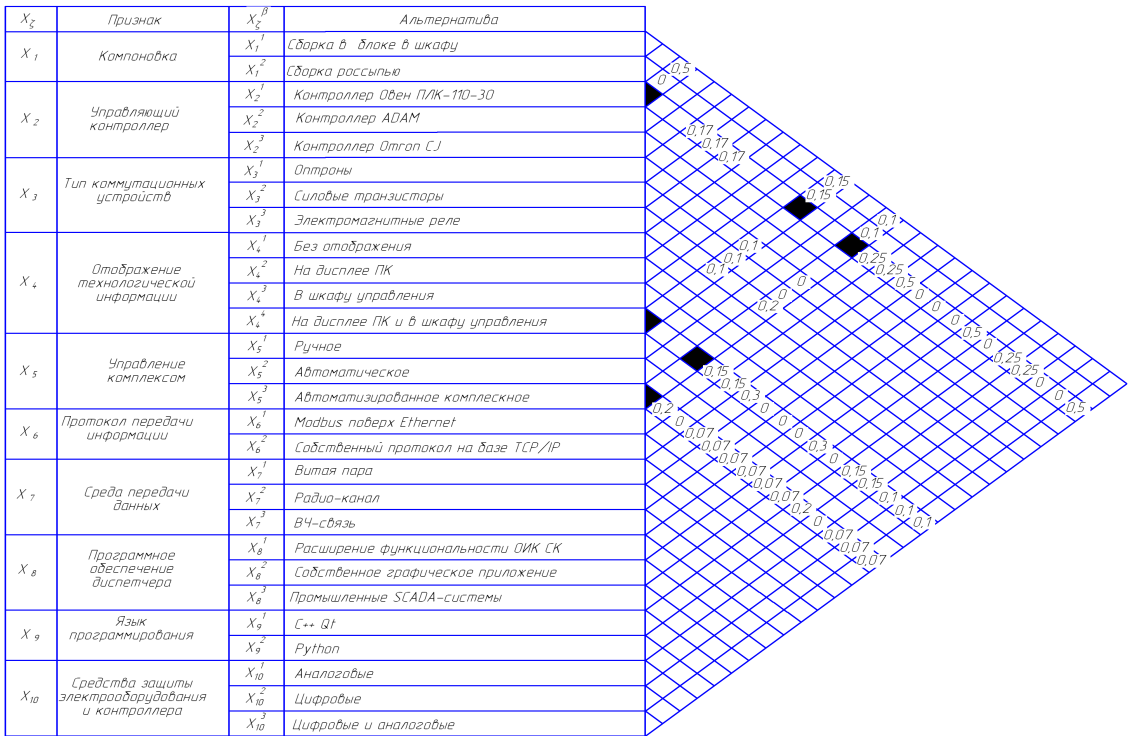
$$X = \{ X_1^\beta, \dots, X_i^\beta, \dots, X_n^\beta \}. \quad (2.1)$$

Геометрической интерпретацией пространства  $R^n$  является его развертка на плоскость, состоящая из матричных блоков, образованных пересечением всех блоков 1-го уровня декомпозиции и альтернатив, образующих клетки матриц:

$$R_B = n ( n-1 ) / 2. \quad (2.2)$$

Развертка  $n$ –мерного поискового пространства для задачи

проектирования системы диспетчерского управления подстанции представлена на рисунке 2.1.



$X_{6j}$	$X_{6j}^{\beta}$	$X_{51} = X_2^1, \lambda_1 = 0,5$		$X_{52} = X_4^1, \lambda_2 = 0,3$		$X_{53} = X_5^3, \lambda_3 = 0,2$	
		Код	Оценка $\lambda_{16j}$	Код	Оценка $\lambda_{26j}$	Код	Оценка $\lambda_{36j}$
$X_{61} = X_1$	$[X_1^1]$	1	0,5	2	0,15	2	0,1
	$X_1^2$	3	0	2	0,15	2	0,1
$X_{62} = X_3$	$X_3^1$	2	0,17	2	0,1	3	0
	$X_3^2$	2	0,17	2	0,1	3	0
	$[X_3^3]$	2	0,17	2	0,1	1	0,2
$X_{63} = X_6$	$[X_6^1]$	2	0,25	2	0,15	1	0,2
	$X_6^2$	2	0,25	2	0,15	3	0
$X_{64} = X_7$	$[X_7^1]$	1	0,5	1	0,3	2	0,07
	$X_7^2$	3	0	3	0	2	0,07
	$X_7^3$	3	0	3	0	2	0,07
$X_{65} = X_8$	$X_8^1$	3	0	3	0	2	0,07
	$[X_8^2]$	1	0,5	1	0,3	2	0,07
	$X_8^3$	3	0	3	0	2	0,07
$X_{66} = X_9$	$[X_9^1]$	2	0,25	2	0,15	1	0,2
	$X_9^2$	2	0,25	2	0,15	3	0
$X_{67} = X_{10}$	$X_{10}^1$	3	0	2	0,1	2	0,07
	$X_{10}^2$	3	0	2	0,1	2	0,07
	$[X_{10}^3]$	1	0,5	2	0,1	2	0,07

Рисунок 2.1 - Развертка n-мерного поискового пространства



Общее количество вариантов структуры N определяется:

$$N = m_1 \cdot m_2 \cdot \dots \cdot m_n. \quad (2.3)$$

Итого общее количество вариантов структуры N определяется:

$$N = 2 \cdot 3 \cdot 3 \cdot 4 \cdot 3 \cdot 2 \cdot 3 \cdot 3 \cdot 2 \cdot 3 = 23328.$$

Таблица 2.1 Декомпозиционная схема анализа задачи формирования структуры системы управления на базе контроллера серии Овен ПЛК-110-30.

$X_\xi$	1 – й уровень декомпозиции	$X_\xi^\beta$	2 – й уровень декомпозиции
$X_1$	Компоновка	$X_1^1$	Сборка в блоке в шкафу
		$X_2^1$	Сборка россыпью
$X_2$	Тип управляющего контроллера	$X_2^1$	Контроллер серии Овен ПЛК-110-30
		$X_2^2$	Контроллер ADAM
		$X_2^3$	Контроллер Omron CJ
$X_3$	Тип коммутационных устройств	$X_3^1$	Оптроны
		$X_3^2$	Силовые транзисторы
		$X_3^3$	Электромагнитные реле
$X_4$	Отображение технологической информации	$X_4^1$	Без отображения
		$X_4^2$	На дисплее ПК
		$X_4^3$	В шкафу управления
		$X_4^4$	На дисплее ПК и в шкафу управления
$X_5$	Управление комплексом	$X_5^1$	Ручное
		$X_5^2$	Автоматическое
		$X_5^3$	Автоматизированное комплексное
$X_6$	Протокол передачи технологической информации	$X_6^1$	Modbus поверх Ethernet

Продолжение таблицы 2.1

		$X_6^2$	Собственный протокол на базе TCP/IP
$X_7$	Среда передачи данных	$X_7^1$	Витая пара
		$X_7^2$	Радио-канал
		$X_7^3$	ВЧ-связь
$X_8$	Программное обеспечение диспетчера	$X_8^1$	Расширение функциональности ОИК СК
		$X_8^2$	Графическое приложение собственной разработки
		$X_8^3$	Промышленные SCADA-системы
$X_9$	Язык программирования	$X_9^1$	C++ Qt
		$X_9^2$	Python
$X_{10}$	Средства защиты электрооборудования и контроллера	$X_{10}^1$	Без средств защиты
		$X_{10}^2$	Средства пожаро- и электробезопасности
		$X_{10}^3$	Средства электробезопасности

Заключительный третий этап выбора и принятия решения после проведения системного анализа создания нового объекта путем её декомпозиции существенно облегчается, но всё же остается непростым из-за большого числа комбинаций  $N$ , являющихся вариантами искомого решения. Задача синтеза состоит в том, чтобы в каждом из  $n$  – блоков первого уровня  $X_\xi$  декомпозиционной схемы выбрать по одной альтернативе  $X_\xi^b$ , подбор которых должен сформировать наилучший вариант.

Простейшим при выборе наиболее эффективного решения является традиционный способ, используемый при обработке морфологических таблиц и предусматривающий последовательное сокращение (свертку) множества вариантов путем отбрасывания наименее перспективных, нереализуемых, наиболее дорогих и других, кажущихся неэффективными, альтернатив. Однако

этот способ при увеличении количества признаков и реализующих их альтернатив соответственно до  $n > 5$ ,  $m > 2$  становится весьма трудоемким, а выбранное решение, в значительной мере, - субъективным. Проблема состоит не только в выборе лучших альтернатив в отдельных блоках, но и в наилучшей их совместимости друг с другом.

Выбор целевых условий:

Далее будет использован метод двухступенчатого ранжирования с присвоением оценок и при необходимости расчетом «весов» для выбираемых элементов, многократно апробированный при создании новых сложных объектов на уровне изобретений.

Предложенный метод предусматривает на первой ступени упорядочения выделение из  $n$  – блоков  $X_\xi$  декомпозиционной схемы  $S$  – блоков, содержащих на уровне альтернатив  $X_\xi^\beta$  наиболее важные характеристики, которые могут быть отражены в задании на создание объекта. Такие блоки  $X_\xi$  обычно в количестве  $S = 2...4$  несут  $S$  – целевых условий (по одному условию для каждого блока), представленных альтернативами, реализующими эти блоки.

Тогда остальные  $g$  – блоков ( $g = n - s$ ) будут содержать на уровне альтернатив локальные решения типа условий – ограничения, а множество, формирующее вариант синтезируемого решения  $X$ , составят два подмножества  $X_S$  и  $X_G$

$$X = \{ X_S, X_G \}, x \in R^n \quad (2.4)$$

Набор условий  $X_S$ , выбираемых на второй ступени упорядочения, определяет некоторую  $S$  – мерную цель синтеза:

$$X_S = \{ X_{Si}^\beta \}, i = 1...S; \beta = 1...m \quad (2.5)$$

Выбранные локальные целевые условия  $X_{Si}^\beta$ , как правило, неодинаково влияют на эффективность синтезируемого решения и поэтому должны быть ранжированы по их значимости и оценены с помощью некоторого параметра

$\lambda_i$ , т.е.:

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_s \quad (2.6)$$

Значения оценочных параметров  $\lambda_i$  устанавливаются с соблюдением дополнительного условия, в качестве которого удобно использовать соотношение:

$$\lambda_1 + \lambda_2 + \dots + \lambda_s = 1 \quad (2.7)$$

Для данного проекта набор целевых условий следующий:

Тип контроллера – Контроллер серии Овен ПЛК:  $X_{S1} = X_2^1$ :  $\lambda_1 = 0,5$ .

Способы отображения информации –

На дисплее ПК и в шкафу:  $X_{S2} = X_4^4$ :  $\lambda_2 = 0,3$ .

Управление комплексом – автоматизированное комплексное (сочетание ручного и автоматического):  $X_{S3} = X_5^3$ :  $\lambda_3 = 0,2$ ;  $0,5 + 0,3 + 0,2 = 1$ .

В итоге трехмерная цепь синтеза:

$$X_S = \{X_2^1, X_4^4, X_5^3\} \quad (2.8)$$

Выбор условий ограничения:

Любой объект, процесс или проект нетехнической сферы деятельности человека характеризуется структурой и параметрами. Начальное представление о строении объекта несет его структура. С её формирования как облика проекта и начинается синтез. Проведя декомпозицию и выбрав в полученном поисковом пространстве  $R^n$  некоторую  $S$  – мерную цель, необходимо решить вторую часть задачи синтеза – сформировать  $g$  – мерное «решение – ограничение».

На основе метода ранжирования определяем условия ограничения:

$$X_G^* = \{ X_{Gj}^\beta \}, j = 1, \dots, g = n - s, \quad (2.9)$$

где  $X_{Gj}^\beta$  - альтернатива, реализующая  $j$ -й блок декомпозиционной схемы и удовлетворяющая всем  $S$  - целевым условиям множества  $X_S = \{ X_{Si}^\beta \}$ .

Если при выборе целевых условий  $X_{Si}^\beta$  можно было использовать исходные данные, то при выборе условий ограничения  $X_{Gj}^\beta$  такие возможности уменьшились.

Оптимизация неформальных решений при нескольких целях традиционными методами неосуществима. В этом случае в качестве оптимизационного подхода целесообразно формировать паретовское множество альтернатив.

Итак, эффективное решение – ограничение  $X_G^*$ , в отличие от  $X_G$  (2.6) определяется паретовским множеством наиболее предпочтительных альтернатив при «мягкой» конкуренции всех ранее выбранных целевых условий  $X_{Gj}^\beta$ . Для решения этой задачи предлагаются следующие шаги:

1. Каждой альтернативе  $X_{Gj}^\beta$  в блоках  $X_{Gj}$  по каждому условию  $X_{Si}^\beta$  присваивается оценка  $\lambda_{iGj}^\beta$ , например, по трехступенчатой шкале в виде кодов (оценок): «лучшая альтернатива (решение)» – код 1, «альтернативы в блоке по отношению к данной локальной цели равнозначные» – код 2, «в блоке уже есть лучшая альтернатива» – код 3.

2. Каждой оценке в соответствие с её кодом присваивается численное значение по правилу:

Код	Значение $\lambda_{iGj}^\beta$
1	$\lambda_{iGj}^\beta = \lambda_i,$
2	$\lambda_{iGj}^\beta = \lambda_i / m,$
3	$\lambda_{iGj}^\beta = 0,$

где  $m_i$  – количество альтернатив  $\lambda_{Gj}^\beta$  в блоке  $\lambda_{iGj}$ .

3. В каждом блоке  $X_{Gj}$  выбираются оценки  $\lambda_{iGj}^\beta$  с наилучшими численными значениями и соответствующие им альтернативы  $X_{Gj}^\beta$  по принципу:

$$\max_{\lambda_1, \dots, \lambda_S} \lambda_{iGj}^\beta \rightarrow X_{G1}^\beta, \dots, \max_{\lambda_1, \dots, \lambda_S} \lambda_{iGg}^\beta \rightarrow X_{Gg}^\beta \quad (2.10)$$

4. Формируется в виде множества  $X^*$  эффективное решение – наилучший вариант.

Таблица 2.2 - Результаты выбора решений-ограничений  $\rightarrow X_{Gj}^\beta$   
целевыми условиями  $X_{Si}^\beta$

$X_{Si}^\beta$	$X_{Gj}^\beta$
$X_2^1$	$X_1^1, X_7^1, X_8^2, X_{10}^3$
$X_4^4$	$X_7^1, X_8^2$
$X_5^2$	$X_3^3, X_6^1, X_9^1$

Для нашего варианта эффективное решение - ограничение:

$$X_G^* = \{X_1^1, X_3^3, X_6^1, X_7^1, X_8^2, X_9^1, X_{10}^3\}$$

Формирование оптимальной по Парето структуры объекта

Согласно этому подходу, альтернатива считается оптимальной по Парето, если всякая другая альтернатива, являющаяся более предпочтительной для одних целей, в то же время будет менее предпочтительна для остальных целей. Принцип оптимальности по Парето далее утверждает, что никогда не следует выбирать альтернативу, которая не является Парето – оптимальной. Лишь при таком (паретовском) выборе можно увеличить степень удовлетворения некоторых целей, не ущемляя при этом других целей. Таким образом, альтернатива паретовского множества обладает тем свойством, что дальнейшее увеличение степени предпочтения для достижения одних целевых условий возможно только за счет других. Проблема состоит в том, чтобы выбрать наилучшие паретовские альтернативы во всех блоках  $g = n - S$ . И эта проблема

всегда достаточно серьезна, однако сам подход ведущий к общему выигрышу, уже является практически важным достижением в решении противоречивых интересов в любой сложной ситуации. К тому же можно утверждать, что оптимальность по Парето является категорией нравственной.

Оптимальный по Парето вариант формируется в виде множества:

$$X^* = \{ X_{Si}, X_{gj}^* \} \quad (2.11)$$

В развернутом виде с указанием всех целевых условий  $X_{Si}^\beta$  и условий – ограничений  $X_{Gj}^\beta$  множество (9) запишется так:

$$X^* = \{ X_{Si}^\beta, \dots, X_{Ss}^\beta, X_{G1}^\beta, \dots, X_{Gg}^\beta \} \quad (2.12)$$

Для данного случая эффективное решение будет иметь вид:

$$X^* = \{ X_2^1, X_4^4, X_5^3, X_1^1, X_3^3, X_6^1, X_7^1, X_8^2, X_9^1, X_{10}^3 \} \quad (2.13)$$

## 2.2 Разработка структурной схемы системы диспетчерского управления

Система диспетчерского управления на аппаратном уровне состоит из следующих компонентов:

- корпус (стойка) с электромонтажными и установочными изделиями;
- сервер ОИК основной;
- сервер ОИК резервный;
- приемо-передающее устройство основное;
- приемо-передающее устройство резервное;
- мультиплексор телемеханики;
- управляемого сетевого коммутатора Cisco;
- рабочие станции обслуживающего персонала и дежурных;
- АРМ диспетчера с проекционным видеощитом;
- кабели соединений.



Рисунок 2.2 – Архитектура ОИП

Типичный двухмашинный комплекс представляет собой группу (программный кластер) из двух серверов, один из которых постоянно работает в режиме горячего резерва. Оба сервера получают данные телемеханики из сервера ТМ либо с помощью механизма онлайн-репликации между основной и резервной БД РВ, но эталонным в каждый момент времени является только один. Типовой конфигурацией является случай, когда данные ТМ после обработки основным сервером в темпе процесса реплицируются в резервный. Возможен вариант связи каждого сервера ОИП с каждым сервером ТМ, что осуществляется за счет коммутаторов локальной сети. В этом случае дополнительные связи используются в целях контроля работоспособности связи, минимизации времени смены ролей серверов и максимальной гибкости реконфигурации. Плановые данные, например, различные виды диспетчерских графиков, прогнозы потребления и т.п., попадая из внешних систем в основной сервер ОИП, автоматически реплицируются в резервный. Это же относится и к ручному вводу данных персоналом.

Передача информации в направлении от ОИП к серверу ТМ возможна только со стороны основного сервера ОИП, который должен быть единственным. Данное требование выполняется программным обеспечением



автоматически.

Каждая пара Сервер ОИП - Сервер ТМ подключается к своему коммутатору или к двум коммутаторам параллельно. Эти коммутаторы, в свою очередь, могут быть связаны между собой высокоскоростной магистралью. Диспетчерские рабочие станции должны в этом случае подключаться также к разным коммутаторам через одну. То же относится и к другим клиентам комплекса. Выход из строя любого элемента описанной системы не приводит к потере наблюдаемости режима энергосистемы диспетчером.

Приемо-передающее устройство «SMART-FEP» предназначено для построения локальных и распределенных систем автоматизации и телемеханизации энергосистем с передачей телемеханической информации оперативно-информационным комплексам (ОИК) диспетчерских пунктов с возможностью ретрансляции телемеханической информации на другие объекты энергосистем. Основным элементом ЦППС «SMART-FEP» является приемо-передающее устройство ППУ-ТМ, которое реализовано как изделие типа Front-End-Processor (FEP), т.е. как коммуникационное устройство, используемое для обработки передаваемой информации, контроля и устранения ошибок, кодирования сообщений. Устройства ППУ-ТМ, входящие в ЦППС, функционально разделены на два комплекта – основной и резервный.

Клиенты подключаются к серверам ОИК и получают необходимую информацию, выполняя запросы в базу данных БД РВ. Права пользователей разделены согласно локальной политики на чтение, запись и изменение записей [5].

Основная обработанная телемеханическая информация, предупреждения и ошибки о выходе за установленные пределы транслируются на проекционный видеосит диспетчера, для чего используется графический пакет обработки и вывода векторной графики Toraz Graphics.

Программное обеспечение проекционного видеосита Varco представляет собой приложение клиент-сервер, обменивающееся информацией по локальной сети с проекционными модулями серии OV-D2.

Проекционные модули серии OV-D2, таким образом, могут управляться и контролироваться любым компьютером в сети, на котором установлено программное обеспечение Barco Wall Control Manager. Если проекционные модули и Barco Wall Control Manager находятся в разных подсетях, эти подсети должны быть соединены с помощью маршрутизатора [6].

Сетевая инфраструктура реализована на базе управляемого коммутатора Cisco серии SF300-48, имеющего 48 Ethernet-портов и обеспечивающего пропускную способность 17.6 Гбит/с.

Также для задачи телеуправления необходим вспомогательный ПК-контроллер Raspberry Pi, служащий для конвертирования протоколов в Modbus. Контроллер подключается напрямую к приемо-передающему устройству, так как в интеграции с серверами ОИК нет необходимости.

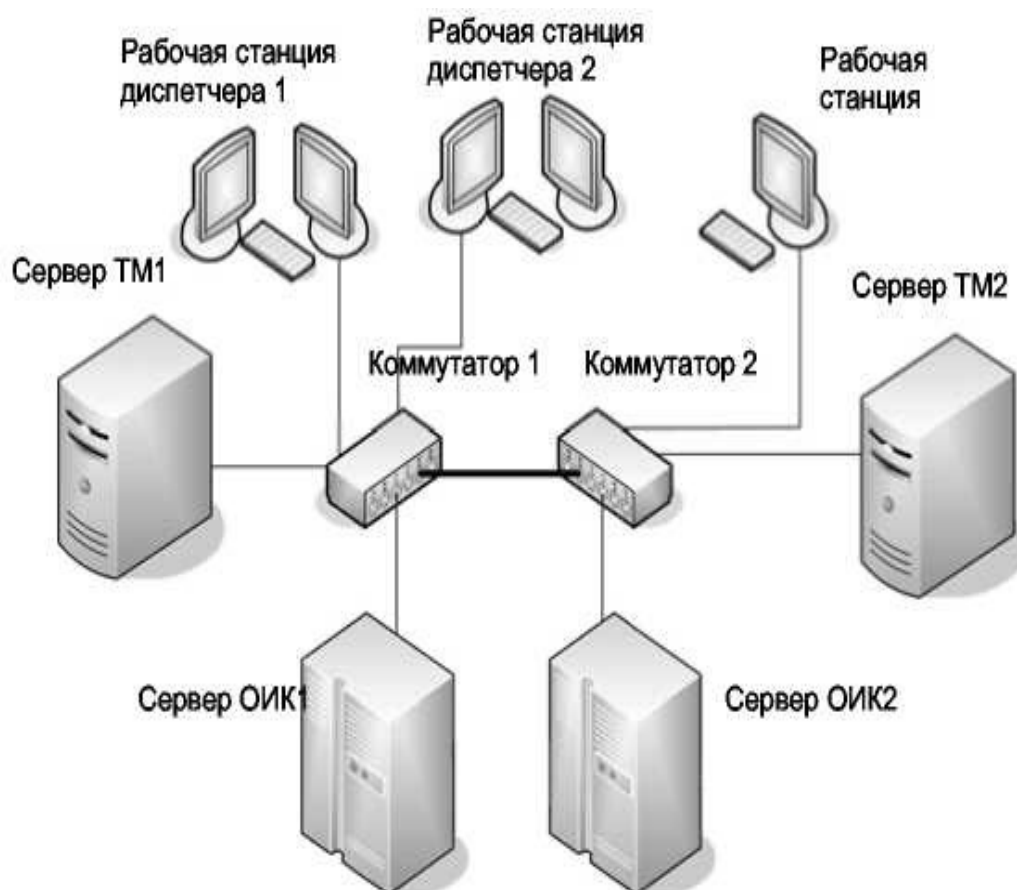


Рисунок 2.3 – Структурная схема двухмашинного комплекса

## 2.3 Разработка структурной схемы системы телемеханизации подстанции

Закрытое пространство подстанции разделено противопожарными перегородками и состоит из:

- помещения трансформаторов собственных нужд, питающих систему связи;
- помещения распределительного устройства закрытого типа;
- помещения релейных панелей;
- помещения связи.

Подключение подстанции «Левашово» в существующую линию связи осуществляется по оптоволоконным линиям. Шкаф связи, обслуживающий оптоволоконные каналы связи. Волоконно-оптическая линия связи предназначена для организации каналов связи по опорам ВЛ от подстанции «Левашово» до подстанций ПС «Заозерная» и ПС «Рябково» и далее по имеющимся каналам связи в помещения связи Курганских электрических сетей, а так же в помещение связи Курганского РДУ. Все оборудование связи состоит из нескольких частей:

- оборудования ВОЛС;
- оборудования мультиплексирования;
- оборудования резервного электропитания.

Оборудование ВОЛС состоит из двух волоконно-оптических кабелей (ВОК) с количеством жил 16, крепежной арматуры для кабеля, материалов для прокладки кабеля по территории подстанций. Все основное оборудование устанавливается на опорах линий электропередач.

Оборудование мультиплексирования состоит из основного мультиплексора уровня STM-1 марки NetRing 600C и первичного мультиплексора марки TC-30, устанавливаемых в шкафу в помещении связи в здании ОПУ на территории ПС «Левашово».

Для подключения к локальной вычислительной сети на ПС «Левашово» предусматривается установка маршрутизатора и коммутатора производства Cisco

Systems.

Основным элементом телемеханизации подстанции является шкаф телемеханики «КП Исеть». Стандартный набор базовых функций – ввод дискретных сигналов (телесигнализация – ТС), ввод аналоговых сигналов (телеизмерение текущее – ТИТ), ввод числоимпульсных сигналов (телеизмерение интегральное – ТИИ), вывод дискретных сигналов (телеуправление – ТУ). Преимущественно по шине RS-485 шкаф телемеханики «КП Исеть» связывает между собой в единую информационную систему шкафы питания цепей оперативной блокировки, шкафы защит трансформаторов, панель учета и преобразователей, а также проектируемый шкаф управления по шине Ethernet.

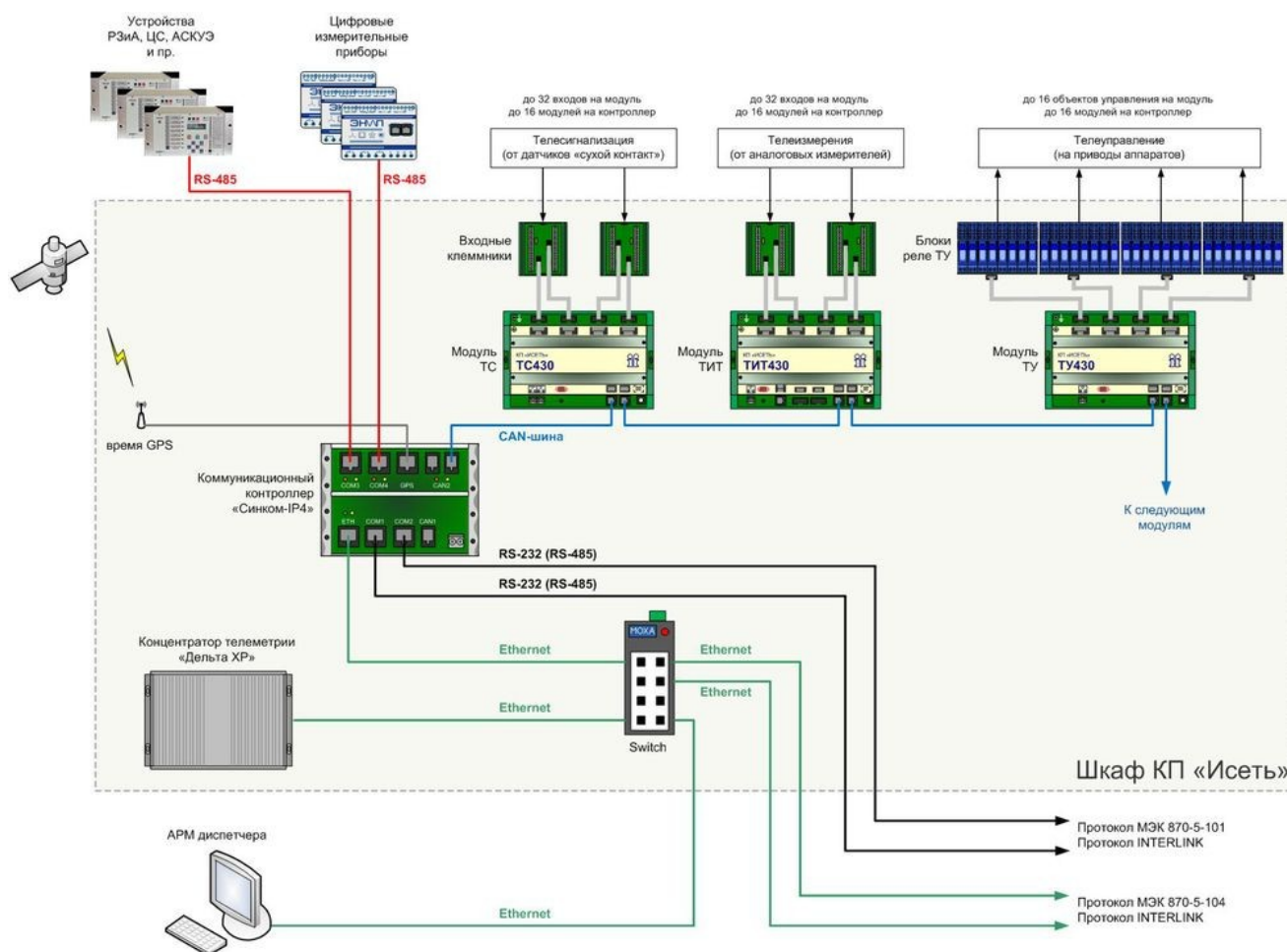


Рисунок 2.4 – Структурная схема телемеханизации на базе «КП Исеть»

Структурная схема телемеханизации подстанции представлена на листе графической части дипломного проекта.

## 2.4 Выбор основных технических средств

### 2.4.1 Требования к техническим средствам

Персональный компьютер АРМ диспетчера удовлетворяет требованиям и для эксплуатации разрабатываемого программного обеспечения управления подстанции «Левашово» в усовершенствовании не нуждается. Для комфортной работы требуется не менее 512 Мбайт оперативной памяти и процессор не хуже Pentium IV, операционная система – семейство ОС Windows или Unix-подобные ОС, в частности дистрибутивы Gnu/Linux с необходимыми для работы программы библиотеками.

Для оперативного персонала и диспетчеров, интенсивно работающих с графическими схемами, желательна установка больших экранов высокого разрешения или многомониторных видеосистем.

Требования к локальной сети могут меняться в зависимости от конфигурации комплекса. Рекомендуется использовать технологию не хуже чем FastEthernet для подключения к сети серверов ОИП и рабочих станций.

Надёжность используемых каналов связи удовлетворяет требованиям, по необходимости следует обеспечить бесперебойное питание для контроллеров и приемо-передающего устройства. Сетевые устройства должны иметь возможность для ретрансляции протокола Ethernet Modbus, использующего технологию TCP/IP. Для подстанции «Левашово» необходимости шифрования канала связи нет, однако, это может быть важным критерием безопасности для управления удаленным оборудованием стратегически важного характера, так как разрабатываемый комплекс позволяет полностью контролировать режим работы объекта управления, что может стать причиной аварии при намеренном неправильном управлении.

Телемеханическое оборудование подстанции должно обеспечивать непрерывную работу, принимать сигналы телеуправления и отправлять в диспетчерский пункт телеметрические данные.

Шкаф телемеханики КП «Исеть» должен иметь свободные порты для

подключения разрабатываемого шкафа, а также необходим в предварительной настройке для трансляции сигналов телеуправления контроллеру Овен ПЛК.

#### 2.4.2 Электрошкаф

Разрабатываемая система управления монтируется в металлическом электрошкафу со степенью защиты от влаги и пыли IP55 и размещается в помещении релейных панелей подстанции.

Выбираем настенный шкаф на базе корпуса серии ШЭН фирмы «ЦМО» с монтажной панелью 450х400 мм, одинарной дверью и замком, он предназначен для установки сетевого и телекоммуникационного оборудования внутри офисных и производственных помещений. Шкаф настенного типа, с возможностью прямого обслуживания, с передней открывающейся дверью, имеющей замок, с верхней вентиляторной панелью, обеспечивающей вытяжную вентиляцию.

Система заземления входит в стандартную комплектацию шкафа. Вертикальные направляющие регулируются по глубине. Возможна установка задней стенки.

В шкафу установлены стандартные DIN-рейки омега-типа (А) ГОСТ Р МЭК 60715-2003 [X] для навесного монтажа электросетевых устройств и контроллера.

Шкаф оснащен каналом ввода для электрокабелей питания и телемеханики.

Технические характеристика шкафа:

- габаритные размеры шкафа (В × Ш × Г)..... 470 × 450 × 180 мм;
- масса .....до 10 кг;
- материал.....листовая сталь;
- исполнение .....передняя дверь;
- количество DIN-реек .....4;

### 2.4.3 Программируемый логический контроллер Овен

Контроллер ПЛК110 предназначен для создания систем автоматизированного управления технологическим оборудованием в энергетике, на транспорте, в т.ч. железнодорожном, в различных областях промышленности.

В задачи программируемого логического контроллера, входящего в состав системы управления параметрами подстанции входит:

- декодирование телесигналов управления, поступающих из диспетчерского пункта в протоколе Ethernet Modbus;
- интеграция с регулятором РНМ и контроль состояния оборудования;
- реализация основного алгоритма управления;
- ведение журнала изменений параметров;
- предотвращение аварийных ситуаций.

Для поставленных задач подходит контроллер фирмы Овен серии ПЛК-110-30, основные технические характеристики которого следующие [7]:

- напряжение питания номинальное ..... 24 В;
- потребляемая мощность не более .....25 Вт;
- количество входов, из них быстродействующих ..... 18 (2);
- подключаемые входные устройства .... коммутационные устройства, датчики, дискретные сигналы 24 В;
- количество релейных входных каналов ..... 12;
- интерфейсы связи ... 2xRS-485, RS-232 Debug, Ethernet 100 Base T, USB;
- центральный процессор ..... RISC (ARM-9), 32 разряда, 180 МГц;
- объем оперативной памяти ... 8 Мб (SDRAM), из них 1 Мб памяти программ, 128 Кб памяти данных;
- габаритные размеры, мм .....140x110x73;



Рисунок 2.5 – Внешний вид контроллера Овен ПЛК-110-30

Данный контроллер поддерживает связь по протоколу Modbus-TCP, дополнительные модули для конвертации протокола не требуется.

Поддерживаются языки программирования семейства МЭК 61131 3, среда для разработки и отладки – Codesys. Для связи контроллера со средой CoDeSys может использоваться один из каналов – RS232 (Debug), Ethernet или USB Device. Система команд контроллера обеспечивает выполнение следующих функциональных операций: логические операции, операции счета времени и числа импульсов, операции с данными, арифметические операции.

ПЛК110 оснащен встроенными часами реального времени, питание которых может осуществляться от автономного источника питания – аккумулятора.

Данный контроллер сочетает в себе необходимый функционал и невысокую стоимость, что делает его идеальным выбором для создания на его базе локальной системы управления напряжением и переключателями подстанции.



#### 2.4.4 Блок питания контроллера Овен

Выбираем блок питания Овен ПЛК из каталога продукции фирмы Овен.

Одноканальный блок питания Овен БП60Б-Д4-24 предназначен для питания стабилизированным напряжением постоянного тока различных радиоэлектронных устройств.

Технические характеристики блока питания БП60Б-Д4-24 [8]:

- номинальное выходное напряжение, В: .....24;
- максимальная погрешность выходного напряжения: .....1%;
- амплитуда пульсации напряжения, мВ: .....120;
- максимальный ток нагрузки, А: .....2.5;
- габаритные размеры (Ш × В × Г), мм: .....72 × 90 × 58.

Блок является импульсным по принципу действия и выполнен по схеме однотактного обратного преобразователя; блок имеет фильтр радиопомех на входе, гальваническую развязку между входом и выходом. Блок защищён от перегрузки, перегрева и короткого замыкания на выходе. На лицевой панели блока расположен световой индикатор наличия выходного напряжения.

Блок изготавливается в пластмассовом корпусе с креплением на DIN-рейку. Корпус состоит из двух частей, соединяемых между собой при помощи защёлки. Для обеспечения отвода тепла, выделяющегося при работе блока, на нижней и верхней гранях корпуса предусмотрены вентиляционные отверстия. Крепление блока на DIN-рейке обеспечивается за счет фиксатора, входящего в комплект поставки.



Рисунок 2.6 – Внешний вид блока питания контроллера

#### 2.4.5 Вспомогательный контроллер Raspberry Pi

Необходимость вспомогательного контроллера возникает по причине отсутствия поддержки протокола Modbus-TCP центральным преемо-передающим устройством SmartFer в составе автоматизированной диспетчерской системы.

На вспомогательный контроллер возлагаются следующие функции:

- обеспечить прием сигналов управления от программы диспетчерского управления;
- обработать команды управления и преобразовать их в протокол Modbus;
- передать преобразованный поток телеуправления для дальнейшей ретрансляции на приемо-передающее устройство через Ethernet.

Задача является достаточно специфической и программируемые логические контроллеры для этой цели не являются достаточно пригодными, тогда как полноценный компьютер является неоптимальным решением.

Критериями выбора послужили: оптимальное соответствие решаемым задачам, наличие мощного процессора, наличие Ethernet интерфейса, небольшие размеры, надёжность, хорошая документированность.

Компьютер распространяется полностью собранным на четырёхслойной печатной плате размером с банковскую карту. Корпус в комплект поставки не входит.

Raspberry Pi выпускается в двух комплектациях: модель «А» и модель «В». Обе версии оснащены ARM11 процессором Broadcom BCM2835 с тактовой частотой 700 МГц и модулем оперативной памяти на 256МБ/512МБ, размещенными по технологии «package-on-package» непосредственно на процессоре. Модель «А» оснащается одним USB 2.0 портом, тогда как модель «В» — двумя. Также у модели «В» присутствует порт Ethernet. Помимо основного ядра, BCM2835 включает в себя графическое ядро с поддержкой OpenGL ES 2.0, аппаратного ускорения и FullHD-видео и DSP-ядро [9].

Тактико-технические характеристики Raspberry pi следующие:

- операционная система .....Debian, Fedora, RISC OS, Android;
- оперативная память.....512 (Model B) Мб;
- постоянная память..... флеш-карта MMC;
- электропитание... .....микро-USB 5 В, от 700 мА;
- потребляемая мощность .....1Вт;
- интерфейсы .... Ethernet,GPIO ,UART, JTAG, SPI, I2C, DSI, CSI, USB;
- вес без корпуса.....45 г;
- размер, мм .....86x53x17;
- тактовая частота.....700 мГц.

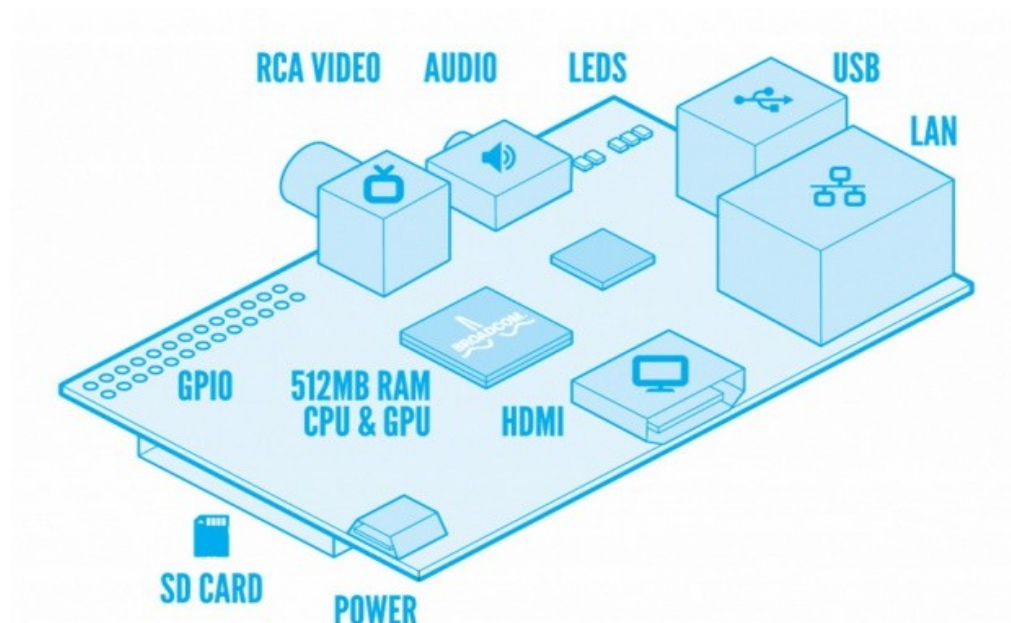


Рисунок 2.7 – Основные узлы контроллера Raspberry Pi

#### 2.4.6. Модуль телеуправления ТУ430 для «КП Исеть»

Модуль ТУ430 предназначен для управления объектом телеметрического контроля в составе шкафа телемеханики КП «Исеть».

Модуль состоит из двух плат: платы КП430 и соединительной платы. Соединительная плата выполнена методом печатного монтажа на пластине с размерами 130 x 195 мм (рисунок 2.8).



Рисунок 2.8 – Внешний вид модуля ТУ430

На плате размещены следующие компоненты:

- разъемы для подключения цепей от контролируемого пункта;
- защитные элементы входных цепей;
- энергонезависимая память;
- разъем питания (XP3);
- разъемы CAN шины (XP4, XP5);

Работой модуля ТУ430 управляет КП430 по программе, хранящейся в его внутренней памяти.

Модуль ТУ430 имеет два режима работы – самоконтроль и выполнение команды телеуправления. В режиме самоконтроля модуль проверяет наличие и уровень основного напряжения питания. Проверка производится через 30 с. После проверки отправляется сообщение УК о соответствии или несоответствии напряжения питания [10].

Режим выполнения команды телеуправления проходит в несколько этапов:

- 1) прием команды от УК КП – «Запрос ТУ»;
- 2) проверка питающего напряжения (допустимое отклонение составляет минус 10 %);
- 3) установка управляющего выхода в активное положение при пониженном напряжении питания (приблизительно 2,4 В);
- 4) измерение коэффициента проводимости обмотки реле;
- 5) проверка измеренного коэффициента на соответствие заданным пределам;
- 6) передача в УК сообщения о готовности или отказе выполнения команды управления;
- 7) прием команды исполнения – «Исполнить ТУ»;
- 8) подача рабочего напряжения на обмотку реле;
- 9) снятие всех управляющих сигналов;
- 10) передача в УК сообщения о завершении исполнения команды.

#### 2.4.7 Регулятор микропроцессорный РНМ-1

Регулятор напряжения микропроцессорный РНМ-1 предназначен для управления электроприводами РПН при автоматическом регулировании коэффициента трансформации силовых трансформаторов. Регулятор применяется на подстанциях с плавно или резко изменяющейся нагрузкой.

Особенность РНМ-1 – это возможность его подключения к двум секциям типовой подстанции, питаемым от разных обмоток одного силового трансформатора. При этом основная секция, в которой регулируется напряжение, является регулируемой, а вторая – контролируемой. Контроль сразу двух секций позволяет при превышении в контролируемой секции измеряемыми величинами значений уставок граничных условий либо только сигнализировать об этом, либо запрещать регулирование (определяется уставкой).

Регулятор обеспечивает следующие функции:

- автоматическое поддержание напряжения в заданных пределах;
- коррекцию уровня регулируемого напряжения по току нагрузки;
- формирование импульсных или непрерывных команд управления электроприводами РПН;
- контроль исправности электроприводов РПН в импульсном режиме работы;
- одновременный контроль двух систем шин;
- оперативное переключение регулирования с одной системы шин на другую;
- блокировку работы и сигнализацию при обнаружении неисправности электропривода РПН;
- блокировку регулирования внешними релейными сигналами;
- блокировку регулирования и сигнализацию при обнаружении перегрузки, превышении  $3U_0$  (или  $U_2$ ) или при пониженном измеряемом напряжении;
- оперативное изменение уставки по напряжению поддержания с одного, заранее выбранного значения, на другое;

Технические характеристики регулятора РНМ-1:

- число входов по току: .....4;
- число входов по напряжению: .....4;
- номинальный входной переменный ток, А: .....5;
- номинальное входное переменное напряжение, В: .....100;
- напряжения зоны нечувствительности ( $\Delta U$ ), % от  $U_n$  .....1-20;
- габаритные размеры ( $Ш \times В \times Г$ ), мм: .....280x260x162;

В состав регулятора входят следующие модули:

- модуль центрального процессора (ПРЦ);
- модуль ввода-вывода (ВВ);
- модуль трансформаторов секции 1 (ТР);
- модуль трансформаторов секции 2 (ТР);
- модуль питания;

– модуль клавиатуры и индикации (КИ).

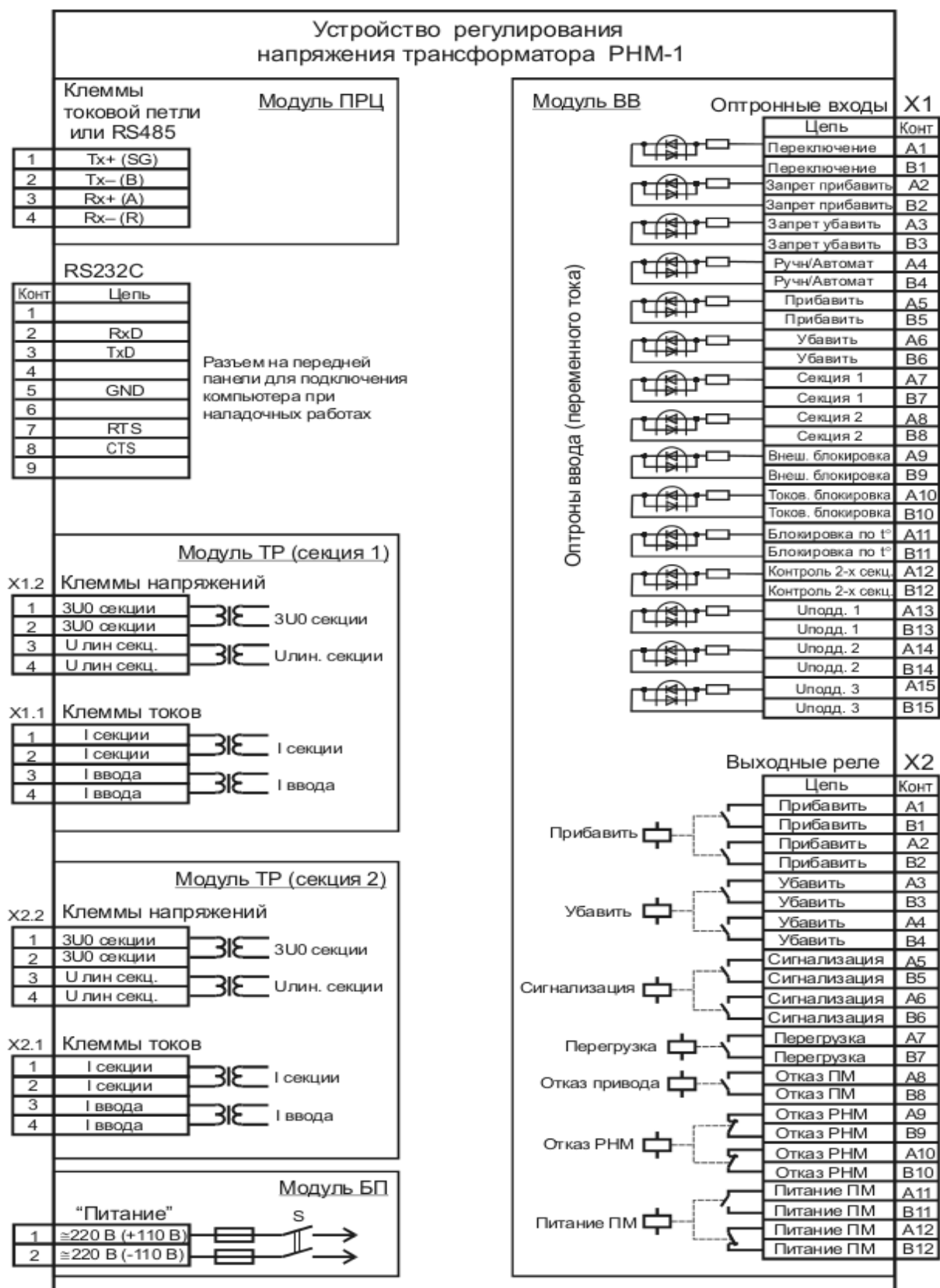


Рисунок 2.9 – Функциональное назначение клемм и контактов разъемов регулятора.

Электрическое соединение модулей между собой осуществляется с помощью ленточных кабелей. Внешние цепи подключаются к регулятору через клеммники, установленные на торцевых поверхностях модулей.

Сигналы со вторичных обмоток измерительных трансформаторов токов и напряжений поступают на согласующие трансформаторы модулей ТР. С модулей ТР аналоговые сигналы поступают на модуль ПРЦ, где фильтруются и с помощью АЦП преобразуются в цифровую форму. Микропроцессор TN80C196KC20 обрабатывает поступающие сигналы и выполняет действия в соответствии с алгоритмом, записанным в ПЗУ. Значения уставок, номер регулируемой секции, номер ступени переключения и значение счетчика ресурса хранятся в энергонезависимой перепрограммируемой памяти и сохраняются при отключении питания [11].

Функциональное назначение дискретных входов:

- «Переключение» – сигнал, поступающий от привода переключателя. Наличие сигнала свидетельствует о том, что идет процесс переключения;
- «Запрет прибавить» (SQn) – сигнал, поступающий от верхнего концевого переключателя. Наличие сигнала свидетельствует о том, что переключатель находится в верхнем положении, регулирование в сторону увеличения напряжения невозможно;
- «Запрет убавить» (SQ1) – сигнал, поступающий от нижнего концевого переключателя. Наличие сигнала свидетельствует о том, что переключатель находится в нижнем положении, регулирование в сторону уменьшения напряжения невозможно;
- «Ручное / Автоматическое». При отсутствии сигнала управление приводом осуществляется от внешних кнопок или удаленно; при наличии – регулирование автоматическое, кнопки внешней регулировки игнорируются;
- «Прибавить» – сигнал, определяющий состояние внешней кнопки «Прибавить»;
- «Убавить» – сигнал, определяющий состояние внешней кнопки «Убавить»;



– «Секция 1», «Секция 2» – сигналы, определяющие, какая секция выбирается в качестве регулируемой;

– «Вн. блокировка», «Ток. блокировка», «Блокировка по  $t^{\circ}$ » – сигналы внешней блокировки. Наличие сигнала на любом из входов запрещает регулирование. Сигнал «Блокировка по  $t^{\circ}$ » подается от датчика температуры масла привода.

– «Контроль двух секций» – сигнал, определяющий количество контролируемых секций. При наличии данного сигнала контролируются обе секции. Одна из секций определяется как регулируемая, вторая – как контролируемая.

## 2.5 Разработка шкафа управления

Шкаф управления включает в себя следующее оборудование:

- 1) регулятор РНМ-1;
- 2) контроллер Овен ПЛК-110-30;
- 3) блок питания контроллера Овен БП60Б;
- 4) трехфазный автоматический выключатель S203P;
- 5) электромагнитные реле R4-2014-23-5230-WTL 8 штук;
- 6) пакетные переключатели kk0-20;
- 7) клеммы Weidmuller WTL, WTR4, WDU4;
- 8) монтажные панельки для реле GZT4 8 штук;
- 9) соединительные мостики.

Шкаф подключается в существующую систему управления и выполняет следующие функции:

- обмен данными с пунктом диспетчерского управления;
- обработка телеметрической информации;
- хранение информации;
- система отображения;
- контроль уровня напряжений;
- управление технологическим оборудованием (приводом РПН);
- контроль работы противоаварийной автоматики.

Используемое оборудование имеет возможность монтажа на DIN-рейке,

необходимости в печатных платах и пайке не возникает, что обеспечивает быстрое монтирование, лёгкое обслуживание и быструю замену выведенных из строя элементов.

Шкаф питается от трехфазной сети трансформаторов собственных нужд.

Общий вид шкафа управления представлен на листе графической части дипломного проекта.

## 2.6. Разработка электрической схемы подключения программируемого контроллера

Шкаф управления представляет собой совокупность стандартных устройств: программируемый логический контроллер, регулятор, автоматические выключатели, коммутационные устройства, соединенных между собой соответствующим образом в единую структуру.

Каждый из устройств соединяется между собой согласно схеме соединения проводниками, собранными для удобства в системы шин.

Трехфазное электропитание от трансформаторов собственных нужд ~220В 50Гц подводится одним 5-х жильным кабелем (фазы А, В, С, ноль, защитная земля) 3х(1,5...2,5) кв.мм к группе клемм ввода питания сверху шкафа.

В качестве устройств токовой защиты от перегрузки и короткого замыкания используются автоматический выключатель SF1, устройство защиты от перенапряжения Mains Plugtrab (A2), сетевой фильтр Wavefilter 3A (A3) как показано на листе «Программируемый контроллер. Схема электрическая подключения» в графической части проекта.

Так как при включении защищаемых автоматом устройств, величина пускового тока по отношению к номинальному току относительно невелика, то можно выбрать автомат с время-токовой характеристикой отключения типа. Для защиты шкафа управления выбираем 3-х полюсный автомат S203P C25 фирмы АВВ.

Устройство по категории монтажа (категории перенапряжения) по ГОСТ Р 51350-99 относится ко II категории и обеспечивает устойчивость электрической изоляции цепей питания к максимальному импульсному напряжению 250 В.

По степени защиты человека от поражения током в соответствии с ГОСТ 12.2.007.0-75 (Приложение Н ГОСТ Р 51350-99) Устройство относится к классу «I», т.е. обеспечена электрически непрерывная цепь соединения токопроводящих частей Устройства с защитным заземлением (заземляющей шиной).

Эксплуатация устройства без соединения заземляющей клеммы шкафа с заземляющей жилой питающего кабеля или заземляющего болта корпуса шкафа с контуром защитного заземления здания запрещается.

Все электрические цепи и составные элементы устройства имеют маркировку согласно схемы соединений.

## 2.7 Моделирование потребления энергии с помощью программы «Консоль ПП»

С программно-аппаратной платформой автоматизированного диспетчерского управления производится дополнительная интеграция программных продуктов «Консоль ПП» и «Космос», предназначенных для оперативных расчетов, анализа ситуации в единой энергосистеме, моделирования краткосрочного и долгосрочного перетоков энергии и прогнозирования прогноза потребления. Это позволит диспетчеру правильно реагировать на сложившуюся ситуацию, вести заданный режим работы и избегать аварий.

Задача формирования расчетной схемы является вспомогательной и решается перед оцениванием состояния, расчетом установившегося режима или оптимизацией. Цель ее решения заключается в подготовке схемы для выполнения расчетов по перечисленным программам с учетом реального положения выключателей и разъединителей на станциях и подстанциях энергосистемы. Очевидно, что без такой задачи при оперативном использовании программного комплекса невозможно решать, например, вопросы, связанные с разделением шин, односторонним отключением линий, выделением блоков для работы на линии [12].

Для формирования расчетной схемы необходимо, чтобы были подготовлены описания схем первичной коммутации всех или части объектов. С целью подготовки таких описаний разработан специальный графический

редактор. Изображение, сформированное с помощью этого редактора, является не только графическим образом, но и определяет способ соединения отдельных элементов, составляющих коммутационную схему. В соответствии с набором установленных на схему элементов формируются таблицы, позволяющие задавать параметры генераторов, нагрузок, трансформаторов и реакторов.

В момент ввода телеметрической информации корректируется состояние выключателей коммутационных схем. Уточнить при необходимости состояние отдельных выключателей и разъединителей можно вручную, воздействуя на изображения соответствующих элементов на схеме.

Целью решения задачи оптимизации режимов по активной мощности является максимизация функции эффективности работы рынка:

$$F = \sum_{i=1}^n C_{i_i} \cdot P_{i_i} - \sum_{j=1}^k \tilde{N}_{\tilde{A}_j} \cdot P_{\tilde{A}_j}, \quad (2.14)$$

где  $n$  - общее число потребителей, подающих ценовые заявки на энергорынок;  
 $C_{i_i}$  - максимальная цена, которую согласен платить  $i$ -ый потребитель за поставляемую электроэнергию;  
 $P_{i_i}$  - мощность  $i$ -го потребителя;  
 $k$  - общее число генераторов, предлагающих электроэнергию для реализации на энергорынке;  
 $C_{\tilde{A}_j}$  - минимальная цена, по которой согласен отпускать электроэнергию  $j$ -ый генератор;  
 $P_{\tilde{A}_j}$  - максимальный объем электроэнергии, которую предлагает для реализации на рынке  $j$ -ый генератор.

Оптимизация режимов по реактивной мощности. Решение задачи позволяет определить загрузку источников реактивной мощности и номера анцапф трансформаторов с регулированием под нагрузкой, при которых обеспечивается минимум потерь активной мощности. В качестве целевой функции задачи используется следующая:

$$F(X) = \sum_{i=1}^n I_i^2(X) \cdot R_i + \sum_{j=1}^k U_j^2 \cdot Y_j, \quad (2.15)$$

где  $k$  - число узлов в схеме замещения;  $n$  - число ветвей в схеме замещения;  
 $X$  - вектор независимых параметров оптимизации;  $I_i(X)$  - величина тока в

$i$ -й ветви;  $R_i$  - активное сопротивление  $i$ -й ветви;  $U_j$  - величина напряжения в  $j$ -м узле;  $Y_j$  - активная проводимость шунта в  $j$ -м узле.

Ограничения составляют уравнения балансов активных и реактивных мощностей в узлах схемы замещения и диапазоны изменения независимых переменных. И целевая функция, и ограничения являются нелинейными и минимизация потерь выполняется итерационным методом, на каждом шаге которого решается задача квадратичного программирования.

В графической панели программы «Консоль ПП» располагаются кривые заявленного, фактического, расчетного потребления.

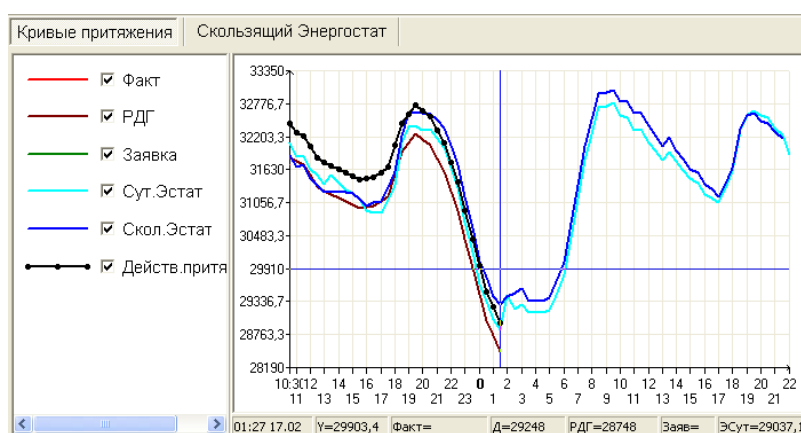


Рисунок 2.10 – Кривые потребления энергии в программе «Консоль ПП»

На странице «скользящий энергостат» размещается дополнительная информация для анализа кривых Энергостат-БР, рассчитанных в разное время для одного района, просмотр которой даёт представление о данных, используемых при прогнозе.



Рисунок 2.11 – Анализ кривых потребления энергии

Программные пакеты интегрируются с основным сервером ОИК-СК 2007 и позволяют производить оперативную оценку сложившейся ситуации. Информация моделирования энергорежимов и прогноза потребления позволит диспетчеру более эффективно управлять энергосистемой и избегать аварийных режимов работы, увеличив степень автоматизации диспетчерского управления.

## 2.8. Разработка схемы электрической принципиальной шкафа управления

Схема электрическая принципиальная разработана на основе типовой схемы подключения регулятора РНМ-1, приведенной в документации на изделие. Регулятор РНМ-1 имеет несколько групп оптронных входов и выходов для пары контролируемых секций подстанции, что позволяет строить на его базе различные варианты систем управления.

Регулятор поддерживает несколько режимов управления, в том числе автоматический, ручной и режим удаленного управления. Используя удаленный режим управления, мы получаем возможность непосредственно контролировать технологическое оборудование, при этом регулятор берет на себя корректную обработку команд в соответствии с алгоритмом.

В процессе работы измеряются токи и напряжения, контролируется состояние дискретных входов (клеммник X1 модуля А3). На индикаторе в верхней строке отображается текущее значение напряжения в регулируемой секции и номер ступени переключения. После каждого переключения изменяется номер ступени переключения и увеличивается значение счетчика выработанного ресурса В нижней строке отображается расчетное значение напряжения поддержания Упр. Светодиод «Переключение» горит при приходе сигнала от ПМ на дискретный вход «Переключение».

Горящий светодиод «Секция 1» или «Секция 2» определяет, какая из секций является регулируемой. Для выбора регулируемой секции необходимо подать сигнал на соответствующий дискретный вход, т.е. при подаче сигнала

на дискретный вход «Секция 1» регулирование ведется по первой секции, при подаче сигнала на вход «Секция 2» – по второй секции.

О полной или частичной блокировке работы регулятора сообщает горящий светодиод «Блокировка» и включенная подсветка индикатора, в нижней строке которого выводится сообщение о причине блокировки.

К блокировкам от внешних дискретных сигналов, подаваемых на клеммник X1 модуля АЗ, относятся:

- «Вн. блокировка» – наличие сигнала на клеммах А9/В9;
- «Ток. блокировка» – наличие сигнала на клеммах А10/А10. На данный дискретный вход подается сигнал с токовой защиты входной цепи силового трансформатора;
- «Блокировка по  $t^\circ$ » – наличие сигнала на клеммах А11/А11. На данный дискретный вход подается сигнал от датчика температуры масла ПМ.

В устройстве предусмотрена блокировка регулирования и сигнализация в случае превышения измеряемыми величинами значений уставок граничных условий. При превышении максимального значения тока ввода запрещается выдача команды «Прибавить», а через 10 с включаются реле «Перегрузка» и «Сигнализация». При превышении напряжением  $3U_0$  значения уставки запрещается выдача команды «Прибавить». При снижении в течение 10 с напряжения ниже минимального значения запрещается регулирование и включается реле «Сигнализация». При превышении максимального значения напряжения запрещается команда «Прибавить», включается реле «Сигнализация» и запускается программа ускоренной отработки перенапряжения, т.е. команда «Убавить» будет выдаваться после завершения очередного цикла переключения через задаваемую задержку, пока напряжение не снизится до напряжения поддержания.

Автоматическое регулирование полностью блокируется. Регулятор реагирует только на кнопку «Выход» и кнопку «Выбор». При снятии всех сигналов внешней блокировки автоматическое регулирование возобновляется, гаснет светодиод «Блокировка», гаснет подсветка индикатора.

К блокировкам от привода относятся:

- «Запрет убавить» – наличие сигнала на клеммах А3/В3 запрещает регулирование в сторону снижения напряжения;
- «Запрет прибавить» – наличие сигнала на клеммах А2/В2 запрещает регулирование в сторону повышения напряжения.

В проекте используется существующая схема подключения регулятора к приводу, не нуждающаяся в существенной модернизации (см. рис 2.12).

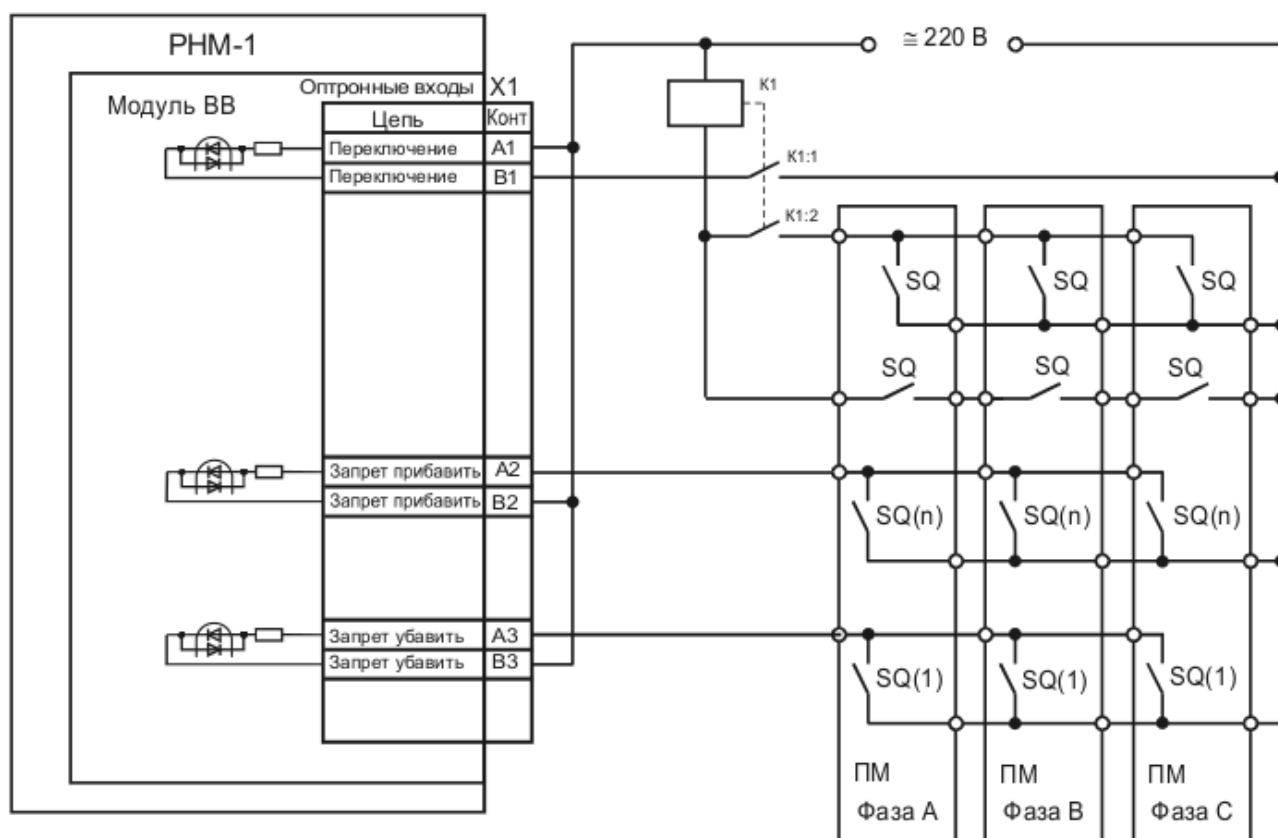


Рисунок 2.12 – Схема электрическая подключения привода

На рисунке 2.12 изображены:

- ПМ – привод механический;
- РПН-1 – устройство регулирования напряжения трансформатора;
- К – реле;
- SQ – контакты привода, сигнализирующие о процессе переключения;
- SQ(1) – контакты концевого выключателя нижнего положения переключателя;
- SQ(n) – контакты концевого выключателя верхнего положения



переключателя.

Контроллер Овен ПЛК-110-30 имеет 12 релейных входов 24 В, что обеспечивает его интеграцию в схему без применения специальных устройств согласования уровня. Контроллер подключен на соответствующие входы регулятора согласно схеме, позволяя переключать тип управления, контролируемую секцию, дискретно прибавлять и убавлять напряжение. Дополнительно на контроллер заводятся сигналы о запрете изменения напряжения и токовой блокировке.

В качестве коммутационных устройств выбраны электромагнитные реле с четырьмя переключающими контактами. Максимальное напряжение контактов AC/DC 250 В, минимальное коммутируемое напряжение 5 В. Нарботка на отказ для реле типа R4-2014-23-5230-WTL составляет 1200 циклов/час. Для быстрой замены предусмотрены колодки.

Пакетные переключатели позволяют производить коммутацию на месте, в случае неисправности удаленного управления. Выбраны ручные переключатели Ganz серии kk0-20.

К регулятору подключаются измерительные преобразователи, нуждающиеся в стабильном питании, особенно в защите от импульсных перенапряжений.

Защиту по питающей сети обеспечивают:

- автоматический выключатель S203P;
- защитное устройство MainsPlugtrab;
- сетевой фильтр Wavefilter 3A.

Схема электрическая принципиальная системы управления представлена на листе графической части дипломного проекта.

### 3 Информационное и программное обеспечение системы управления

#### 3.1 Информационная структура, ведомости входных и выходных сигналов

##### 3.1.1 Информационная структура

Серверное ПО комплекса можно разделить на две части: системное и прикладное.

В состав системного ПО входят операционная система Microsoft Windows Server 2003 R2 Enterprise Edition, система управления базами данных MS SQL Server 2000 SP3, DNS-серверы (для преобразования символических имен в IP-адреса), контроллеры MS Active Directory, драйверы для доступа к реляционной базе данных. MS SQL Server предназначен для хранения описания сетевой конфигурации комплекса, нормативно-справочной информации (НСИ), программного обеспечения, долговременных архивов. Контроллеры Active Directory служат для обеспечения централизованного управления правами пользователей и однократной процедуры аутентификации при входе в сеть.

Серверная часть программного обеспечения комплекса состоит из следующих крупных компонент:

- инсталляция сервера;
- общие библиотеки;
- сервис ОИП — супервизор и планировщик заданий, осуществляющий управление задачами ОИП, а также динамическую реконфигурацию сети ОИП;
- база данных реального времени и сопутствующие программы, включая задачу архивирования, которая отвечает за перенос заданной части архивов из БД РВ в долговременный архив, хранящийся в таблицах реляционной базы данных;
- программы обработки информации, к которым относятся программа обработки ТИ/ТС, программы расчетов по заданным формулам, программа расчета агрегированных величин (сумм, интегралов,

средних, максимумов, минимумов и т.д.) по заданным при настройке правилам, задачи формирования различных суточных ведомостей и сводок, программы информационной поддержки работы с диспетчерским графиком;

- набор программ связи с внешними системами: серверами ТМ, другими ОИП, загрузки данных из файлов и т.п.;
- Технологические задачи контроля перетоков в опасных сечениях, контроля токовой нагрузки и уровней напряжения.

Схема серверного ПО для двухмашинного комплекса приведена на рисунке 3.1

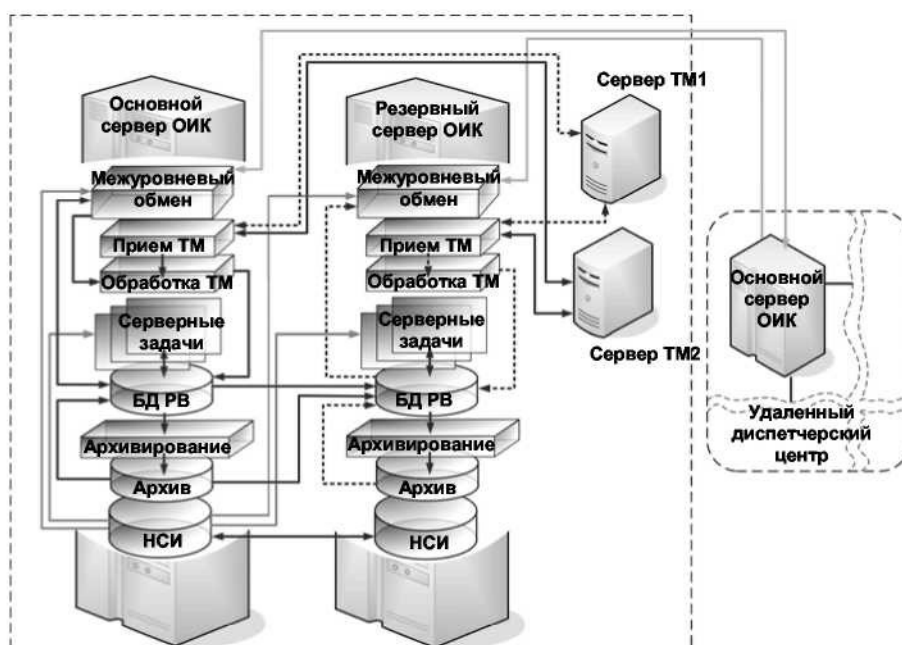


Рисунок 3.1 – Схема серверного ПО

На каждом сервере должны быть установлены все программные компоненты ОИП, включая СУБД MS SQL Server, сервис ОИП, БД РВ, обмен с сервером ТМ и архивирование. Эти компоненты работают параллельно на всех серверах домена ОИП. Задачи обработки выполняются только на одном из серверов в каждой группе, который в данный момент времени является основным. На резервном сервере задачи обработки должны быть отключены (эта настройка осуществляется автоматически в процессе инсталляции комплекса). Все данные,

поступающие по каналам телемеханики, команды диспетчера, ручной ввод, планы, прогнозы и так далее направляются только в основной сервер. После их обработки данные направляются в БД РВ, которая рассылает их клиентам. Особую роль играет резервная БД РВ. Она подписывается у основной базы на все изменения и получает их. Таким образом, осуществляется репликация в темпе поступления данных на основной сервер. Процессы архивирования протекают параллельно на каждом из серверов.

Лицом комплекса, с которым чаще всего сталкивается любой пользователь, является система отображения, состоящая из Монитора отображения, gol-файлов (динамических библиотек графической системы Graphic Object Library), дополнительных специализированных библиотек, исполняемого модуля векторной графики Toraz, набора форм, разработанных для каждого центра управления.

Программное обеспечение комплекса сопряжено с программами моделирования режима и прогноза потребления «Консоль ПП» и «Космос 2003». Имеется возможность как циклических расчетов с получением вектора оцененных параметров режима практически в реальном времени (на реальных схемах объемом 400-600 узлов успешно проводились расчеты с циклом в 10 секунд), так и экспресс-расчетов в интерактивном режиме по заданию диспетчера, с последующим проигрыванием ситуации по включению/отключению оборудования и анализом ситуации.

### 3.1.2 Ведомости входных и выходных сигналов

Телеизмерение – это измерительная информация от измерительных преобразователей (датчиков), которая передается на пункт управления или контроля непрерывно или циклически, а иногда по вызову — после послышки оператором специального сигнала-запроса, содержащего адрес (кодированное обозначение) измеряемого параметра. В программно-аппаратном комплексе

контроля параметров энергосистемы телеизмерения несут в себе информацию о значениях напряжения, тока, активной и реактивной мощности, частоты и других параметров энергосистемы, требующих контроля.

Телесигнал – это сигнал системы сигнализации на расстоянии, осуществляемой средствами телемеханики. Сигнал несет в себе дискретную информацию о состоянии контролируемого объекта. В программно-аппаратном комплексе контроля параметров энергосистемы эта информация о состоянии масляных выключателей, положении ремонтных переключателей на линиях и т.п.

Обработанная порция телеметрической информации и рассчитанных параметров передается в базу данных реального времени. Каждый параметр состоит из уникального номера параметра (идентификатора), значения в формате с плавающей точкой (для подавляющего большинства типов оперативной информации), метки времени и слова признаков, состоящего из 32 битов. Для телесигналов используется специальный формат — целое 2-байтовое число для значения и 16-битное слово признаков [13].

При обнаружении недостоверной информации в аргументах функции результат дорасчета помечается, как недостоверный по параметрам функции. Если в процесс расчета формулы какую-то операцию не удастся корректно выполнить (происходит деление на 0, извлечение квадратного корня из отрицательного числа, возведение числа 200 в 3456-ю степень и т.п.), то результату присваивается старое значение с признаком «сбой расчета».

Идентификация параметров телеметрической, отчетной, плановой и иной информации ограничена числом 4 194 303. Это не означает, что ОИП сможет обработать 4 миллиона параметров каждой категории, однако, поддержка «больших» идентификаторов обеспечена с целью предоставления максимальной гибкости в настройках комплекса.

ГОСТ Р МЭК 870-5-104 обеспечивает профили связи для передачи основных телемеханических сообщений между центральной телемеханической станцией (ПУ - пункт управления) и телемеханическими

подстанциями (КП - контролируемые пункты), которые используют постоянные некоммутируемые каналы связи между центральной станцией и отдельными подстанциями.

Для реализации функции телеуправления вводятся следующие входные и выходные дискретные сигналы, обозначенные в таблицах 3.1 и 3.2.

Таблица 3.1 – Ведомость выходных дискретных сигналов

№ Сигнала	Наименование	Тип сигнала	Направление сигнала
1	2	3	4
1	Прибавить напряжение	ТУ	На подстанцию
2	Убавить напряжение	ТУ	На подстанцию
3	Автоматику РПН ввести	ТУ	На подстанцию
4	Автоматику РПН вывести	ТУ	На подстанцию
5	Активна секция I	ТУ	На подстанцию
6	Активна секция II	ТУ	На подстанцию

Таблица 3.2 – Ведомость входных дискретных сигналов

№ Сигнала	Наименование	Тип сигнала	Направление сигнала
1	2	3	4
1	Блокировка управления	ТС	В РДУ
2	Запрет прибавить	ТС	В РДУ
3	Запрет убавить	ТС	В РДУ
4	Успех последней команды	ТС	В РДУ
5	Ошибка последней команды	ТС	В РДУ

### 3.2 Разработка алгоритмов диспетчерского управления

#### 3.2.1 Разработка алгоритма программы диспетчерского управления

Алгоритм программы диспетчерского управления разработан с целью

предоставить возможность управления удаленными объектами энергетики по каналам связи.

Программа предоставляет графический интерфейс, реализуя концепцию событийно-ориентированного программирования. Событие в объектно-ориентированном программировании (ООП) — это сообщение, которое возникает в различных точках исполняемого кода при выполнении определённых условий. Данные сообщения направляются обработчикам (слушателям), что позволяет своевременно реагировать на изменившееся состояние системы. Событийно-ориентированное программирование — парадигма программирования, заключающаяся в том, что выполнение программы определяется событиями — действиями пользователя (клавиатура, мышь), сообщениями других программ и потоков, событиями операционной системы.

Основная ветвь алгоритма в бесконечном цикле ждет событий, проверяет пользовательский ввод, запуская соответствующие подпрограммы — обработчики.

В алгоритме реализована связь с объектом управления по протоколу Modbus TCP (по каналам Ethernet технологии TCP/IP). Передача команд переключения режима управления, изменения контролируемой секции, прибавления и убавления напряжения.

В управлении важными объектами, такими как подстанция, важно предусмотреть обработку ошибок. Алгоритмом предусмотрена обработка следующих исключительных ситуаций:

- ввод некорректных данных;
- ввод заведомо больших значений;
- ввод неправильного пароля;
- отсутствие канала связи с управляемым объектом;
- завершение работы программы.

Прием изменений происходит, если форма ввода заполнена надлежащим образом, данные корректны, введен верный пароль. При наличии ошибок

диспетчер будет проинформирован во всплывающем окне с текстом ошибки. В течение 5 секунд с момента приема изменений имеется возможность отказаться от передачи, выбрав во всплывающем окне передачи сигнала отказ.

Также алгоритмом обеспечено журнализация всех действий, выполняемых диспетчером.

### 3.2.2 Разработка алгоритма программируемого логического контроллера

Алгоритм для Овен ПЛК реализует необходимые для управления функции, а именно:

- получает и обрабатывает сигналы телеуправления;
- реализует сигналы в физические команды регулятору согласно алгоритму;
- контролирует текущее состояние;
- предотвращает работу в аварийных режимах;
- обеспечивает автономную работу системы в отсутствии связи.

С соответствующих дискретных выходов регулятора считывается информация о текущем состоянии – доступность управления и режим управления. Если управление доступно, происходит обработка команды, в противном случае возвращается код ошибки.

Если управление доступно, запрета на изменение напряжения нет, то есть не достигнуты пределы изменения напряжения, выставляется логическая единица на соответствующий порт (X3:A8, X3:A9), запускается таймер T0 на время 20 мс, в течение которого уровень удерживается, что позволяет регулятору точно принять сигнал и произвести соответствующее изменение напряжения. Аналогично выполняются команды переключения типа управления и изменения управляемой секции.

Алгоритм имеет обработку следующих исключаяющих ситуаций:

- отсутствие связи с диспетчерским пунктом;
- проверка опасности принятия текущей команды;



– поступление серии опасных или не имеющих точного смысла команд.

При успешном выполнении текущей команды производится передача сигнала об успешном приеме изменения, в противном случае генерируется сигнал ошибки.

Блок-схемы алгоритмов работы программ диспетчерского управления представлены на двух листах графической части дипломного проекта.

### 3.3 Разработка экранных форм диспетчера

#### 3.3.1 Разработка мнемосхемы подстанции «Левашово» для системы Toraz

В СК-2007 для работы с графическими схемами используется система векторной графики Toraz Graphic. За счёт векторного представления элементов схем и наличия слоев обеспечивается возможность при изменении масштаба динамически изменять представление, прореживать мелкие элементы, не слишком важные на том или ином уровне увеличения, создавать схемы для работы с большими проекционными экранами.

Система Toraz обеспечивает создание интерактивных мнемосхем из типовых элементов с необходимым уровнем детализации. Выводимые величины связываются с соответствующими телесигналами из ОИК СК-2007 и в режиме реального времени выводятся на проекционные видеоэкраны диспетчера. На схеме отображаются переток электроэнергии, напряжение, ток и частота. При выходе из заданных пределов одной из величин отображаются предупреждения, привлекающие внимание диспетчера и требующие незамедлительного решения.

В проекте разрабатывается мнемосхема подстанции «Левашово», отличительной особенностью которой является элемент – кнопка «управление», по нажатию на которую запускается разрабатываемое графическое приложение по управлению подстанцией, позволяющее оперативно изменить параметры энергообъекта.

### 3.3.2 Разработка графического интерфейса программы диспетчерского управления подстанции

Графический интерфейс программы диспетчерского управления разрабатывается с учетом современных тенденций в разработке графических настольных приложений. Для графического интерфейса используется современный фреймворк QT5, представляющий набор библиотек для создания графического приложения любой сложности.

Основное окно программы содержит стандартные элементы, необходимые элементы отображения и ввода информации, позволяющие диспетчеру производить корректировку режима работы:

- переключать тип управления подстанции;
- переключать управляемую секцию подстанции;
- плавно изменять напряжение автотрансформаторов;
- производить планирование переключений.

Логически пространство главного окна разделено на две части. Слева отображается информация, характеризующая текущий режим, слева расположены элементы изменения и ввода новых значений. Это позволяет диспетчеру быстрее ориентироваться и производить необходимую корректировку.

В нижней части окна имеется область отображения состояния и предупреждений о наличии связи и возможности управления.

Из меню основного окна программы доступны окна настройки параметров связи, окно просмотра журнала, окно справки.

При разработке графического интерфейса были учтены эргономические факторы, что позволило создать максимально удобный и понятный интерфейс, а также избежать неоднозначности интерпретации информации. Наиболее важные элементы интерфейса выделены другим цветом и контрастируют на фоне. Все наиболее важные действия требуют от диспетчера подтверждения, что исключает случайные изменения, способные привести к аварийным

последствиям.

Графические формы программ диспетчерского управления представлены на листах графической части дипломного проекта.

### 3.4 Разработка программы диспетчерского управления

#### 3.4.1 Программный интерфейс расширения функциональности ОИК СК-2007

При проектировании внешних программ (тех программ, которые производят обмен данными с ОИК СК-2007) необходимо учитывать правила поведения категорий оперативной информации.

Каждая программа, обменивающаяся данными с ОИК СК-2007, должна выполнять следующие действия:

- подключиться к нужному сервису ОИК;
- открыть соединение с БДРВ;
- подписаться на события об изменении НСИ и, при получении события, обновлять НСИ в памяти программы;
- послать запрос в БДРВ (синхронный или асинхронный).

Клиентская программа, запускаемая как форма программы Монитор отображения, должна быть реализована в виде динамической библиотеки (Dll), в которой должна быть описана стандартная экспортная функция открытия форм ОИК СК-2007 – NewForm2. Внешняя программа должна быть опубликована в РБД, в таблице StorageDll.

Связь с панелью управления осуществляется посредством обмена сообщений. Монитор и форма обмениваются стандартными для СК-2007 сообщениями.

После подключения внешнего модуля запуск его будет идентичен открытию формы отображения ОИК СК-2007. Для того чтобы в списке появилась форма, отвечающая за запуск внешнего модуля, необходимо описать новую форму в РБД. Для этого следует добавить запись в таблицу Form. Для добавляемой формы в поле DLL необходимо прямо указать название внешнего

модуля.

После вышеперечисленных действий форма внешний модуль будет подключён и появится в дереве форм Монитора отображения, в описанной для него группе форм. Для использования внешнего модуля необходимо его открыть, как обычную форму отображения СК-2007.

Монитор отображения ОИК СК-2007 (СКMonitor.exe) осуществляет взаимодействие с различными модулями (такими, как: Graphic Object Library (GOL), DLL, Topaz Modules и др.), управляющими отображением данных на формах СК-2007 для:

- передачи данных формам СК-2007;
- управления соединениями с БДРВ и РБД;
- управления навигацией между формами;
- управления формами.

Взаимодействие между панелью управления и формами СК-2007 осуществляется посредством обмена следующими сообщениями:

- $\text{MMI\_FreshControl} = \text{MMI\_BASE} + 2$  – перерисовывает панель управления формы;
- $\text{MMI\_StartRT} = \text{MMI\_BASE} + 6$  – сообщение форме – работа в режиме слежения;
- $\text{MMI\_SyncRT} = \text{MMI\_BASE} + 8$  – сообщение форме – просмотр архивных данных. LPARAM – время просмотра формы в формате UnixTime;
- $\text{MMI\_NavForm} = \text{MMI\_BASE} + 20$  – сообщение от формы – открытие формы через механизм навигации WPARAM – идентификатор, LPARAM – идентификатор формы;
- $\text{MMI\_Appl} = \text{MMI\_BASE} + 27$  – сообщение от формы – запрос Handle объекта Application;
- $\text{MMI\_RBDSName} = \text{MMI\_BASE} + 28$  – сообщение от формы – запрос имени рабочего сервера РБД, результат – указатель на строку, содержащую имя рабочего сервера РБД;
- $\text{MMI\_RBD} = \text{MMI\_BASE} + 29$  – сообщение от формы – запрос имени

рабочей РБД, результат – указатель на строку, содержащую имя рабочей РБД;

– MMI\_ReconRBD=MMI\_BASE+30 – сообщение форме – необходимость пересоединиться с РБД. WPARAM – указатель на строку с именем сервера;

– MMI\_ReconRTDB=MMI\_BASE+56 – сообщение форме – произошло переключение к другой БДРВ;

– MMI\_NoConnectView=MMI\_BASE+60 – сообщение форме – монитор работает в режиме «нет данных»;

Подсистема событий является одним из центральных звеньев архитектуры ОИК и к любым изменениям в её структуре следует подходить с предельной осторожностью. Базовой конфигурации событий вполне достаточно для подавляющего большинства центров управления.

Заведение нового события подразумевает наличие приложения, которое будет его генерировать. Следовательно добавление пользовательских событий может осуществляться в двух случаях:

– заведённое событие будет использоваться системой напоминаний;

– событие предполагается использовать в приложении собственной разработки, предназначенном для работы в среде ОИК СК-2007.

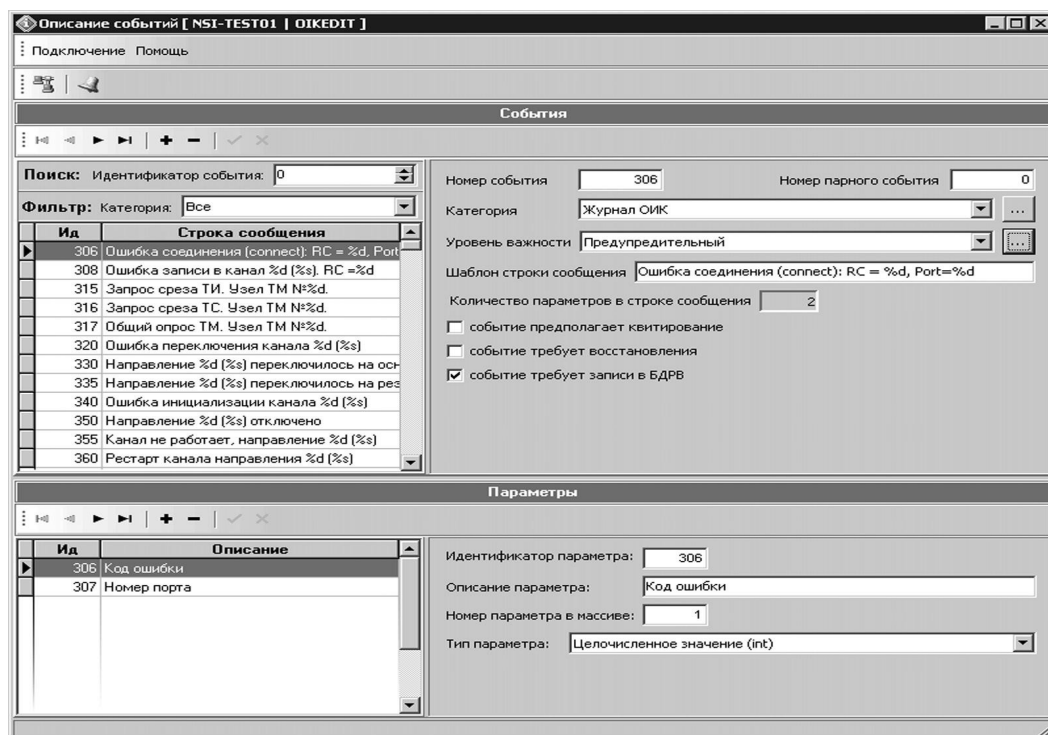


Рисунок 3.2 – Главное окно описания событий

Для включения серверной задачи в ОИК необходимо использовать программу Управление ОИК. Каждую программу необходимо описать для всех серверов домена, на которых планируется её запуск. Каждый пакет программ необходимо описать для всех серверов группы, на которых планируется его запуск. Управление запуском пакета в зависимости от роли сервера осуществляется в диалоговых окнах, которые появляются после выбора пункта главного меню .

Существует возможность передавать значения параметров ОИ в управляющие параметры графических блоков Тораз. Для этого в данных экземпляра блока для каждого такого параметра должен быть описан источник данных.

Для выбора параметра ОИ, значение которого будет использоваться при работе блока, необходимо чтобы редактор Тораз имел активное соединение с РБД ОИК.

### 3.4.2 Разработка программы контроля параметров подстанции ДиспетчерПро

Разработка программы ведется на языке программирования C++ в среде разработки QT Creator, для построения графического интерфейса используются компоненты QT. Выбор фреймворка QT позволил создать приложение, отвечающее современным требованиям программного обеспечения. Язык программирования C++ стал де-факто стандартом при разработке больших приложений, внедряемых на производстве, это обусловлено наличием обширной стандартной библиотеки функций и классов, поддержкой парадигмы программирования ООП, существования множества компиляторов для сборки программ под различные платформы, в том числе для встраиваемых (embedded) решений [14]. Отличительной особенностью C++ от своего предшественника языка C является наличие в языке поддержки исключений, что позволяет ловить ошибки времени исполнения и обрабатывать их в соответствии с алгоритмом, не

останавливая исполнение программы. Выбор также был обусловлен наличием программного интерфейса расширения функциональности СК-2007, позволяющим писать и подключать модули на языке C++.

При разработке программы использовался прием разделения кода. Логически программа разделена на несколько файлов исходного кода и заголовочных файлов.

Основным файлом является `main.cpp`, содержащий в себе точку входа в программу – функцию `main`. В этом коде создается инстанс приложения и запускается обработка событий в бесконечном цикле.

Код, описывающий основной алгоритм работы, находится в файле `mainwindow.cpp`. В нем определены обработка событий, генерируемых при пользовательском вводе, в терминологии QT обработчики сигналов называются слотами. При нажатии любой из кнопок графического интерфейса генерируется событие-сигнал, программно связывающийся со слотом, выполняющим код обработки.

Файлы `logreader.ui`, `logreader.h` и `logreader.cpp` описывают второе окно программы контроля параметрами подстанции – журнал. Данный код позволяет вести журнал событий, записывая в текстовый файл все изменения, производимые диспетчером, а также ошибки связи. Окно журнала позволяет просматривать архив записей, фильтровать и очищать. Непосредственно файл размещается на диске в текстовом виде и доступен для просмотра любым другим текстовым редактором.

Программа имеет файл настроек `settings.conf`, код, ответственный за чтение и запись которых находится в файлах `settings.h` и `settings.cpp`. В текстовом файле хранятся параметры:

- сетевой адрес и название подстанции, с которой осуществляется связь;
- хеш-функция SHA1 от пароля, необходимого для приема изменений;
- путь до файла журнала;
- текущие тип управления и секция, сохраняющиеся между запусками.

Параметры настроек записываются в файл с новой строки, формат записи

– параметр=значение.

Файлы `parameters.h` и `parameters.cpp` описывают класс `Parameters`, инкапсулирующий в себя логику проверки изменений, в нем реализован код, позволяющий контролировать и корректировать пользовательский ввод. Предоставлены методы для получения и установки текущих напряжений, режима управления.

Файлы `networking.h` и `networking.cpp` содержат код, описывающий сетевое взаимодействие с объектом управления по протоколу МЭК 870-5-104.

При телеуправлении (ТУ) для обеспечения достоверности отраслевой протокол МЭК 870-5-10 предусматривает использование только двухэтапной процедуры ТУ. КП должен отказываться от выполнения команды, если не было предварительной. Механизм отказа в этом и других случаях: причина отраженной передачи – завершение активации, в причине передачи бит  $N/P=1$ ,  $FC=1$ .

При выполнении телеуправления вопросы протокольной процедуры тесно связаны со схемными особенностями КП:

- наличие автодиагностики выходных цепей ТУ,
- тип контакта, являющегося датчиком сигнала ТС.

Если автодиагностика есть, ее следует начинать после получения первой посылки (предварительный выбор). К моменту получения второй (исполнение) автодиагностика будет закончена (при скорости до 2400 Бод), то есть не надо будет корректировать значение таймаута, рассчитанное без учета автодиагностики при ТУ.

Отрицательная квитанция на первую посылку означает, что для данного объекта информации (выключателя) ТУ невозможно – запрещено либо ранее была выявлена неисправность. Отрицательная квитанция на вторую посылку посылается при неуспешной автодиагностике.

Приводимая ниже процедура соответствует стандарту МЭК 870-5-104 и предусматривает передачу с КП посылки с  $COT=10$  – «Завершение активации», которая, по-видимому, должна была соответствовать моменту отключения



выходных цепей ТУ и прекращения выходного импульса. Однако в России весьма распространены такие схемы вторичной коммутации, в которых выходной импульс ТУ воздействует не только на привод выключателя, но и на то реле, которое является датчиком ТС. При этом сигнал как бы об исполнении команды может появиться существенно раньше фактического переключения выключателя и даже при отказе привода. Поэтому универсальная процедура (технологическая, а не только протокольная), которая всегда даст правильный результат, должна быть следующей:

- при успешной автодиагностике и получении исполнительной команды замыкается выходная цепь, на ПУ отсылается «подтверждение активации»;
- выдержка времени для работы привода;
- отключение выходной цепи;
- выдержка времени 20 мс, чтобы датчик ТС пришел в истинное состояние;
- ввод ТС;
- передача состояния выключателя, на который подавалось управление;
- передача «завершения активации».

В производственной сфере существует необходимость в переносимости исходных кодов между процессорными архитектурами, в т.ч. 32-битными и 64-битными платформами. Для этого в проекте используется скрипт для сборки Makefile. Правила сборки должны поддерживать условное подключение различных версий библиотек. В некоторых случаях требуется подключение разных библиотек под разные условия компиляции. Программа может собираться различными компиляторами, в частности GCC, Intel C/C++, MSVC. Проекты должны собираться под Windows и Linux. В условиях вычислительного кластера ПО должно собираться из командной строки без привлечения среды разработки, через терминальное соединение. Проект должен собираться на персональном компьютере с использованием привычной среды разработки проектов. Задача утилиты make - автоматически определять, какие файлы проекта были изменены и требуют компиляции, и применять необходимые для

этого команды. Для использования утилиты необходимо описать сценарии в файле Makefile. Сценарии указывают взаимосвязь между файлами проекта и определяют необходимые действия для обновления каждого файла проекта.

Программа разработана и оттестирована в среде операционной системы Linux, исполняемый 64-х битный файл занимает 203 Кб на диске, программа использует 36.1 Мб оперативной памяти и требует для комфортной работы процессор не ниже уровня Pentium IV. В зависимости от сборки, может иметь тип исполняемого файла PE (Windows OS) или ELF (Unix-подобные ОС), а также 32-х или 64-х исполняемый машинный код, что существенно на системные требования не влияет.

Рассмотрим логику работы программы при запуске более подробно, это поможет лучше понять алгоритм и находить возможные ошибки при эксплуатации.

```
void MainWindow::showEvent(QShowEvent *event) {  
    // функция вызывается каждый раз при появлении главного окна,  
    // переопределение стандартной функции showEvent необходимо для реализации  
    // логики инициализации  
    if( event->spontaneous())  
        return; // возврат, если окно уже было отрисовано ранее  
    if(isInitialized) {  
        return; // возврат, если программа уже была инициализирована  
    }  
    param.init_parameters_from_file(); // читаем параметры конфигурации из  
    // файла settings.conf, файл должен существовать, в противном случае возникнет  
    // ошибка  
    QString strnum;  
    _refresh_lcd(ui->lcdNumber_2, strnum.sprintf("%d", param.get_U()));  
    _refresh_control(); // отображаем в окне текущее напряжение  
    if (param.get_control() == "auto") { // если режим управления Авто, то  
        // отключаем элементы управления (кнопки прибавить, убавить)
```

```

        ui->radioButton->setDisabled(true);
        ui->radioButton_2->setDisabled(true);
        ui->pushButton->setDisabled(true);
        ui->pushButton_2->setDisabled(true);
    }
    ui->lineEdit_3->setText(strnum.sprintf("%d", param.current_section()));
    qInstallMessageHandler(SimpleLoggingHandler); // устанавливаем свой
    обработчик механизма отладки для журналирования событий
    qDebug() << param.get_logfile() << "|запуск приложения";
    QList<QString> *eobject = param.get_eobject(); // получаем из файла
    settings.conf сетевой IP адрес подстанции для подключения
    if (tryConnect((*eobject)[0], (*eobject)[1])) {
        on_action_triggered(); // если удалось подключиться, вызываем
        обработчик успешного подключения, зажигаем иконку соединения в области
        уведомлений
    } else {
        on_action_7_triggered(); // если подключиться не удалось, вызываем
        обработчик ошибки, записываем в журнал, зажигаем иконку отсутствия
        соединения в области уведомлений
    }
    delete eobject; // во избежание утечки памяти удаляем объекты с
    динамически выделенной памятью
    ui->label_block->setStyleSheet("background-image:url(/new/prefix1/green_lock.png); background-repeat: no-repeat;");
    isInitialized = true; // устанавливаем флаг успешной инициализации
    приложения
}

```

Исходный код программы составляет более 1500 строк на C++, основные компоненты программы приведены в приложении А.

## 4 Эксплуатационная документация

### 4.1 Инструкция по эксплуатации

#### 4.1.1 Инструкция по установке программы контроля параметров ДиспетчерПро

Оперативно-информационная подсистема (ОИП) – это программно-аппаратный комплекс, предназначенный для надежного получения данных о текущем режиме энергетической системы (единой, объединенной), высокопроизводительной обработки поступающей информации и выдачи оперативному персоналу всех изменений режима, состояния оборудования и аварийно-предупредительных сообщений в темпе поступления информации.

ОИП включает в себя функции, обеспечивающие безопасное проведение ремонтно-восстановительных работ в энергосистеме, поддержание баланса мощности и ведение согласованного режима.

ОИП обеспечивает архивирование заданного набора оперативной информации, включая данные о режиме энергосистемы, произошедших событиях, действиях операторов, диспетчеров и других пользователей на указанную глубину.

ОИП предоставляет пользователям удобный и единообразный графический интерфейс.

Конфигурация рабочих мест оперативного дежурного персонала.

Рабочие места оперативного дежурного персонала подключены к разным сетевым устройствам, что обеспечивает возможность контроля работоспособности программно-аппаратного комплекса и выполнения функциональных обязанностей при выходе из строя одного из них.

Программа контроля параметров подстанции интегрируется на АРМ диспетчера, подключаясь к серверу ОИК СК-2007. Перед использованием необходимо описать события к окну описания событий и привязать к ним соответствующие телемеханические сигналы. Убедиться, что программа имеет доступ до базы данных и может выполнять sql-запросы.

Установка программы происходит простой распаковкой файлов программы в заданную директорию. Имеется возможность собрать программу контроля параметров подстанции из исходных кодов или использовать предкомпилированный исполняемый файл. Защитить файл программы политикой безопасности Active Directory, выдав соответствующим пользователям права на запуск. Отредактировать конфигурационный файл, вписав основные параметры в секцию General, отделив переводом строки, согласно примеру:

[General]

logfile=C:\PathTologFile.txt

objectaddress=192.168.5.31

objectname=Левашово

password=4511F8C0E213C8EC51559EB4A1FFCCBC7E42710E

Где password – хеш-функция SHA1 от настоящего пароля. Так, например, если мы хотим использовать пароль «секрет», то посчитать хеш-сумму для него можно следующим образом, используя командную оболочку shell: `echo "секрет" | gpg --print-md sha1`. Будет получен ответ «6852FE59829316D4C02F15A551DBEBC6B50B4A35», являющийся хеш-суммой от пароля, который необходимо записать в конфигурационный файл. Во время исполнения при вводе пароля программа вновь посчитает хеш-функцию и сравнит с тем, что записан в файле. Использование хеш-сумм вместо хранения пароля в открытом виде позволяет избежать компрометации пароля простым просмотром файла конфигурации, так как математически нет возможности вычислить пароль по образу его хеш-суммы.

После запуска программы убедиться в отсутствии ошибок подключения к сетевым ресурсам, провести диагностику можно, используя файл журнала.

Формат ведения журнала следующий: дата: тип события: сообщение. Где тип события может быть Error, Warning, Event. Простые события, связанные с изменением параметров режима считаются за Event. Warning обозначаются события соединения с подстанцией и появление возможности управления. Error

содержит ошибки каналов связи, внутренние ошибки приложения. При диагностике проблем внимание в первую очередь необходимо обратить на Error, при использовании просмотрщика журнала из приложения имеется возможность фильтрации записей по типу и дате, также типы сообщений подсвечиваются разным цветом.

Для примера ниже представлен отрывок журнала:

Sun May 11 00:50:42 2014: Event: запуск приложения

Sun May 11 00:50:42 2014: Warning: соединение установлено

Sun May 11 00:50:52 2014: Error: ввод неправильного пароля

Sun May 11 00:55:59 2014: Error: управление заблокировано

Sun May 11 00:56:04 2014: Warning: управление доступно

Sun May 11 00:56:10 2014: Error: нет соединения, работа в автономном режиме

Sun May 11 00:56:12 2014: Warning: соединение установлено

Sun May 11 00:56:59 2014: Event: напряжение изменено на: -48

Sun May 11 00:56:59 2014: Event: текущее напряжение: 9952

Sun May 11 00:56:59 2014: Event: управление изменено на: "manual"

Sun May 11 00:56:59 2014: Event: активная секция изменена на: 2

Sun May 11 00:57:08 2014: Event: выход из программы

Дополнительная документация на программное обеспечение не требуется, так как графический интерфейс позволяет интуитивно разобраться с предоставленным функционалом.

#### 4.1.2 Инструкция по подключению микропроцессорного регулятора РНМ-1

При работе с микропроцессорным регулятором РНМ-1 необходимо соблюдать все требования безопасности, распространяющиеся на устройства релейной защиты и автоматики энергосистем. К эксплуатации регулятора допускаются лица, изучившие настоящее руководство по эксплуатации и

прошедшие проверку знаний техники безопасности и эксплуатации электроустановок электрических станций и подстанций.

Перед подключением регулятора к сети он должен быть подсоединен к контуру заземления медным проводом сечением не менее 2 кв. мм.

При вводе регулятора в эксплуатацию необходимо произвести: внешний осмотр, измерение электрического сопротивления изоляции, проверку работоспособности. При проведении внешнего осмотра произвести контроль на отсутствие вмятин на корпусе, трещин и сколов на экране дисплея, целостности клеммников. Измерение электрического сопротивления изоляции проводить между закороченными цепями входных токов и напряжений (клеммники X1 и X2 модулей ТР), выходных контактов реле и входов оптронов (клеммники X1 и X2 модуля ВВ), питания регулятора (клеммник «Питание» модуля БП), а также между этими цепями и корпусом.

Перейти в состояние контроля «Дискретных входов». Поочередно подавая на дискретные входы клеммника X1 модуля ВВ переменное или постоянное напряжение 220 (110) В убедиться, что изменяется состояние соответствующей позиции индикатора [11].

#### 4.1.3 Инструкция по программированию и настройке контроллера Овен ПЛК

Перед использованием контроллер ПЛК110 необходимо запрограммировать, т.е. создать пользовательскую программу, реализующую алгоритм, приведенный в пункте 3.2.1 и на листе графической части. После создания пользовательская программа может быть сохранена в энергонезависимой Flash-памяти контроллера и запускаться на выполнение после включения питания или перезагрузки.

Программирование осуществляется с помощью ПО CoDeSys 2.3. Для связи со средой программирования CoDeSys может использоваться один из интерфейсов контроллера: Debug RS-232, USB-Device или Ethernet.

Время реакции контроллера зависит от типов применяемых входов и выходов. Аналитическое выражение для определения времени полного отклика контроллера ( $T_{откл}$ ) имеет вид :

$$T_{откл} = T_{вх} + 2T_{цикла} + T_{вых}, \quad (4.1)$$

где  $T_{вх}$  – время реакции входа на изменение физического сигнала (включая фильтрацию). По умолчанию значения для обычных входов – 1,0 мс, для быстродействующих входов – 0,1 мс,  $T_{цикла}$  – время цикла ПЛК. Установленное значение по умолчанию – 1 мс (стабилизированное). Настраивается в окне «Конфигурация ПЛК (PLC Configuration)» ПО CoDeSys,  $T_{вых}$  – задержка на срабатывание выхода. Значения для релейных выходов – 20 мс; для обычных транзисторных выходов – 10 мс; для быстродействующих транзисторных выходов – 0,02 мс [7].

Для работы ПЛК по протоколу Modbus TCP, необходимо [15]:

- 1) создать новый проект в CoDeSys, указав соответствующий target;
- 2) на вкладке «Ресурсы» («Resources»), выбрать «Конфигурация ПЛК» («PLC\_Configuration»), и добавьте модуль Modbus(slave) к базовой конфигурации (Рис.4.1);

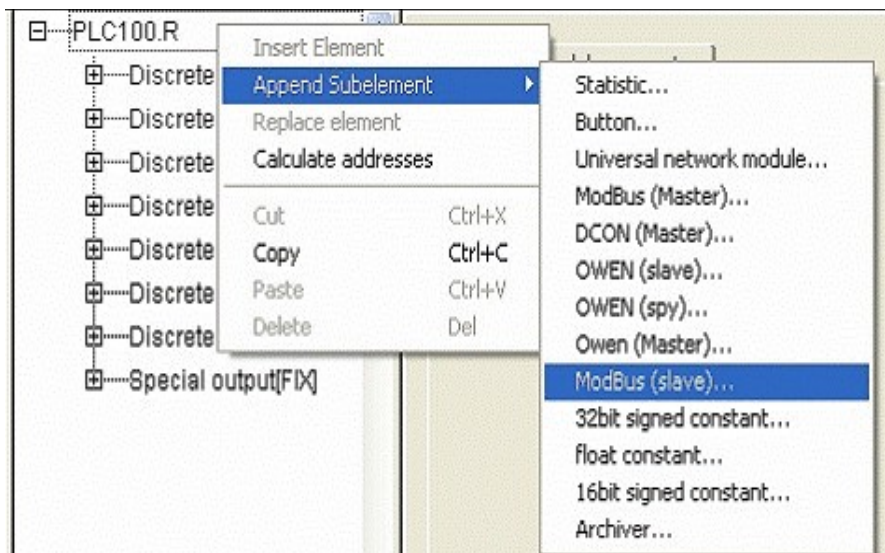


Рисунок 4.1 – Настройка ПЛК Овен на работу с протоколом Modbus TCP

- 3) В качестве интерфейса модуля добавить интерфейс «TCP». Задать имя переменным для обращения к ним из программы ПЛК (Рис 4.2);



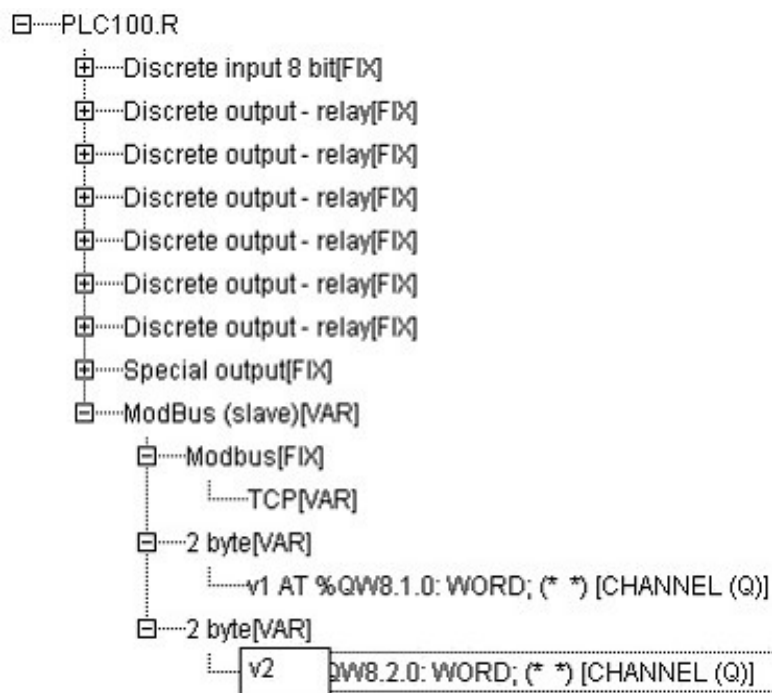


Рисунок 4.2 – Объявление переменных и связывание с интерфейсом

4) определить IP-адрес ПЛК 1 для использования в программе контроля параметров ДиспетчерПро, для этого зайти на вкладку «Ресурсы» и выбрать «PLC\_Browser», затем в открывшемся окне ввести команду «PLCInfo» (Рис.4.3).

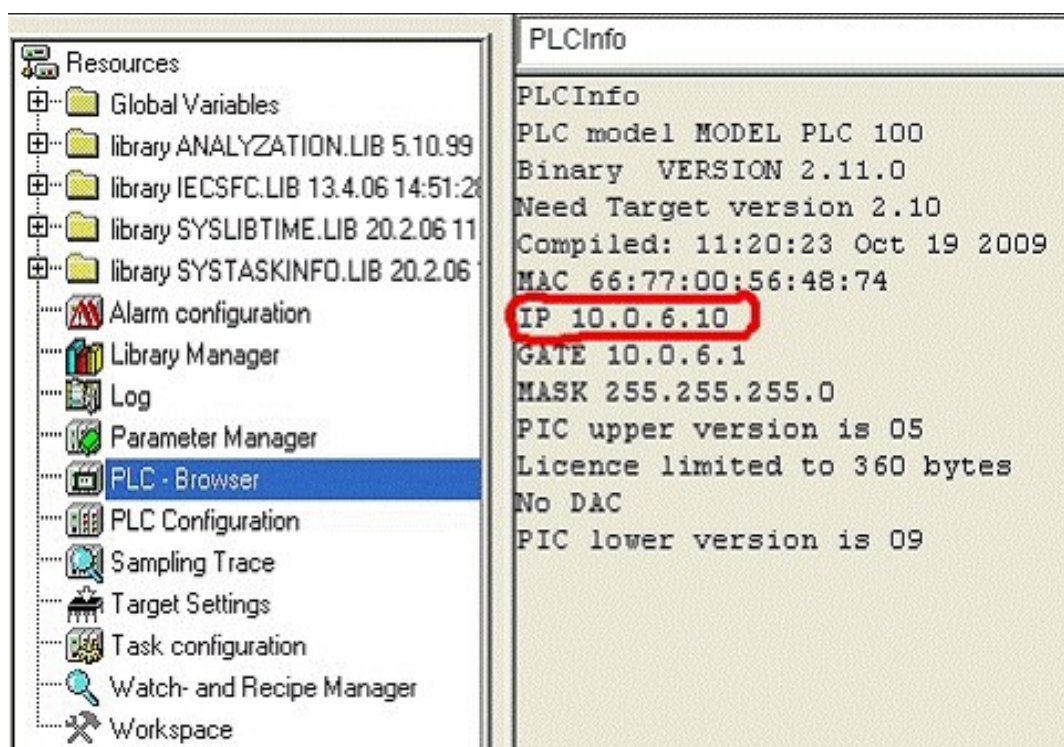


Рисунок 4.3 – Определение IP-адреса контроллера

#### 4.1.4 Указания по монтажу

Установка контроллеров на DIN-рейке осуществляется в следующей последовательности:

- 1) производится подготовка на DIN-рейке места для установки контроллера в соответствии с габаритами контроллера;
- 2) контроллер устанавливается на DIN-рейку в соответствии с рисунком 4.4 а) по стрелке 1;
- 3) контроллер с усилием прижимается к DIN-рейке в направлении, показанном стрелкой 2, до фиксации защелки;
- 4) для съема контроллера с DIN-рейки в проушину защелки вставляется острое отвертки (см. рисунок 4.4б), и защелка отжимается по стрелке 1, после чего контроллер отводится от DIN-рейки по стрелке 2.

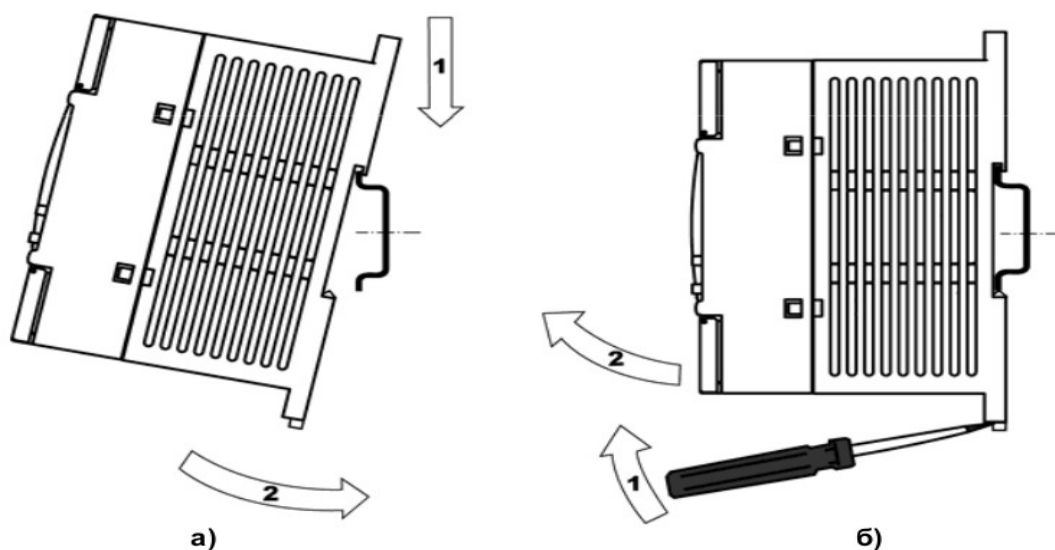


Рисунок 4.4 – Монтаж контроллера с креплением на DIN-рейку

Подключение к интерфейсу Ethernet выполняется восьмижильным кабелем «витая пара» категории 5. На кабель устанавливаются оконечные соединители без экрана. Ответная часть кабеля подключается к Ethernet-порту шкафа телемеханики. Используется бель Up-Link – кабель с перекрестным монтажом первой и второй пар.

Подготовить место для установки регулятора РНМ-1 в соответствии с разметкой крепления. Регулятор закрепляется четырьмя винтами. Соединить

винт заземления регулятора с контуром заземления подстанции медным проводом сечением не менее 2 кв. мм.

Измерительные цепи напряжения и тока регулятора подключить к вторичным обмоткам измерительных трансформаторов. При контроле двух секций измерительные трансформаторы тока и напряжения первой секции подключить к модулю ТР (секция 1), а измерительные трансформаторы тока и напряжения второй секции – к модулю ТР (секция 2). Измерительные трансформаторы тока подключаются в том случае, если необходимо вводить токовую компенсацию.

Контакты выходного релейного сигнала «Сигнализация», замыкающиеся при полной или частичной блокировке регулирования, подключить к цепям центральной сигнализации подстанции. Контакты выходных релейных сигналов «Отказ РНМ», «Отказ ПМ», «Перегрузка» подключить к цепям сигнализации при необходимости.

Контакты входных релейных сигналов «Прибавить», «Убавить», «Автоматическое/ручное управление», подключить к контроллеру Овен ПЛК согласно схеме подключения.

Питание регулятора подается на контакты «Питание» модуля БП. Питание осуществляется от сети переменного тока напряжением 220 В.

Питание цепей дискретных входов и выходов осуществляется от сети переменного или постоянного тока напряжением 220 В. Полярность подключения – произвольная.

Коммутационное оборудование и оборудование защиты питания монтируются на DIN-рейку.

## 5 Функционально-стоимостной и экономический анализ проекта

### 5.1 Функционально–стоимостной анализ проектируемой информационной системы

В данном разделе обоснована экономическая целесообразность проектируемого варианта системы диспетчерского управления. В общем случае экономический анализ целесообразности применения того или иного устройства должен базироваться не на сопоставлении их стоимости, а на сопоставлении общих затрат на создание систем управления данными устройствами и расходов на их обслуживание. С использованием корректирующей формы функционально-стоимостного анализа производится анализ проектируемого варианта технической системы, подвергшийся инновационным преобразованиям. В результате данного анализа в проектируемом варианте обнаруживаются функциональные и структурные элементы системы, обладающие экономической несостоятельностью или функциональной недостаточностью. При этом, кроме технических эффектов от разработки, появляются экономические, социальные, экологические и иные эффекты.

Структурная модель (СМ) - это упорядоченное представление элементов объекта и отношений между ними, дающее представление о составе материальных составляющих объекта, их основных взаимосвязях и уровнях иерархии.

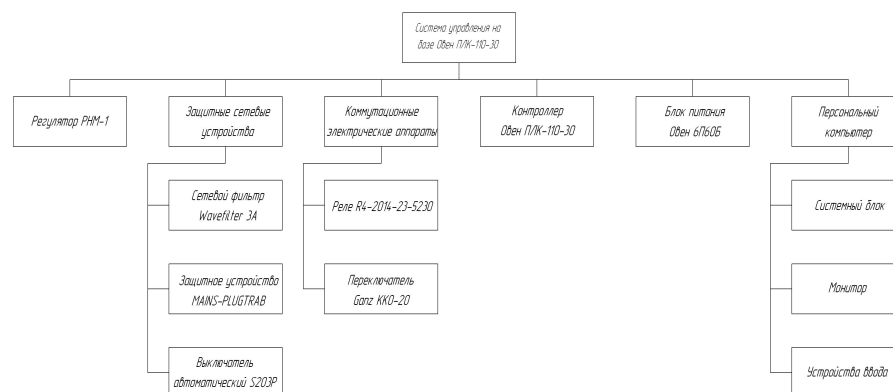


Рисунок 5.1 – Структурная модель проектируемого варианта

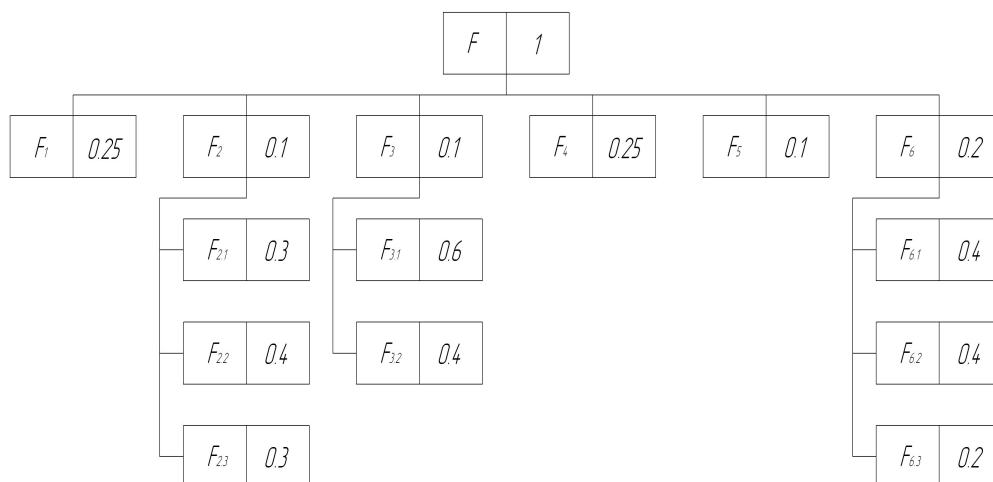


Рисунок 5.2 – Функциональная модель проектируемого варианта

Таблица 5.1 – Функционально-стоимостная модель для проектируемого варианта.

Индекс функции	Наименование функции	Материальный носитель ф-и	r	R	Q	S <sub>абс</sub>	S <sub>отн</sub>
1	2	3	4	5	6	7	8
F <sub>2.1</sub>	Фильтрация напряжения	Сетевой фильтр Wavefilter 3A	0.3	0.03	0.2	3100	0.038
F <sub>2.2</sub>	Защита от перенапряжения	Защитное устройство MAINS	0.4	0.04	0.2	2150	0.026
F <sub>2.3</sub>	Защита от КЗ	Автоматический выключатель S203P	0.3	0.03	0.1	720	0.008
F <sub>3.1</sub>	Коммутация сигналов	Реле R4-2014-23-5230-W TL	0.6	0.06	0.1	1800	0.02
F <sub>3.2</sub>	Переключение режимов	Переключатель Ganz KK0	0.4	0.04	0.1	1800	0.02
F <sub>6.1</sub>	Подготовка и трансляция программ	Системный блок	0.4	0.08	0.2	7000	0.085
F <sub>6.2</sub>	Отображение информации	Монитор	0.4	0.08	0.2	5000	0.06
F <sub>6.3</sub>	Набор программ	Устройства ввода	0.2	0.04	0.1	500	0.006



Продолжение таблицы 5.1

1	2	3	4	5	6	7	8
F1	Регулирование привода	Регулятор РНМ-1	0.25	0.25	0.4	45000	0.55
F2	Питание комплекса	Сетевой фильтр, защитное устройство	0.1	0.1	0.1	6320	0.077
F3	Коммутация сигналов	Реле и переключатели	0.1	0.1	0.2	3600	0.04
F4	Выполнение задач управления	Овен ПЛК-110-30	0.25	0.25	0.3	13000	0.16
F5	Питание контроллера	Блок питания Овен 6П60Б	0.1	0.1	0.1	1800	0.02
F6	Эксплуатация диспетчерского ПО	Персональный компьютер	0.2	0.2	0.3	12500	0.15

$$\Sigma S_{abc} = 82220$$

Путем совмещения структурной и функциональной модели строим совмещенную функционально-стоимостную модель проектируемого варианта, которая приведена на листе графической части.

## 5.2 Экономическая оценка проекта

Экономическая оценка проекта осуществляется с использованием следующих показателей:

а) чистая приведенная величина дохода (Net Present Value - NPV).

NPV представляет собой разность между приведенными к началу реализации проекта поступлениями от реализации проекта и инвестиционными затратами, т.е. сумму дисконтированного чистого денежного потока за период реализации проекта.

$$NPV = \sum_{t=0}^T NCF_t * PV, \quad (5.1)$$

где  $T$  - продолжительность реализации проекта;

$t$  - порядковый номер года реализации проекта;

$NCF_t$  - чистый денежный поток года  $t$ ;

$PV$  - коэффициент дисконтирования в году  $t$ ;

б) коэффициент дисконтирования ( $PV$ -фактор) для года  $t$  определяется по формуле:

$$PV_t = \frac{1}{(1+r)^t}, \quad (5.2)$$

где  $r$  - ставка дисконта.

Период реализации проекта может определяться:

- периодом времени, в течение которого модернизируемый или рационализируемый объект будет амортизирован;
- периодом морального износа объекта;
- внутренняя норма доходности.

Internal Rate of Return (IRR) - это то значение ставки дисконтирования, при котором сумма дисконтированных поступлений денежных средств равна сумме дисконтированных платежей или чистая приведенная величина дохода (NPV) обращается в ноль.

Таким образом, значение внутренней нормы доходности определяется из соотношения:

$$\sum_{t=0}^T NCF_t * \frac{1}{(1+IRR)^t} = 0 \quad (5.3)$$

г) периодом окупаемости проекта или периодом возврата инвестиций (Discounted Payback Period - DPP) называется период времени, за который дисконтированные поступления от результатов внедрения проектных решений покроют инвестиции, т.е. значение периода окупаемости определяется из



соотношения:

$$\sum_{t=1}^{DPP} CF_t * IRR = IN, \quad (5.4)$$

где  $CF_t$  - поступление денежных средств от эффектов, связанных с внедрением проекта;

$IN$  - инвестиционные затраты.

Проект считается эффективным, если приведенная величина дохода положительна, внутренняя норма доходности больше ставки дисконтирования ( $IRR > r$ ) [16].

Инвестиции в данный проект составляют 50000 рублей:

- 13000 – на покупку Овен Плк-100-30;
- 12000 – на покупку персонального компьютера;
- 10000 – на покупку шкафа и монтаж контроллера;
- 15000 – на разработку программного и методического обеспечения.

В качестве значения ставки дисконта принимается ставка Центрального банка России – 8.25%. Значение коэффициентов дисконтирования по годам:  $PV_1 = 0.92$ ;  $PV_2 = 0.85$ ;  $PV_3 = 0.79$ ;  $PV_4 = 0.73$   $PV_5 = 0.67$ .

Методом трассировки находим  $IRR=80\%$ .

Схема формирования чистого денежного потока представлена в таблице на листе графической части дипломного проекта.

Исходя из приведенных расчетов, можно сделать вывод о том, что проект является эффективным, т.к. чистая приведенная величина дохода положительна, внутренняя норма доходности больше ставки дисконтирования.

По формуле (5.4):

$$\sum_{t=1}^1 CF_t * 0,8 = 27600$$
$$\sum_{t=1}^2 CF_t * 0,8 = 55200$$

Проект окупается в промежутке между первым и вторым годом.

Так как окупаемость считаем с дискретой в 1 год, то делаем вывод о том, что проект полностью окупается за 2 года.

## 6 Безопасность и экологичность проекта

В современном высоко индустриальном обществе проблема обеспечения безопасности становится все более острой. По мере развития и совершенствования технических средств, используемых человеком, ужесточаются требования надежности и экологичности, предъявляемые к ним. При этом на данном этапе социального развития общества особо необходимыми являются такие показатели надежности, как смягчение последствий, связанных с авариями на производстве, уменьшения травмоопасности, загрязнения окружающей среды.

Несмотря на переход в оценке крупных объектов производства с понятия «безопасность» к понятию «приемлемый риск» на производственных комплексах меньшего масштаба необходимо с максимальной полнотой обеспечить безопасность людей при работе с данными объектами, естественно, при соблюдении ими правил техники безопасности.

Еще более острой эта проблема видится в аспекте социально-экономических последствий, возникающих при несоблюдении правил безопасности или недооценке со стороны разработчиков опасности своего объекта.

В процессе труда человек вступает во взаимодействие с предметами труда, орудиями труда, другими людьми. Кроме того, на него воздействуют различные факторы производственной обстановки (температура, влажность и подвижность воздуха, шум, вибрация, вредные вещества, различные излучения и т. п.). От условий труда в большой степени зависят здоровье и работоспособность человека, его отношение к труду и результаты труда. При плохих условиях резко снижается производительность труда, и создаются предпосылки для возникновения травм и профессиональных заболеваний.

### 6.1 Анализ безопасности труда диспетчера

Профессия диспетчера является сложной и ответственной, требующий

наличия теоретических знаний и навыка их практического применения в конкретных условиях, данный род деятельности является уникальным. Для работы диспетчером требуется высшее техническое образование по специальностям, прямо или косвенно связанным с электротехникой. Кроме этого необходимо иметь стаж работы на энергообъекте не менее 5 лет, пройти специальный тест по психическому состоянию, уровню слуховой и зрительной памяти, скорости и адекватности реакции на ситуацию. Актуальность проблемы безопасности труда диспетчеров энергосистемы очевидна. Рассмотрим трудовую деятельность диспетчера энергосистемы. Основными функциями труда диспетчера являются:

- 1) контроль за режимом электроснабжения, поддержание заданной величины параметров энергосистемы (напряжение, нагрузка, частота);
- 2) в случае сбоя или создания аварийной ситуации – быстрое принятие решения о необходимых мерах и средствах по ликвидации аварии, включению резервного оборудования, восстановлению нормального питания потребителей;
- 3) проведение мероприятий по плановым остановам оборудования сетей (электроподстанций, воздушных линий), что подразумевает правильную последовательность операций по выводу в ремонт, организации безопасного ведения работ на высоковольтном оборудовании, обеспечению недопущения поражения персонала электротоком.

Рабочий день диспетчера строго регламентирован. Работа имеет сменный характер: дневная смена 11 часов (с 8:00 до 19:00), через сутки ночная смена 13 часов (с 19:00 до 8:00), двое суток отдыха. Смена начинается с приемки её у предыдущего диспетчера, выяснения состояния схемы, оборудования, предстоящих плановых переключений, текущих директивных документов и распоряжений. Далее следует прием рапортов подчиненного персонала с объектов по спецсвязи (расположены по региону, расстояния большие), отдача рапорта своему оперативному руководству и выполнение своих обязанностей.

Диспетчерский пункт работает круглосуточно без выходных и

праздников, вне зависимости от природных условий. Диспетчер работает вдвоём с напарником – помощником диспетчера, сменяющим его во время обеда. Оставлять пост без команды вышестоящего диспетчера (диспетчерского пункта) запрещено. Перерыв в электроснабжении потребителей, отключенных действием специальной автоматики отключения нагрузки, не должен превышать 20 минут. Соответственно за время, меньшее 20 минут, диспетчер обязан правильно среагировать и предотвратить развитие аварии.

Комплекс контроля параметров энергосистемы представляет собой набор радиоэлектронных устройств, источником питания которых является сеть с напряжением переменного тока 220 В и частотой 50 Гц, установленных в специализированном помещении - машинном зале. Численность обслуживающего персонала – 4 человека, работающих посменно, в результате чего круглосуточно осуществляется контроль за работой комплекса. Машинный зал по «Правилам устройства электроустановок» (ПУЭ (6-е изд.) в разд. 1.1.13) относится к помещениям без повышенной опасности (напряжение переменного тока не более 220 В, влажность менее 75%, отсутствие токопроводящей пыли, токопроводящих полов, высокой температуры (выше 35 °С), возможности одновременного прикосновения человека к имеющим соединение с землей металлоконструкциям зданий, технологическим аппаратам, механизмам и т.п., с одной стороны, и к металлическим корпусам электрооборудования - с другой). Статистических данных о несчастных случаях, инцидентах, связанных с эксплуатацией комплекса нет.

Рабочее место диспетчера это особое помещение на предприятии, изолированное от основного пространства с ограничением доступа. Для круглосуточного пребывания на работе созданы все необходимые условия: мебель, холодильник, микроволновка, посуда и другие. Зрительную работу можно охарактеризовать как работу со светоизлучающими материалами, коэффициент естественного освещения равен 3, искусственное освещение включает общее и аварийное освещение. Микроклимат поддерживается системой кондиционирования. В рабочей комнате имеется телевизор,

специальные средства канальной и радиосвязи, персональные компьютеры и проекционный видеосит с оперативной информацией.

Интенсивные источники шума отсутствуют, уровень шума складывается из систем охлаждения электронной аппаратуры и не превышает допустимого уровня, однако длительное воздействие шума может повлечь к дискомфорту и повышенной утомляемости.

Работа является однообразной и напряжённой, из-за чего диспетчер может испытывать различные нагрузки на психику, утомление, дискомфорт, влекущие за собой снижение производительности труда и утрату способности правильно реагировать на обстановку. Диспетчер самостоятельно принимает ответственные решения, по этой причине вынужден контролировать психоэмоциональное состояние.

Длительное пребывание в одном и том же положении, и монотонность движений приводит к так называемому синдрому длительных статических нагрузок, вызывающему мышечные боли и воспалительные процессы. Большой объем перерабатываемой информации, приводит к значительной нагрузке на глаза и различным заболеваниям (астенопия, головные боли, двоение в глазах и нарушение фокусировки зрения).

Вредными факторами являются: широкий спектр излучения от дисплея, который включает в себя рентгеновскую, ультрафиолетовую и инфракрасную области, а также широкий диапазон электромагнитных волн других частот от устройств радиоаппаратуры.

Условия труда в целом оцениваются по четырём классам: оптимальные, допустимые, вредные, опасные (экстремальные).

На работе диспетчер сталкивается со специфическими для этой профессии вредными факторами, являющимися, однако, допустимыми. На текущем уровне технологического развития отказаться от диспетчера в пользу автоматических систем управления не является возможным ввиду отсутствия алгоритмов высокоуровневого управления и планирования.

#### 6.1.1 Специальная оценка условий труда

Условия труда — это совокупность факторов производственной среды и трудового процесса, оказывающих влияние на здоровье и работоспособность человека в процессе труда.

С 1 01.2014 года в организациях вводится специальная оценка условий труда, проведение которой регламентируется федеральным законом № 426-ФЗ «О специальной оценке условий труда». [20]

Методика проведения специальной оценки труда устанавливает обязательные требования к следующим процедурам:

- идентификации потенциально вредных и (или) опасных факторов производственной среды и трудового процесса;
- исследованию (испытанию) и измерению идентифицированных потенциально вредных и (или) опасных факторов производственной среды и трудового процесса;
- отнесению условий труда на рабочих местах к классам (подклассам) условий труда по степени вредности или опасности по результатам проведения исследований (испытаний) и измерений идентифицированных потенциально вредных и (или) опасных факторов производственной среды и трудового процесса;
- оформление результатов специальной оценки условий труда.

До проведения оценки работодателю нужно выполнить следующие подготовительные мероприятия:

- создать комиссию;
- утвердить перечень рабочих мест, на которых будет осуществляться оценка условий труда, и график ее проведения.

Конкретный состав комиссии и порядок ее деятельности утверждается приказом руководителя.

Далее выполняется идентификация потенциально вредных и (или) опасных производственных факторов. Идентификация представляет собой процедуру, которая выявляет наличие на рабочем месте факторов, указанных в классификаторе как вредные и (или) опасные. Выявление на рабочих местах факторов осуществляется путем изучения:

- технической (эксплуатационной) документации на производственное оборудование, используемое на рабочих местах конкретных работников;

- технологической документации, характеристик технологического процесса;
- проектов строительства и (или) реконструкции производственных объектов (зданий, сооружений, производственных помещений);
- характеристик применяемых в производстве веществ, материалов, сырья;
- деклараций о соответствии и (или) сертификатов соответствия производственного оборудования, машин, механизмов, инструментов и приспособлений, технологических процессов, веществ, материалов, сырья установленным требованиям.

Выявление на рабочих местах факторов может проводиться путем обследования рабочих мест методом осмотра и ознакомления с работами, фактически выполняемыми работниками в режиме штатной работы, а также путем опроса работников и (или) их непосредственных руководителей. Если идентификация выявила вредные или опасные производственные факторы, то их исследуют и измеряют в порядке, установленном статьей 12 Закона. [20]

Условия труда на рабочем месте, при которых отсутствуют идентифицированные потенциально вредные (опасные) факторы, признаются комиссией по проведению специальной оценки условий труда допустимыми условиями труда. В случае несовпадения наименований выявленных факторов и факторов, указанных в соответствующих разделах Классификатора, выявленные факторы признаются не идентифицированными.

В пункте 4 методики проведения специальной оценки приведена исчерпывающая информация по отнесению условий труда к соответствующему классу (подклассу) при воздействии различных факторов.

Составляются карты специальной оценки условий труда на каждом из рабочих мест, заполняются протоколы измерений (оценки) для каждого из факторов. По результатам оценки организация, которая ее проводила, составляет отчет. В него включаются перечень рабочих мест, на которых проводилась оценка, протоколы проведения исследований, перечень



мероприятий по улучшению условий и охраны труда работников и др. Отчет подписывают все члены комиссии. После этого работодателю нужно ознакомить работников под роспись с результатами оценки.

От результатов специальной оценки будет зависеть размер страховых взносов в Пенсионный фонд по дополнительному тарифу за сотрудников, занятых на вредном и опасном производстве. В соответствии со ст.11 закона №421-ФЗ, с 01 января 2015 года, вступит в силу новая редакция КоАП, согласно которой за не проведение спецоценки, а так же за нарушение правил ее проведения, на организацию могут наложить административный штраф.

## 6.2 Экологическая безопасность и охрана окружающей среды

### 6.2.1 Влияние подстанции «Левашово» на окружающую среду

Объект расположен в Курганской области, г. Кургане, Шадринский тракт (напротив пос. Чистое поле), в северо-западной части по направлению от центра города.

В геоморфологическом отношении площадка приурочена к 1-й надпойменной террасе р. Тобол. Рельеф участка трассы ВЛ относительно ровный. Наблюдается небольшой уклон в западном направлении. Абсолютные отметки изменяются в пределах 68,59 – 74,17 м. Территория трассы ВЛ свободна от застройки и от лесонасаждений. Дорожная сеть представлена грунтовой автодорогой п. Радужный - г. Курган.

Рельеф площадки ПС слабо наклонен в юго-западном направлении.

Территория участка строительства ПС свободна от застройки и от лесонасаждений, находясь на открытом пространстве.

Воздействие объекта на атмосферный воздух оценивается для двух периодов: строительного и эксплуатационного.

Уровень загрязнения атмосферы в районе расположения реконструируемого объекта определяется выбросами загрязняющих веществ от источников непосредственно самого предприятия, а также выбросами близ

расположенных источников загрязнения атмосферы. В районе расположения проектируемого объекта отсутствуют крупные промышленные объекты. Основным видом воздействия на атмосферный воздух – выбросы ЗВ от автотранспорта.

Основным видом воздействия на состояние воздушного бассейна в районе проведения работ является загрязнение атмосферного воздуха выбросами от автотранспорта. На стадии строительства источниками выбросов загрязняющих веществ являются двигатели автотранспорта и дорожно-строительной техники, а также выбросы от сварочных и окрасочных работ.

Для расчета выбросов ЗВ в атмосферу принята одновременная работа экскаватора и автосамосвала. При работе двигателей автотранспорта в атмосферу выбрасывается: азота диоксид, азота оксид, серы диоксид, углерод черный (сажа), углерода оксид, бензин нефтяной, керосин.

Для выполнения сварочных работ используются электроды марки УОНИ 13/45. При сварочных работах в атмосферу выбрасываются: углерода оксид, железа оксид, марганец и его соединения, азот (IV) оксид (Азота диоксид), азот (II) оксид (Азота оксид), фториды газообразные, фториды плохо растворимые, пыль неорганическая: 70-20% SiO<sub>2</sub>.

Строительные машины и оборудование должны находиться на объекте только на протяжении периода производства соответствующих работ. Все источники выбросов при строительстве являются неорганизованными и нестационарными.

Расчеты выбросов загрязняющих веществ в атмосферу от основных технологических операций при производстве строительных работ, работе автотранспорта и дорожных машин выполнены на основе удельных показателей выбросов с применением нормативных документов по расчету выбросов в атмосферу различными производствами.

Таблица 6.1 – Перечень загрязняющих веществ, выбрасываемых в атмосферу на период строительства

Вещество		Критерий	Значение кг/м <sup>3</sup>	Класс опасности	Суммарный выброс	
Код	Наименование				г/с	т/год
0123	Железа оксид	ПДК с/с	0.04	3	0.0003	0.0001
0143	Марганец	ПДК м/р	0.01	2	0.0001	0.00003
0301	Азота оксид (IV)	ПДК м/р	0.2	2	0.037	0.194
0304	Азота оксид (II)	ПДК м/р	0.4	3	0.006	0.03
0328	Углерод чёрный (сажа)	ПДК м/р	0.15	3	0.0076	0.035
0330	Сера диоксид	ПДК м/р	0.5	3	0.0047	0.022
0337	Углерод оксид	ПДК м/р	5	4	0.107	0.196
0342	Фториды газообразные	ПДК м/р	0.02	2	0.00007	0.000031
0344	Фториды плохорастворимые	ПДК м/р	0.2	2	0.00005	0.00002
2732	Керосин	ОБУВ	1.2	0	0.018	0.05
2908	Пыль неорганическая	ПДК м/р	0.3	3	0.00005	0.000021
Всего веществ: 11					0.18	0.53
в том числе твёрдых: 2					0.00038	0.00015
жидких и газообразных: 9					0.18	0.53

Для всех вредных веществ, выбрасываемых в атмосферу на период строительства рассматриваемого объекта, определены нормативы предельно-допустимых выбросов. В связи с тем, что приземные концентрации по всем ингредиентам не превышают значение в 1 ПДК, то выбросы на период строительства считаются допустимыми.

На период эксплуатации источники выброса загрязняющих веществ в атмосферу отсутствуют, не включая локальных выбросов при проведении ремонтных работ. В период эксплуатации рассматриваемого объекта загрязнение атмосферы ЗВ исключено в виду отсутствия источников выброса ЗВ.

Гидрогеологические условия на участке изысканий характеризуются наличием горизонта грунтовых вод, приуроченного к суглинкам текучей

консистенции, встреченного и установившегося на глубине 2,0-4,0 м. В различные периоды года уровень может колебаться на 0,5 – 1,0 м, в связи с инфильтрацией атмосферных осадков, а также в результате техногенного воздействия (утечки из коммуникаций). По химическому составу кальциевые. Коррозионная агрессивность воды по отношению к свинцовой оболочке кабеля – высокая, по отношению к алюминиевой оболочке кабеля – высокая.

Для снижения отрицательных воздействий на гидрологический режим участка и прилегающей к нему территории на период строительства предусмотрены следующие мероприятия:

- проектом предусмотрена установка системы очистки колес «Мойдодыр» на выезде с строительных городков;
- все канализационные и бытовые стоки собираются в емкости, а затем собираются вместе специализированным автотранспортом и вывозятся на специализированное предприятие.

Основное значение будут иметь механические нарушения поверхности почв под влиянием передвижных транспортных средств и земляных работ, связанных с разработкой грунта и устройством временных проездов и площадок.

При снятии техногенных нагрузок (т.е. по окончании строительства) большая часть указанных нарушений ликвидируется благодаря организационно-техническим мероприятиям и благоустройству территории.

По окончании строительства образовавшийся строительный мусор вывозится, выполняются планировочные работы, и проводится благоустройство территории.

В зоне проектирования объекта отсутствуют редкие исчезающие виды растений и животных. В связи с достаточной освоенностью района расположения объекта, места гнездования и пути миграции животных на данной территории отсутствуют. Фауна рассматриваемой территории бедна, в связи с близостью людских поселений и транспортных магистралей. Представители фауны адаптированы к шумовым воздействиям, возникающим

в городской черте.

Проведение строительных работ приведет к нарушению естественной среды обитания животного мира и изъятию мест и регулярно используемых убежищ и сооружений (гнезд). Кроме того, шум строительной техники отпугнет животных различных экологических групп от мест их обычной кормежки и охоты. Воздействие на животный мир прогнозируется допустимым. Следует также отметить, что производственная деятельность человека на площадке проектируемого строительства, не окажет негативного влияния на миграционные пути животных и птиц.

#### 6.2.2 Влияние электромагнитного поля на население и животный мир

Вредное действие магнитного поля на живые организмы, и в первую очередь на человека, проявляется только при высоких напряженностях порядка 150 – 200 А/м, возникающих на расстояниях до 1 – 1,5 м от проводов фаз ВЛ, и представляет опасность при работе под напряжением.

Основные проблемы для линий СВН и УВН связаны с влиянием электрического поля, создаваемого ВЛ. Это поле определяется, в основном, зарядами фаз. С повышением напряжения ВЛ, числа проводов в фазе и эквивалентного радиуса расщепленного провода заряд фазы быстро увеличивается. Так, заряд фазы линии 750 кВ в 5—6 раз больше заряда одиночного провода линии 220 кВ, а линии 1150 кВ в 10—20 раз. Это создает напряженности электрического поля под проводами ВЛ, опасные для живых организмов.

Непосредственное (биологическое) влияние электромагнитного поля линий СВН и УВН на человека связано с воздействием на сердечно-сосудистую, центральную и периферийную нервные системы, мышечную ткань и другие органы. При этом возможны изменения давления и пульса, сердцебиение, аритмия, повышенная нервная возбудимость и утомляемость. Вредные последствия пребывания человека в сильном

электрическом поле зависят от напряженности поля  $E$  и от продолжительности его воздействия.

Без учета длительности воздействия на человека допускаемая напряженность электрического поля составляет:

- 20 кВ/м - для труднодоступной местности;
- 15 кВ/м - для ненаселенной местности;
- 10 кВ/м для пересечений с дорогами;
- 5 кВ/м для населенной местности.

При напряженности 0,5 кВ/м на границах жилых застроек допускается пребывание человека в электрическом поле по 24 ч в сутки в течение всей жизни.

Для эксплуатационного персонала подстанций и линии СВН и УВН установлена допустимая продолжительность периодического и длительного пребывания в электрическом поле при напряженностях на уровне головы человека (1,8 м над уровнем земли):

- 5 кВ/м — время пребывания неограниченно;
- 10 кВ/м — 180 мин;
- 15 кВ/м — 90 мин;
- 20 кВ/м — 10 мин;
- 25 кВ/м — 5 мин.

При невозможности ограничения времени пребывания персонала под воздействием электрического поля применяется экранирование рабочих мест, тросовые экраны над дорогами, экранирующие козырьки и навесы над шкафами управления, вертикальные экраны между фазами, съемные экраны при ремонтных работах и др. Как показали эксперименты, надежный экранирующий эффект создают кустарники высотой 3—3,5 м и плодовые деревья высотой 6—8 м, растущие под ВЛ. Это объясняется тем, что кусты и плодовые деревья обладают достаточной проводимостью и выполняют роль экрана на высоте, превышающей рост человека или высоту транспортных средств.

Косвенное воздействие электрического поля включается в возникновении тока или кратковременных разрядов при прикосновении человека, имеющего хороший контакт с землей, к изолированным объектам или, наоборот, при прикосновении изолированного от земли человека к заземленным объектам. Такие явления объясняются наличием повышенных потенциалов и ЭДС, наведенных электромагнитным полем на машинах, механизмах или протяженных металлических предметах, изолированных от земли.

Разрядный ток, протекающий через человека, зависит от напряжения линии, активного сопротивления человека, объема и емкости объектов относительно линии. Длительный ток, достигающий 1 мА, для большинства людей является «порогом восприятия». При токе 2—3 мА возникает испуг, при 8-9 мА («порог отпускания») — болевые ощущения и мышечные судороги. Токи свыше 100 мА, протекающие через человека более 3 с, могут привести к смертельному исходу.

### 6.3 Предупреждение техногенных аварий и защита в ЧС

#### 6.3.1 Анализ возможных ЧС в районе расположения подстанции «Левашово»

Предупреждение и ликвидация чрезвычайных ситуаций — это комплекс мероприятий по предотвращению рисков, включает в себя подготовку к бедствию или катастрофе до их наступления, реакцию на чрезвычайную ситуацию (например, эвакуация, карантин, дезактивация и т. д.), поддержка населения и участие в восстановлении после природных и техногенных катастроф.

Подстанция «Левашово» является стратегически важным объектом, входящим в единую энергосистему России и обеспечивающим связь КТЭЦ-1 и КТЭЦ-2 с отпайкой на КЗММК. Сбой на подстанции может привести к нестабильной работе энергосистемы г. Кургана, частичному и полному

отключению потребителей. По этой причине необходимо детально проанализировать и предупредить возможные ЧС.

Подстанция размещена на огороженной территории и состоит из следующих объёмно-планировочных элементов: модульное здание ЗРУ 10 кВ, совмещенное с ОПУ, ОРУ 110 кВ с порталами и прожекторными мачтами, 2 трансформатора 110/10 кВ, один въезд на территорию, ограждение территории из сборных железобетонных элементов, с калитками и воротами.

Отключение электричества само по себе является серьезной техногенной аварией, способной повлечь за собой остановку производства, нарушить работу спец-предприятий, больниц, парализовать работу органов власти и управления, банков, связи, торговли, транспорта, коммунальных служб. Наиболее известной техногенной аварией подобного рода является авария в энергосистеме Москвы на подстанции «Чагино» в 2005 году, когда была отключена подача электроэнергии в несколько районов Москвы, Подмосковья, а также Тульской, Калужской и Рязанской областей. Несколько десятков тысяч человек оказались заблокированы в остановившихся поездах московского метро и лифтах, было нарушено железнодорожное сообщение и парализована работа многих коммерческих и государственных организаций.

Наиболее очевидным и вероятным нарушением работы подстанции может явиться пожар. Причиной которого может быть как внутренние, так и внешние факторы, в частности пробой изоляции и образование электрической дуги, сезонные осенне-весенние обострения пожарной обстановки ввиду поджога сухого травяного покрова.

Силовая подстанция по функциональной пожарной опасности относится к классу Ф5.1 (производственные здания и сооружения, производственные и лабораторные помещения, мастерские) ст. 32 №123-ФЗ, п.5.21\* СНиП 21-01-97 [24].

В соответствии с требованиями ПУЭ 7-ое издание [23], для предотвращения растекания масла силовых масляных трансформаторов и распространения пожара при аварии маслонаполненного силового



трансформатора на ПС под трансформатором предусмотрен маслоприемник (без отвода масла). Маслоприемник выполнен маслостойким, монолитным, железобетонным из бетона класса В15 по уплотненному щебню слоем 500 мм, пропитанному битумом до полного насыщения. В соответствии с п. 4.2.69 ПУЭ[23], 7-е издание, в маслоприемник засыпан слой чистого гравия или промытого гранитного щебня толщиной не менее 0,25 м поверх металлической решетки.

Помещения модульного здания отделены друг от друга перегородками. Перегородки принимаются 1-го типа (EI 45) согласно No 123-ФЗ [24]. Тип заполнения в перегородке 2-й (EI 30). Установлена металлическая дверь с EI 30. Наружная дверь выполнена в противопожарном исполнении с пределом огнестойкости 30 EI по ГОСТ 31173-2003 [10]. В местах ответвлений кабелей и прохода их через стены и перекрытия предусмотрены огнестойкие перегородки с пределом огнестойкости не менее EI45(п.8.3, 8.26 РД 153-34.0-49.101-2003 [11]). В соответствии с «Правилами пожарной безопасности для энергетических предприятий» СО 34.03.301-00 (РД 153-34.0-03.301-00 [26]) после прокладки кабелей выполнены огнестойкие пояса и уплотнения с пределом огнестойкости не менее REI45 в кабельных лотках через 50,0 м по длине, в узлах поворота и ответвлений, в местах прохода кабелей через строительные конструкции.

Другими причинами чрезвычайных ситуаций могут явиться аварии с участием поездов, перевозящих опасные и вредные вещества по железнодорожной линии, находящейся на расстоянии 1 км от подстанции. Так как постоянный персонал на подстанции отсутствует, необходимости в дополнительной биологической защите не требуется, но подобного рода ситуации могут негативно сказаться на обслуживании и ремонте оборудования.

Известной проблемой в регионе с резко-континентальным климатом и холодными продолжительными зимами является явление оледенения проводов. Данное явление, как правило, происходит в осенне-зимний период. Осенью реальной угрозой являются туман и заморозки, а зимой перепады

температур и налипание снега на провода ЛЭП, что нередко приводит в разрыву провода и обрыву с места крепления.

Однако наиболее серьёзной и высоковероятной причиной возникновения аварии на подстанции являются удары молнии при грозах.

### 6.3.2 Расчёт молниезащиты подстанции

Все объекты по требованиям к устройству их молниезащиты подразделяются на обычные и специальные. Обычные объекты — это жилые и административные строения, а также здания и сооружения, высотой не более 60 м, предназначенные для торговли, промышленного производства, сельского хозяйства. Специальные объекты – объекты, представляющие опасность для непосредственного окружения:

- для социальной и физической окружающей среды (объекты, которые при поражении молнией могут вызвать вредные биологические, химические и радиоактивные выбросы);

- прочие объекты, для которых может предусматриваться специальная молниезащита, например, строения высотой более 60 м, игровые площадки, временные сооружения, строящиеся объекты.

К выполнению молниезащиты различных объектов принят дифференцированный подход, здания и сооружения разделены на три категории, отличающиеся по тяжести возможных последствий поражения молнией [27].

К I категории отнесены производственные помещения, в которых в нормальных технологических режимах могут находиться и образовываться взрывоопасные концентрации газов, паров, пылей, волокон. Любое поражение молнией, вызывая взрыв, создает повышенную опасность разрушений и жертв не только для данного объекта, но и для близрасположенных.

Во II категорию попадают производственные здания и сооружения, в которых появление взрывоопасной концентрации происходит в результате

нарушения нормального технологического режима, а также наружные установки, содержащие взрывоопасные жидкости и газы.

К III категории отнесены объекты с пожароопасными помещениями или строительными конструкциями низкой огнестойкости. Кроме того, к III категории отнесены объекты, поражение которых представляет опасность электрического воздействия на людей и животных: большие общественные здания, животноводческие строения, высокие сооружения типа труб, башен, монументов.

При проектировании устройств молниезащиты объекта требуется установить его класс, принять необходимый уровень надежности защиты от прямых ударов молнии, определить объем защитных мер от вторичных воздействия молнии.

Таблица 6.2 – Уровни надежности защиты от прямых ударов молнии [27]

Уровень защиты	Надежность защиты от прямых ударов молнии (ПУМ)
I	0,98
II	0,95
III	0,90
IV	0,80

Подстанция «Левашово» относится к III категории и 1-му уровню надёжности.

Защиту распреедустройств проектируемой подстанции осуществляем молниеотводами. Молниеотвод состоит из металлического молниеприемника, который возвышается над защищаемым объектом и воспринимает удар молнии, и токопроводящего спуска с заземлителем, через который ток молнии отводится в землю.

Зона защиты одиночного стержневого молниеотвода высотой представляет собой круговой конус, вершина которого находится на высоте

. На уровне земли зона защиты образует круг радиусом .  
Горизонтальное сечение зоны защиты на высоте защищаемого сооружения

представляет собой круг радиусом . Зоны защиты одиночных стержневых молниеотводов высотой имеют следующие габаритные размеры.

Зона А:

(6.1)

Зона Б:

(6.2)

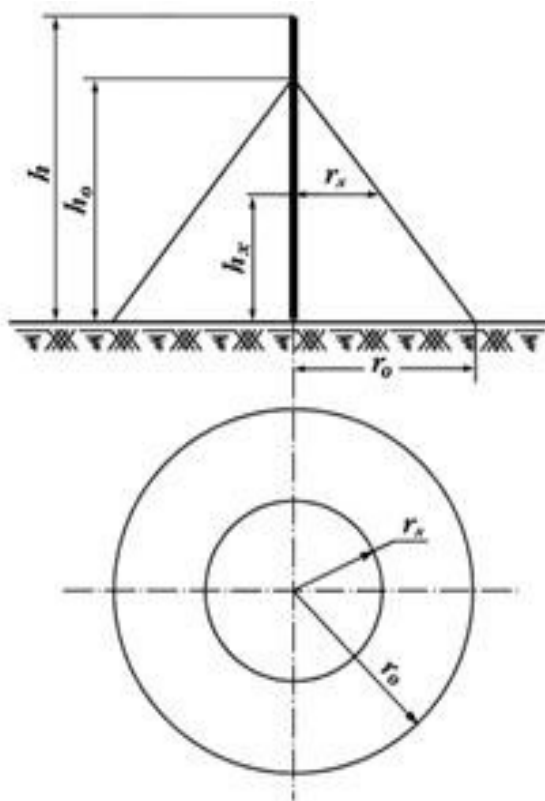


Рисунок 6.1 – Схема одиночного стержневого молниеотвода

Применим четыре стержневых молниеотвода. Габариты ограждения подстанции составляют 60х70 метров, устанавливаем молниеотводы по углам территории подстанции на расстоянии 3 м от ограждения (см. Рис. 6.2).

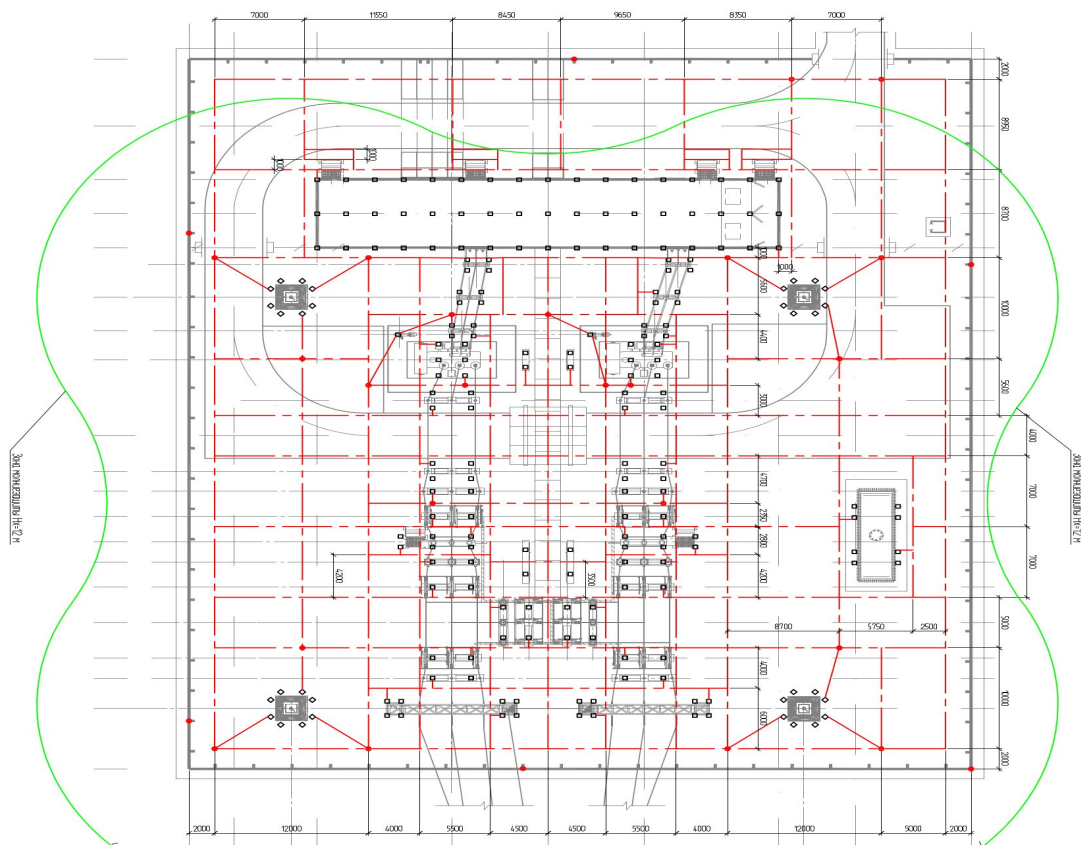


Рисунок 6.2 – План расположения заземления и молниезащиты

Тогда расчетные расстояния между молниеотводами

м;

м;

м.

Высота молниеотвода =29 м.

м;

/

м;

м.

Зона защиты определяется как зона защиты попарно взятых соседних молниеотводов. Условием защищенности объектов высотой является выполнение неравенства для всех попарно взятых молниеотводов. Где принимаем 12 м.

$M_1$

$M_2$

$M_3$

$M_4$

$M_5$

$M_6$

Для всех . Радиус действия молниеотводов позволяет защитить подстанцию от прямых ударов молнии.

## ЗАКЛЮЧЕНИЕ

В данной работе была модернизирована автоматизированная система диспетчерского управления подстанции Левашово. Сформированы аппаратные и сетевые структуры оперативно-информационного комплекса АСКУЭ в составе диспетчерского пункта и системы телемеханизации подстанции. Разработаны схема электрическая принципиальная и схема электрическая подключения программируемого контроллера. Разработан шкаф управления. Разработаны алгоритмы управляющих программ. Произведено моделирование работы энергосистемы, рассмотрены алгоритмы оптимизации режима и планирования потребления. Написано приложение контроля параметров подстанции.

Произведен функционально-стоимостной и экономический анализ проекта и анализ с точки зрения безопасности и влияния на окружающую среду.

Дальнейшее развитие на аппаратном уровне могут получить как компоненты оперативно-информационной системы и обработки телемеханической информации так и элементы телемеханизации подстанции в плане все большей их интеграции и унификации технических средств. Одним из направлений развития является больший охват управляемого оборудования энергообъектов.

На программном уровне необходимо создание недостающих программных модулей для программируемого контроллера Овен ПЛК, совершенствование и оптимизация существующих, расширение функциональных возможностей приложения контроля параметров. По необходимости протокол передачи управляющих сигналов может быть подвергнут криптографической защите

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Документация на оперативно информационный комплекс «СК-2007»;
- 2 Автоматизация диспетчерского управления в электроэнергетике/ Под общ. ред. Ю.Н. Руденко и В.А. Семенова. — М.: Издательство МЭИ, 2000. — 648 с.: ил.
- 3 Системы диспетчерского управления и сбора данных (SCADA-системы), Журнал Мир компьютерной автоматизации (3/1999)
- 4 Пухов А.С. Синтез решений при создании автоматизированных технических объектов: Учебное пособие – Курган: Изд-во Курганского гос. ун-та, 2000. –121с.: ил
- 5 Центральная приемо-передающая станция «SMART-FEP» руководство по эксплуатации. 2006 – 42с.: ил.
- 6 Система проекционных видеоцитов «Varco» руководство по эксплуатации. 2003 - 82с.: ил.
- 7 «Овен ПЛК 110» - Контроллер программируемый логический руководство по эксплуатации. 2009 - 72с.: ил.
- 8 Одноканальный блок питания БП60Б-Д4-х руководство по эксплуатации. 2009 - 2с.: ил.
- 9 [www.raspberrypi.org/documentation](http://www.raspberrypi.org/documentation)
- 10 Аппаратура контролируемого пункта «ИСЕТЬ» руководство по эксплуатации. 2009 - 118с.: ил.
- 11 Регулятор напряжения трансформатора микропроцессорный «РНМ-1» руководство по эксплуатации, паспорт. 2005 -50с.: ил.
- 12 Комплекс «Консоль ПП» руководство по эксплуатации. 2007 - 75с.: ил.
- 13 ГОСТ Р МЭК 60870-5-104-2004 - Устройства и системы телемеханики. Часть 5. Протоколы передачи. Раздел 104. Доступ к сети для ГОСТ Р МЭК 870-5-101 с использованием стандартных транспортных профилей.



14 Qt 4.8. Профессиональное программирование на C++. — СПб.: БХВ-Петербург, 2012. — 912 с.: ил. — (В подлиннике) ISBN 978-5-9775-0736-3

15 «Конфигурирование области ввода/вывода ПЛК Овен» руководство пользователя, 2009 - 119с.: ил.

16 Таранов А.С. Методические указания к выполнению экономической части дипломного проекта для студентов специальности 220301 - Автоматизация технологических процессов и производств, Курган, издательство КГУ, 2006г. – 36с.

17 Методические указания к выполнению раздела «Безопасность и экологичность проекта» в дипломных проектах для студентов специальности 210200, Кузьмин А.П., Левашов С.П., КГУ, 2004. - 27 с.

18 Проектная документация подстанции «Левашово», ЗАО «Высоковольтный союз», 2012.

19 Безопасность жизнедеятельности. Учебник. Под ред. Э.А. Арустамова, 10-е изд., перераб. и доп. — М.: Изд-во «Дашков и К°», 2006. — 476 с

20 Федеральный закон Российской Федерации от 28 декабря 2013 г. N 426-ФЗ «О специальной оценке условий труда»

21 Экологическая безопасность электросетевых объектов. Требования при техническом обслуживании и ремонте./ ОАО «ФСК ЕЭС» - 34 с.

22 Инструкция по устройству молниезащиты зданий и сооружений. РД 34.21.122-87 /Минэнерго СССР. - М.: Энергоатомиздат, 1989. - 56 с.

23 Правила устройства электроустановок, Подготовлена ОАО «Институт Энергосетьпроект» совместно с ОАО "ВНИИЭ", Утверждена Минэнерго России, приказ от 20 июня 2003 г. N 242.

24 Федеральный закон от 22.07.2008 N 123-ФЗ (ред. от 02.07.2013) «Технический регламент о требованиях пожарной безопасности».

25 ГОСТ 31173-2003 - «Блоки дверные стальные ».

26 РД 153-34.0-49.101-2003 Инструкция по проектированию противопожарной защиты энергетических предприятий.

27 Инструкция по устройству молниезащиты зданий, сооружений и промышленных коммуникаций (СО 153-34.21.122-2003) /МИНИСТЕРСТВО ЭНЕРГЕТИКИ РОССИЙСКОЙ ФЕДЕРАЦИИ, 2003. - 31 с.

28 Методические указания к выполнению дипломного проекта для студентов специальности 210200.-Курган, КГУ, 2001г.

29 Разработка и оформление конструкторской документации радиоэлектронной аппаратуры: Справочник /Под ред. Романычевой Э.Т. М:Радио и связь, 1989.-448с.

30 ГОСТ 7.32-2001 «Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления».

31 [www.monitel.ru](http://www.monitel.ru)

## ПРИЛОЖЕНИЕ А

Текст программы контроля параметров подстанции

```
// mainwindow.h

#include "ui_mainwindow.h" // подключить определение окна;
#ifndef MAINWINDOW_H // защита от повторного включения
#define MAINWINDOW_H // заголовочного файла;
#include <QMainWindow> // подключение определения окна
#include "parameters.h" // определение функций контроля переключений
#include "logreader.h" // определение функция работы с журналом
namespace Ui { // создать в пространстве имен Ui новое окно
class MainWindow;
}
//класс MainWindow наследует функции родителя QMainWindow
class MainWindow : public QMainWindow {
    Q_OBJECT // мета-определение класса, необходимо для QT
public:
    explicit MainWindow(QWidget *parent = 0); // пустой конструктор
    void _refresh_lcd(QLCDNumber *, QString); // очистка индикаторов
    void _refresh_control(); // обновление состояния переключателей
    ~MainWindow(); // деструктор класса
    // доступные извне члены класса, интерфейс
private slots:
    void on_pushButton_clicked(); // нажата кнопка «прибавить»
    void on_pushButton_2_clicked(); // нажата кнопка «убавить»
    void on_pushButton_3_clicked(); // нажата кнопка «сбросить»
    void on_radioButton_4_clicked(); // переключение в режим «авто»
    void on_radioButton_3_clicked(); // переключение в режим «ручной»
    void on_radioButton_clicked(); // переключение «контроль секции 1»
    void on_radioButton_2_clicked(); // переключение «контроль секции 2»
    void on_modalButton_clicked(); // нажата кнопка «принять изменения»
```

```

void on_action_2_triggered(); // нажато «меню-файл-выход»
void on_action_3_triggered(); // нажато «меню-связь-ручной_режим»
void on_action_4_triggered(); // нажато «меню-связь-авто_режим»
void on_action_5_triggered(); // нажато «меню-справка-о_программе»
void on_action_triggered(); // нажато «меню-связь-связаться»
void on_action_7_triggered(); // нажато «меню-связь-дисконнект»
void on_action_10_triggered(); // нажато «блокировать управление»
void on_action_11_triggered(); // нажато «разблокировать управление»
void on_action_12_triggered(); // нажато «показать журнал»
void on_dial_valueChanged(int value); // изменено положение ручки
void on_action_8_triggered(); // нажато «очистить журнал»

// приватные члены класса, доступные только объектам класса
private:
    Ui::MainWindow *ui;
    void showEvent(QShowEvent * event); // инициализация приложения
    void closeEvent(QCloseEvent *event); // закрытие приложение
    bool tryConnect(const QString &energyObjectName, const QString
&ipAddress); // соединение с подстанцией
    Parameters param; // объект для проверки вносимых изменений
};

// mainwindow.cpp
#include "mainwindow.h" // определения класса основного окна
#include "ui_mainwindow.h" // определения графического интерфейса
#include "parameters.h" // определения функций проверки изменений
#include <QcryptographicHash> // криптографические хеш-функции
#include <QmessageBox> // функции отображения предупреждений
#include <QshowEvent> // функции обработки событий
#include "settings.h" // функции работы с файлом конфигурации
#include <QprogressDialog> // функции для отображения прогресса

```

```

#include <Qtime> // функции для работы со временем, таймерами
#include <QDebug> // функции для отладки и ведения журнала
#include "aboutdialog.h" // окно справки о программе
#include "logreader.h" // функции для работы с журналом

bool isInitialized = false; // флаг инициализации приложения

// функции задержки 100 мс, используется в функции передачи сигнала
void delay()
{
    QTime dieTime= QTime::currentTime().addMsecs(100);
    while( QTime::currentTime() < dieTime )
        QCoreApplication::processEvents(QEventLoop::AllEvents, 100);
}

// обработчик для журналирования событий
void SimpleLoggingHandler(QtMsgType type, const QMessageLogContext
&context, const QString &msg) {
    QStringList split(msg.split("|")); // разбить сообщение по символу «|»
    QString filename = split.value(0).split("").value(1); // записать имя файла
    в filename
    QString mesg = split.value(1); // записать сообщение в mesg
    QFile file(filename); // открыть файл под именем в filename
    file.open(QIODevice::Append); // открыть файл на добавление данных
    QTextStream logfile( &file ); // создать из файла поток для записи
    switch (type) { // переключение в зависимости от типа сообщения
        case QtDebugMsg: // если сообщение типа Event
            logfile << QDateTime::currentDateTime().toString() <<" Event: " <<
mesg << "\n"; // записать событие в журнал
            break;
        case QtCriticalMsg: // если сообщение типа Error
            logfile << QDateTime::currentDateTime().toString() <<" Error: " <<

```

```

mesg << "\n"; // записать ошибку в журнал
    break;
    case QtWarningMsg: // если сообщение типа Warning
        logfile << QDateTime::currentDateTime().toString() << " Warning: "
<< mesg << "\n"; // записать в журнал предупреждение
        break;
    case QtFatalMsg: // критическое сообщение, завершение работы
        logfile << "Fatal: " << mesg << "\n"; // записать ошибку
        abort(); // завершить работу приложения
    }
    file.close(); // закрыть файл
}
//конструктор класса
MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
}
//деструктор класса
MainWindow::~MainWindow() {
    delete ui; // освобождение памяти
}
// обновление состояния переключателей
void MainWindow::_refresh_control() {
    if (!param.get_control().compare("auto")){ //если тип «авто»
        ui->lineEdit_2->setText("авто"); // отобразить в окне «авто»
    }
    else { // иначе отобразить в окне тип управление «ручной»
        ui->lineEdit_2->setText("ручной");
    }
}

```

```

    }
}
// завершение работы приложения
void MainWindow::closeEvent(QCloseEvent *event)
{ // отобразить диалог подтверждения выхода
    QMessageBox::StandardButton ret;
    ret = QMessageBox::question( this, QApplication::applicationName(),
tr("Вы действительно хотите закрыть программу?"),
                                QMessageBox::Yes | QMessageBox::No ,
QMessageBox::No );
    if (ret == QMessageBox::No) {
        event->ignore(); // продолжить работу, если выход не подтвержден
    } else { // иначе записать сообщение в журнал и выйти
        qDebug() << param.get_logfile() << "|выход из программы";
    }
}
// перерисовка элемента отображение с новым значением text
void MainWindow::_refresh_lcd(QLCDNumber *lcd, QString text = NULL)
{
    lcd->repaint(); // стереть предыдущий текст
    if (text == NULL) { // если не передан текст, отобразить стандартное
значение из конфигурационного файла
        lcd->display(ui->dial->value() + param.get_delta_U());
    } else lcd->display(text); // иначе отобразить переданное значение
}
// соединиться с подстанцией energyObjectName по IP-адресуn ipAddress
bool MainWindow::tryConnect(const QString &energyObjectName, const
QString &ipAddress) {
    Networking::connect( const QString &energyObjectName, const QString
&ipAddress) // создать TCP-сокеты на прием-передачу данных

```

```

    ui->objectName->setText(energyObjectName); // отобразить в окне
    ui->ipAddress->setText(ipAddress); // имя и адрес подстанции
    return 1; // вернуться в точку вызова
}

// нажата кнопка «прибавить напряжение»
void MainWindow::on_pushButton_clicked() {
    if (!param.increase_delta_U()) { // если изменение превышает номинал
        QMessageBox::information(this, "Ошибка ввода", "Изменение
превышает номинал"); // отобразить предупреждение
        ui->dial->setValue(0); // сбросить изменения
    }
    _refresh_lcd(ui->lcdNumber); // обновить значение напряжения
}

// нажата кнопка «уменьшить напряжение»
void MainWindow::on_pushButton_2_clicked()
{
    if (!param.decrease_delta_U()) { // если изменение превышает номинал
        QMessageBox::information(this, "Ошибка ввода", "Изменение
превышает номинал"); // отобразить окно предупреждения
        ui->dial->setValue(0); // сбросить изменения
    }
    _refresh_lcd(ui->lcdNumber); // обновить значение напряжения
}

//нажата кнопка «сброс изменений»
void MainWindow::on_pushButton_3_clicked()
{
    ui->dial->setValue(0); // вернуть ручку регулировку на ноль
    param.reset_deltaU(); // сбросить изменение напряжение
    _refresh_lcd(ui->lcdNumber, "0");
}

```



```

// нажата кнопка перехода в автоматический режим управления
void MainWindow::on_radioButton_4_clicked() {
    param.set_control("auto"); // установить автоматический режим
}

// нажата кнопка перехода в ручной режим управления
void MainWindow::on_radioButton_3_clicked() {
    param.set_control("manual"); // перейти в ручной режим управления
}

// нажато переключения контроля на первую секцию
void MainWindow::on_radioButton_clicked() {
    param.toggle_section(); // переключить контролируемую секцию на 1
}

// нажато переключения контроля на вторую секцию
void MainWindow::on_radioButton_2_clicked() {
    param.toggle_section2(); // переключить контролируемую секцию на 2
}

// нажата кнопка приема изменений
void MainWindow::on_modalButton_clicked() {
    QCryptographicHash datahash(QCryptographicHash::Sha1);
    datahash.addData(ui->lineEdit->text().toUtf8()); // посчитать хеш от
введенного пароля

    QByteArray hashByteArray = datahash.result();
    if ((hashByteArray.toHex())==(param.get_password())) { // введенный
пароль совпадает с записанным в конфигурационном файле
        if (abs(param.get_delta_U()) >= 2000) { // модуль изменения
напряжения превышает 2000 В
            QMessageBox msgBox; // инициализация окна подтверждения
            msgBox.setWindowTitle("Подтверждение");
            msgBox.setText(QString("Подтвердить ввод изменения на %1
В?").arg(ui->lcdNumber->value())); // просьба ввода подтверждения

```

```

msgBox.setStandardButtons(QMessageBox::Yes| QMessageBox::No);
if(msgBox.exec() == QMessageBox::No) { // если ввод не
подтвержден

    on_pushButton_3_clicked(); // сброс изменений
    return; // возврат в основной цикл
}
}

int numTasks = 50; // счетчик для передачи сигнала
QProgressDialog progress("Идёт передача сигнала", "Отменить", 0,
numTasks, this); // отображение окна передачи сигнала
progress.setWindowTitle("Передача сигнала");
progress.setWindowModality(Qt::WindowModal);
// отображение процесса передачи сигнала
for (int i = 0; i < numTasks; i++) {
    delay();
    progress.setValue(i); // передвинуть ползунок прогресса
    if (progress.wasCanceled()) // если передача прервана нажатием
        return; // кнопки «прервать передачу», то возврат
}
progress.setValue(numTasks); // отобразить прогресс завершенным
QString strnum;
if (param.get_delta_U() != 0 || ui->dial->value() != 0) { // если было
изменено напряжение
    param.increaseU_bydelta(ui->dial->value());
    qDebug() << param.get_logfile() << "|напряжение изменено на:" <<
param.get_delta_U() + ui->dial->value(); // увеличение текущего напряжения
    qDebug() << param.get_logfile() << "|текущее напряжение:" <<
param.get_U();
    param.reset_deltaU(); //сброс изменения
    _refresh_lcd(ui->lcdNumber, "0"); // возвращение ручки на 0
}
}

```

```

    ui->dial->setValue(0);
    _refresh_lcd(ui->lcdNumber_2, strnum.sprintf("%d", param.get_U()));
} // перерисовка текущего напряжения
if (ui->radioButton_3->isChecked() || ui->radioButton_4->isChecked())
{ // если был изменен тип управления
    if (ui->radioButton_3->isChecked()) { // если изменение на ручное
        ui->radioButton_3->setCheckable(false); // разблокирование
        ui->radioButton_3->setCheckable(true); // элементов
        ui->radioButton->setDisabled(false); // изменения
        ui->radioButton_2->setDisabled(false); // в главном окне
        ui->pushButton->setDisabled(false); // программы
        ui->pushButton_2->setDisabled(false); // контроля параметров
    }
    else { // если управление изменено на авто
        ui->radioButton_4->setCheckable(false); // блокировка
        ui->radioButton_4->setCheckable(true); // элементов
        ui->radioButton->setDisabled(true); // изменения
        ui->radioButton_2->setDisabled(true); // в главном окне
        ui->pushButton->setDisabled(true); // программы
        ui->pushButton_2->setDisabled(true); // контроля параметров
    }
    _refresh_control(); // обновить тип управления
    Settings::write("control", param.get_control()); // записать текущий
тип управления в файл конфигурации
        qDebug() << param.get_logfile() << "|управление изменено на:" <<
param.get_control(); // записать в журнал текущий тип управления
    }
    // если было изменена контролируемая секция подстанции
    if (ui->radioButton->isChecked() || ui->radioButton_2->isChecked()){
        ui->radioButton->setCheckable(false); // сбросить

```

```

        ui->radioButton_2->setCheckable(false); // элементы
        ui->radioButton->setCheckable(true); //изменения
        ui->radioButton_2->setCheckable(true);

        Settings::write("section", param.current_section()); // записать в файл
        конфигурации текущую секцию управления

        ui->lineEdit_3->setText(strnum.sprintf("%d",
param.current_section())); // отобразить в главном окне контролируемую секцию

        qDebug() << param.get_logfile() << "|активная секция изменена
на:" << param.current_section(); // записать в журнал текущую контролируемую
секцию

    }
}

else { // если введен неправильный пароль

    qCritical() << param.get_logfile() << "|ввод неправильного пароля";
    QMessageBox::warning(this, "Error", "Неправильный пароль",
QmessageBox::Ok); // отобразить окно предупреждения о неправильном вводе

}

    ui->lineEdit->setText(""); // стереть неправильный пароль
}

// нажата кнопка «меню-выход»
void MainWindow::on_action_2_triggered() {
    close(); // инициализировать выход
}

// нажата кнопка переключения контролируемой секции на 1
void MainWindow::on_action_3_triggered() {
    ui->radioButton_3->setChecked(true); // установить выбор
    on_radioButton_3_clicked(); // переключить секцию на 1
}

// нажата кнопка переключения контролируемой секции на 2
void MainWindow::on_action_4_triggered() {

```

```

    ui->radioButton_4->setChecked(true); // установить выбор
    on_radioButton_4_clicked(); // переключить секцию на 2
}

// нажата кнопка «меню-помощь-о программе»
void MainWindow::on_action_5_triggered() {
    AboutDialog *dialog = new AboutDialog; // создание нового окна
    dialog->setModal(true); // установить модальный режим
    dialog->exec(); // отобразить окно «о программе»
    delete dialog; // по закрытию удалить окно, высвободим память
}

// нажата кнопка «соединиться с подстанцией»
void MainWindow::on_action_triggered() {
    qWarning() << param.get_logfile() << "|соединение установлено";
    ui->label_connect->setStyleSheet("background-image:
url(/new/prefix1/green_connect.png);"
                                   "background-repeat: no-repeat;");
} // соединиться и отобразить иконку соединения в области уведомлений

// нажата кнопка «разорвать соединение»
void MainWindow::on_action_7_triggered() {
    qCritical() << param.get_logfile() << "|нет соединения, работа в
автономном режиме"; // записать в журнал ошибку об отсутствии связи
    ui->label_connect->setStyleSheet("background-image:
url(/new/prefix1/red_connect.png);"
                                   "background-repeat: no-repeat;");
} // установить иконку отсутствия связи в области уведомления

// заблокировано управление
void MainWindow::on_action_10_triggered() {
    ui->radioButton->setDisabled(true); // блокировка
    ui->radioButton_2->setDisabled(true); // элементов

```

```

ui->radioButton_3->setDisabled(true); // изменения
ui->radioButton_4->setDisabled(true); // параметров
ui->pushButton->setDisabled(true); // главного
ui->pushButton_2->setDisabled(true); // окна
ui->pushButton_3->setDisabled(true); // программы
ui->lineEdit->setDisabled(true); // контроля
ui->modalButton->setDisabled(true); // параметров
ui->label_16->setText("Управление заблокировано");
qCritical() << param.get_logfile() << "|управление заблокировано";
ui->label_block->setStyleSheet("background-image:
url(/new/prefix1/red_lock.png);"
                                "background-repeat: no-repeat;");
} // установить иконку блокирования управления в области уведомлений
// управление разблокировано
void MainWindow::on_action_11_triggered() {
    ui->radioButton->setDisabled(false); // разблокировать
    ui->radioButton_2->setDisabled(false); // элементы
    ui->radioButton_3->setDisabled(false); // изменения
    ui->radioButton_4->setDisabled(false); // параметров
    ui->pushButton->setDisabled(false); // главного
    ui->pushButton_2->setDisabled(false); // окна
    ui->pushButton_3->setDisabled(false); // приложения
    ui->lineEdit->setDisabled(false); // контроля
    ui->modalButton->setDisabled(false); // параметров
    ui->label_16->setText("Управление доступно");
    qWarning() << param.get_logfile() << "|управление доступно";
    ui->label_block->setStyleSheet("background-image:
url(/new/prefix1/green_lock.png);"
                                "background-repeat: no-repeat;");
} // установить иконку доступности управления

```

```

// нажата кнопка «меню-журнал-показать»
void MainWindow::on_action_12_triggered() {
    LogReader *log = new LogReader; // создание нового окна
    log->setModal(true); // установка модального режима
    log->exec(); // отображение окна журнала
    delete log; // по закрытию удалить объект, высвободив память
}

//изменено положения регулятора напряжения
void MainWindow::on_dial_valueChanged(int value)
{
    int temp;

    temp = param.get_delta_U(); //записать в temp текущее изменение
    ui->lcdNumber->display(temp + ui->dial->value()); // отобразить в окне
изменения напряжения новое значение
}

// нажата кнопка «меню-журнал-очистить»
void MainWindow::on_action_8_triggered() {
    QMessageBox msgBox; // инициализация окна подтверждения ввода
    msgBox.setWindowTitle("Подтверждение");
    msgBox.setText("Вы действительно хотите очистить журнал?");
    msgBox.setStandardButtons(QMessageBox::Yes| QMessageBox::No);
    if(msgBox.exec() == QMessageBox::Yes) { // получено подтверждение
        bool result = QFile::remove(param.get_logfile()); // очистить журнал
        if (result) { // если очистка прошла успешно
            QMessageBox::information(this, "Предупреждение", "Журнал
успешно очищен!"); // отобразить окно предупреждения об очистке
            qWarning() << param.get_logfile() << "|журнал очищен"; // записать
в журнал событие об очистке
        } else { // иначе вывести ошибку очистки журнала
            QMessageBox::information(this, "Ошибка", "Журнал не очищен!");

```

```

    }
}
}

// parameters.h
#ifndef PARAMETERS_H
#define PARAMETERS_H
#include <QString>
#include "settings.h"
#define nominal 10000 // номинальное напряжение трансформатора
#define dopusk 2000 // допуск

class Parameters {
public:
    Parameters(int start_voltage=10000, int delta=0);
    bool increase_delta_U();
    bool decrease_delta_U();
    int get_delta_U();
    bool reset_deltaU();
    bool increaseU_bydelta(int delta = 0);
    int get_U();
    void toggle_section();
    void toggle_section2();
    QString get_password();
    QString get_control();
    QString get_logfile();
    QList<QString>* get_eobject();
    short current_section();
    void set_control(QString cont);
    void init_parameters_from_file();

```



```

private:
    int deltaU;
    int U;
    short section;
    QString control;
    QString password;
    QString logfile;
    QString objectname;
    QString objectaddress;
};

#endif // PARAMETERS_H

//parameters.cpp
#include <QString>
#include <QMessageBox>
#include "settings.h"
#include "parameters.h"

// конструктор, параметры — стартовый вольтаж и изменение
Parameters::Parameters(int start_voltage, int delta) {
    U = start_voltage;
    deltaU=delta;
}

// увеличение текущего delta напряжения
bool Parameters::increase_delta_U() {
    if ((U + deltaU) < nominal + dopusk) {
        deltaU += 100;
        return 1;
    } else {
        deltaU = nominal+dopusk-U;
        return 0;
    }
}

```

```

    }
}
// уменьшение delta напряжения
bool Parameters::decrease_delta_U() {
    if ((U + deltaU) > 0) {
        deltaU -= 100;
        return 1;
    } else {
        deltaU = -U;
        return 0;
    }
}
// получение delta напряжения
int Parameters::get_delta_U() {
    return deltaU;
}
// сброс изменения напряжения
bool Parameters::reset_deltaU() {
    deltaU = 0;
    return 0;
}
// увеличение напряжения на delta
bool Parameters::increaseU_bydelta(int delta) {
    U += deltaU;
    U += delta;
    return 0;
}
//получение текущего напряжения
int Parameters::get_U() {
    return U;
}

```

```

}
// переключение управляемой секции на 1
void Parameters::toggle_section() {
    section = 1;
}
// переключение управляемой секции на 2
void Parameters::toggle_section2() {
    section = 2;
}
// получить текущую секцию управления
short Parameters::current_section() {
    return section;
}
// получить пароль из файла конфигурации
QString Parameters::get_password() {
    return password;
}
// получить текущий тип управления
QString Parameters::get_control() {
    return control;
}
// установить текущий тип управления cont
void Parameters::set_control(QString cont) {
    control = cont;
}
// получить имя файла журнала
QString Parameters::get_logfile() {
    return logfile;
}
// получить список, имя подстанции и сетевой адрес

```

```

QList<QString>* Parameters::get_eobject() {
    QList<QString> *eobject = new QList<QString>();
    eobject->append(objectname);
    eobject->append(objectaddress);
    return eobject;
}

// заполнить параметры из файла конфигурации
void Parameters::init_parameters_from_file() {
    password = Settings::read("password").toString();
    control = Settings::read("control").toString();
    section = Settings::read("section").toInt();
    logfile = Settings::read("logfile").toString();
    objectname = Settings::read("objectname").toString();
    objectaddress = Settings::read("objectaddress").toString();
}

//logreader.h
#ifndef LOGREADER_H
#define LOGREADER_H
#include <QDialog>
#include <QXmlStreamWriter>
#include <QFile>
#include <QTextStream>

namespace Ui {
class LogReader;
class LogMessage;
}

// типы сообщений: событие, предупреждение, ошибка
enum MessageType

```

```
{
    Alert = 3,
    Info = 2,
    Notify = 1,
};
```

```
class LogMessage
```

```
{
    MessageType _type;
    QString _text;
```

```
public:
```

```
    LogMessage ();
    LogMessage (const QString & str);
    void fromString(const QString & str);
    QString ToHtml();
};
```

```
class LogReader : public QDialog
```

```
{
    Q_OBJECT
```

```
public:
```

```
    explicit LogReader(QWidget *parent = 0);
    ~LogReader();
```

```
private:
```

```
    Ui::LogReader *ui;
    void showEvent(QShowEvent *event);
    QList<LogMessage *> GetMessages(const QString & filename);
```

```

        QString MessagesToHtml(const QList<LogMessage *> & messages);
};

#endif // LOGREADER_H

//logreader.cpp
#include "logreader.h"
#include "ui_logreader.h"
#include <QFile>
#include <QMessageBox>
#include <QTextStream>
#include "parameters.h"

LogReader::LogReader(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::LogReader)
{
    ui->setupUi(this);
}

// инициализация окна журнала
void LogReader::showEvent(QShowEvent *event)
{
    Parameters param;
    param.init_parameters_from_file();
    QList<LogMessage *> logmessages = GetMessages(param.get_logfile());
    QString html = MessagesToHtml(logmessages);
    ui->LogBrowser->insertHtml(html);
}

LogReader::~LogReader()
{

```

```

        delete ui;
    }
    // прочитать сообщения из файла
    QList<LogMessage *> LogReader::GetMessages(const QString & filename)
    {
        QList<LogMessage *> result;
        QFile file(filename);

        if(file.open(QIODevice::ReadOnly)) {
            QTextStream stream(&file);
            QString line;
            do {
                line = stream.readLine();
                result.append(new LogMessage(line));
            } while (!line.isNull());
        }
        return result;
    }
    // преобразовать сообщения в html для вывода в окно
    QString LogReader::MessagesToHtml(const QList<LogMessage *> &
messages) {
        QString result = "";
        foreach (LogMessage * msg, messages)
        {
            result.append(msg->ToHtml());
            result.append("<br>");
        }
        return result;
    }

```

```
LogMessage::LogMessage () : _type(Notify), _text("") {  
    }
```

```
LogMessage::LogMessage (const QString & str) {  
    this->fromString(str);  
}
```

```
// выявление типа сообщения
```

```
void LogMessage::fromString(const QString & str) {  
    if (str.contains("Event")) {  
        _type = Info;  
    } else if (str.contains("Error")) {  
        _type = Alert;  
    } else {  
        _type = Notify;  
    }  
}
```

```
    _text = str.trimmed();  
}
```

```
// форматирование в html
```

```
QString LogMessage::ToHtml() {  
    QString result = "";  
    QDomStreamWriter stream(&result);  
    stream.setAutoFormatting(true);  
    stream.writeStartElement("font");  
  
    switch (_type)  
    {  
    case Alert:  
        stream.writeAttribute("color", "Red");  
        break;
```



```

    case Notify:
        stream.writeAttribute("color", "Lime");
        break;
    case Info:
        stream.writeAttribute("color", "Aqua");
        break;
    }
    stream.writeCharacters(_text);
    stream.writeEndElement();
    return result;
}

```

/settings.h

```

#ifndef SETTINGS_H
#define SETTINGS_H
#include <QSettings>

```

```

class Settings {
public:
    static QVariant read(QString value);
    static bool write(QString key, QVariant value);
};

```

```

#endif // SETTINGS_H

```

//settings.cpp

```

#include "settings.h"

```

// прочитать параметр value из файла

```

QVariant Settings::read(QString value) {
    QSettings *settings = new

```

```

QSettings("settings.conf",QSettings::NativeFormat);
    settings->setIniCodec("utf-8");
    return settings->value(value);
    delete settings;
};
// записать параметр value в файл для ключа keys
bool Settings::write(QString key, QVariant value) {
    QSettings *settings = new
QSettings("settings.conf",QSettings::NativeFormat);
    settings->setValue(key, value);
    settings->sync();
    delete settings;
    return 0;
};

```

```

//main.cpp
#include "mainwindow.h"
#include <QApplication>
// точка входа в программу
int main(int argc, char *argv[]) {
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    return a.exec();
}

```

```

//networkong.cpp
#ifndef NETWORKING_H
#define NETWORKING_H

```

```

#include <QObject>
#include <QTimer>
#include <QThread>
#include <QtNetwork/QTcpSocket>
#include <iec104_class.h>

class QIec104 : public QObject, public iec104_class {
    Q_OBJECT
public:
    explicit QIec104(QObject *parent = 0);
    ~QIec104();
    int SendCommands; // 1=поддерживать удаленное управление, 0=не
поддерживать
    QTimer *tmKeepAlive; // таймер 1 сек
    QTcpSocket *tcps; //сетевой сокет для передачи данных
    void terminate();
    void disable_connect();
    void enable_connect();
signals:
    void signal_dataIndication( iec_obj *obj, int numpoints );
    void signal_interrogationActConfIndication();
    void signal_interrogationActTermIndication();
    void signal_tcp_connect();
    void signal_tcp_disconnect();
    void signal_commandActConfIndication(iec_obj *obj);
    void signal_commandActTermIndication(iec_obj *obj);

public slots:
    void slot_tcpdisconnect(); // завершение соединения

```

```

private slots:
    void slot_tcpconnect(); // подключиться к подстанции
    void slot_tcpreadytoread(); // контроллер готов к передачи данных
    void slot_tcperror( QAbstractSocket::SocketError socketError ); //
обработка ошибок соединения
    void slot_keep_alive(); // таймер для поддержания соединения
private:
    QThread tcpThread;
    void connectTCP(); // создать tcp-соединение
    void disconnectTCP(); // завершить tcp-соединение
    int readTCP( char * buf, int szmax ); // прочитав сообщение в буфер buf
    void sendTCP( char * data, int sz ); // отправить сообщение из data
    void interrogationActConfIndication(); // подтверждение приема
команды со стороны контроллера
    void interrogationActTermIndication(); // отклонение команды
контроллером
    void commandActConfIndication( iec_obj *obj ); // пользователь
отправил команду
    void commandActTermIndication( iec_obj *obj ); // пользователь прервал
отправление команды
    void dataIndication(iec_obj *obj, int numpoints);
    bool mEnding;
    bool mAllowConnect;
};
#endif // NETWORKING_H

//networking.cpp
#include <QtWidgets/QApplication>
#include "networking.h"

```

```

QIec104::QIec104( QObject *parent ) : QObject( parent ) {
    mEnding = false;
    mAllowConnect = true;
    SendCommands = 0;
    mLog.activateLog();
    mLog.doLogTime();
    tcps = new QTcpSocket();
    tmKeepAlive = new QTimer();
    connect( tcps, SIGNAL(readyRead()), this, SLOT(slot_tcpreadytoread()) );
    connect( tcps, SIGNAL(connected()), this, SLOT(slot_tcpconnect()) );
    connect( tcps, SIGNAL(disconnected()), this, SLOT(slot_tcpdisconnect()) );
    connect( tcps, SIGNAL(error(QAbstractSocket::SocketError)), this,
SLOT(slot_tcperror(QAbstractSocket::SocketError)),Qt::DirectConnection );
    connect( tmKeepAlive, SIGNAL(timeout()), this, SLOT(slot_keep_alive()) );
    tcps->moveToThread( &tcpThread );
    tcpThread.start( QThread::TimeCriticalPriority );
}
QIec104::~~QIec104()
{
    delete tmKeepAlive;
    delete tcps;
}
void QIec104::dataIndication( iec_obj *obj, int numpoints ) {
    emit signal_dataIndication( obj, numpoints );
}
void QIec104::connectTCP() {
    tcps->close();
    if ( !mEnding && mAllowConnect )
        tcps->connectToHost( getSecondaryIP(), getPortTCP(),
QIODevice::ReadWrite );
}

```

```

}

void QIec104::disconnectTCP() {
    tcps->close();
}

void QIec104::slot_tcperror( QAbstractSocket::SocketError socketError ) {
    if ( socketError != QAbstractSocket::SocketTimeoutError ) {
        char buf[100];
        sprintf( buf, "SocketError: %d", socketError );
        mLog.pushMsg( (const char *)buf );
    }
}

int QIec104::readTCP( char * buf, int szmax ) {
    if (!mEnding)
        return tcps->read( buf, szmax );
    else
        return 0;
}

void QIec104::sendTCP( char * data, int sz ) {
    if ( tcps->state() == QAbstractSocket::ConnectedState )
        if ( !mEnding )
            tcps->write( data, sz );
}

// обработка только что созданного соединения
void QIec104::slot_tcpconnect() {
    tcps->setSocketOption( QAbstractSocket::LowDelayOption, 1 );
    onConnectTCP();
    emit signal_tcp_connect();
}

void QIec104::slot_tcpdisconnect() {
    onDisconnectTCP();
}

```

```

        emit signal_tcp_disconnect();
    }
    void QIec104::slot_keep_alive() {
        if ( !mEnding )
            onTimerSecond();
    }
    void QIec104::interrogationActConfIndication() {
        emit signal_interrogationActConfIndication();
    }
    void QIec104::interrogationActTermIndication() {
        emit signal_interrogationActTermIndication();
    }
    void QIec104::commandActConfIndication( iec_obj *obj ) {
        emit signal_commandActConfIndication( obj );
    }
    void QIec104::commandActTermIndication( iec_obj *obj ) {
        emit signal_commandActTermIndication( obj );
    }
    void QIec104::terminate() {
        mEnding = true;
        tcps->close();
        tcpThread.quit();
        tcpThread.wait( 1000 );
        if ( tcpThread.isRunning() )
            tcpThread.terminate();
        if ( tcpThread.isRunning() )
            tcpThread.wait( 2000 );
    }
    void QIec104::slot_tcpreadytoread() {
        if ( tcps->bytesAvailable() < 6 ) return;

```

```

packetReadyTCP();
int cnt = 0;
while ( (tcps->bytesAvailable() > 5) && (cnt++ < 10) ) {
    tcps->waitForReadyRead( 10 );
    packetReadyTCP();
}
}

void QIec104::disable_connect() {
    mAllowConnect = false;
    if ( tcps->state() == QAbstractSocket::ConnectedState )
        disconnectTCP();
}

void QIec104::enable_connect() {
    mAllowConnect = true;
}

```