

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»

Профиль подготовки: «Вычислительные машины, комплексы, системы и
сети»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
БАКАЛАВРА

на тему

СИСТЕМА АВТОМАТИЧЕСКОЙ
ГЕНЕРАЦИИ МОДЕЛЕЙ ЦИФРОВЫХ
ПОДСТАНЦИЙ МЭК 61850

Студент РМ 31.05.2022 Ручкин Михаил Алексеевич
(подпись, дата) (ФИО полностью)

Руководитель ВДН 31.05.2022 Дубинин В. Н.
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр Ант 02.06.22 Бычков А. С.
(подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 02.06.2022 г. № 13)

Заведующий кафедрой М М.А. Митрохин
(подпись) (фамилия, инициалы)

Работа защищена с отметкой Отлично (протокол заседания ГЭК от
07.06.2022 г. № 5)

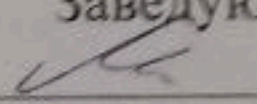
Секретарь ГЭК Е Е.И. Гурин
(подпись) (фамилия, инициалы)

Пенза 2022

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
 М.А. Митрохин
«26» апреля 2022г.

**ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ**

1. Студент Ручкин Михаил Алексеевич гр. 18ВВ1 факультета вычислительной техники
2. Тема работы «Система автоматической генерации моделей цифровых подстанций МЭК 61850»

Тема утверждена приказом ПГУ № 357/о от "26" апреля 2022г.

3. Руководитель бакалаврской работы д.т.н. профессор кафедры ВТ ПГУ Дубинин В.Н.

4. Задание на работу

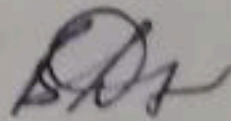
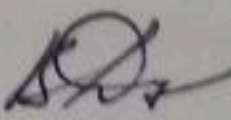
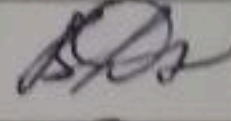
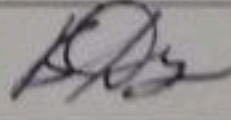
Разработать методики и программную систему, позволяющую автоматически генерировать MATLAB/SIMULINK модели и функционально-блочные модели распределенного управления на основе SCL-описаний цифровых подстанций МЭК 61850. В работе использовать следующие типы оборудования цифровых подстанций МЭК 61850: защитный выключатель, измеритель тока, заземление, выключатель, конденсатор, батарея, преобразователь переменного тока в постоянный, преобразователь постоянного тока в переменный, нагрузка.

Функционально-блочные модели должны быть построены в соответствии со стандартом МЭК 61499. В качестве среды разработки функциональных блоков МЭК 61499 использовать систему nxtSTUDIO.

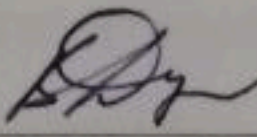
5. Перечень подлежащих разработке вопросов

- 1) Обзор и анализ методов проектирования цифровых подстанций МЭК 61850
- 2) Разработка методики автоматической генерации MATLAB/SIMULINK моделей цифровых подстанций МЭК 61850
- 3) Разработка методики автоматической генерации функционально-блочных моделей распределенного управления цифровых подстанций МЭК 61850
- 4) Разработка программных средств для поддержки методик автоматических генераций

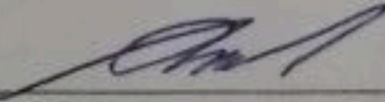
6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Обзор и анализ методов проектирования цифровых подстанций МЭК61850	15%	26.04.22-29.04.22	
2	Разработка методики автоматической генерации MATLAB/SIMULINK моделей цифровых подстанций МЭК61850	20%	30.04.22-6.05.22	
3	Разработка методики автоматической генерации функционально-блочных моделей распределенного управления цифровых подстанций МЭК61850	20%	7.05.22-13.05.22	
4	Разработка программных средств для поддержки методик автоматических генераций	25%	14.05.22-22.05.22	
5	Тестирование и отладка	10%	22.05.22-26.05.22	
6	Оформление выпускной работы.	10%	26.05.22-31.05.22	

Дата выдачи задания "26" апреля 2022 г.

Руководитель бакалаврской работы  26.04.2022 Дубинин В.Н.
(подпись, дата)

Задание к исполнению принял студент  26.04.2022 Ручкин М.А.
(подпись, дата)

Работу к защите допустить
Декан факультета  2.06.22 Фионова Л.Р.
(подпись, дата)

Реферат

Работа содержит 110 листов, 31 рисунок, 5 таблиц, 11 источников, 3 приложения.

МОДЕЛИРОВАНИЕ, МЭК 61850, ЦИФРОВЫЕ ПОДСТАНЦИИ, *SCL*, *MATLAB*, МЭК 61499, ФУНКЦИОНАЛЬНЫЕ БЛОКИ.

Объектом исследования являются цифровые подстанции МЭК 61850.

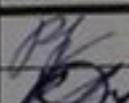
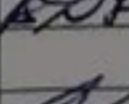
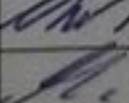
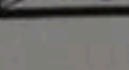
Цель ВКР – разработка программной системы автоматической генерации *MATLAB/SIMULINK*- и функционально-блочных моделей на основе *SCL*-описаний цифровых подстанций МЭК 61850.

В процессе работы выполнялся обзор и анализ существующих публикаций, генерировались *MATLAB/SIMULINK*- и функционально блочные модели цифровых подстанций МЭК 61850.

Основные результаты работы:

- разработана методика автоматической генерации *MATLAB/SIMULINK* моделей цифровых подстанций МЭК 61850;
- разработка методики автоматической генерации функционально-блочных моделей распределенного управления цифровых подстанций МЭК 61850 на основе стандарта МЭК 61499;
- разработана программная система для поддержки методик автоматической генерации.

Использованный в работе подход может быть использован при проектировании и реализации систем управления цифровыми подстанциями МЭК 61850 на основе стандарта МЭК 61499.

<h3 style="margin: 0;">ПГУ 1.09.03.01.35.001 ПЗ</h3>										
Изм.	Лист	№ докум.	Подпись	Дата						
Разраб		Ручкин М.А.								
Пров		Дубинин В.Н.								
Н. Контр.		Бычков А.С.								
Утв		Митрохин М.А.								
Система автоматической генерации моделей цифровых подстанций МЭК 61850 Пояснительная записка										
<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 10%;">Лит.</td> <td style="width: 10%;">Лист</td> <td style="width: 10%;">Листов</td> </tr> <tr> <td></td> <td style="text-align: center;">2</td> <td style="text-align: center;">110</td> </tr> </table>					Лит.	Лист	Листов		2	110
Лит.	Лист	Листов								
	2	110								
ФВТ, каф. ВТ гр. 18ВВ1										

Содержание

Введение	5
1 Система автоматической генерации моделей цифровых подстанций МЭК 61850	7
1.1 Обзор и анализ методов проектирования цифровых подстанций МЭК 61850	7
1.1.1 Сети электроснабжения <i>Smart Grid</i>	7
1.1.2 Стандарт МЭК 61850 для спецификации цифровых подстанций	9
1.1.3 Программа <i>MATLAB</i> для имитационного моделирования систем электроснабжения	12
1.1.4 Стандарт МЭК 61499 для построения распределенных систем управления	14
1.1.5 Модельно-ориентированный подход к проектированию	17
1.2 Методика автоматической генерации <i>MATLAB/SIMULINK</i> моделей цифровых подстанций МЭК 61850	18
1.2.1 Описание <i>MATLAB/SIMULINK</i> моделей	18
1.2.2 Основные положения методики	21
1.2.3 Разработка правил трансформаций исходной модели в целевую модель	21
1.2.3.1 Графовые правила трансформации	21
1.2.3.2 Правила трансформации <i>SCL</i> описания в сценарий <i>MATLAB</i>	23
1.3 Методика автоматической генерации функционально- блочных моделей распределенного управления для цифровых подстанций МЭК 61850	25
1.3.1 Основные положения методики	25
1.3.2 Разработка функциональных блоков	25
1.3.3 Алгоритм трансформации <i>SCL</i> описания в функциональные	

блоки стандарта МЭК 61499	28
1.4 Разработка программных средств для поддержки методик	
автоматической генерации	30
1.4.1 Основные структуры данных	30
1.4.2 Алгоритмы программы	32
1.4.3 Архитектура программных средств	34
1.4.4 Руководство пользователя	35
1.4.5 Тестирование и отладка	38
2 Безопасность жизнедеятельности	46
2.1 Введение	46
2.2 Требования к помещениям для работы с ПЭВМ	48
2.3 Требования к освещению на рабочих местах, оборудованных	
ПЭВМ	49
2.4 Определение нормативных уровней освещенности	52
2.5 Проект электрического освещения помещения при эксплуатации	
ПЭВМ	52
Заключение	57
Список использованных источников	58
Приложение А – Листинг программы	60
Приложение Б – Схемы программ	97
Приложение В – Презентация	102

Введение

Smart Grid и цифровые подстанции – это главные компоненты энергетических комплексов будущего [1]. Основные преимущества *Smart Grid* над традиционной системой электроснабжения:

- интеллектуализация сети;
- способность самоконтроля и предоставление информации о любом участнике сети;
- высокая надежность сети. Даже при переключении источника, пользователь не заметит сбоев;
- повышение эффективности за счёт уменьшения потери в проводах и оптимального распределения нагрузки.

Цифровая подстанция – это термин, применяемый к электрическим подстанциям, где управление работой осуществляется между распределенными интеллектуальными электронными устройствами, соединенными сетями передачи данных. Цифровая подстанция являются компонентом сетей *Smart Grid*, а также являются киберфизическими системами так как структурно их можно разделить на две части физическую (электрооборудование цифровой подстанции), и кибернетическую (система управления цифровой подстанции). Киберфизические системы являются атрибутом четвертой промышленной революции (Индустрия 4.0).

Разработка, ввод в эксплуатацию и эксплуатацию таких сложных систем облегчается с помощью цифровых двойников, которые представляют собой имитационные модели систем, вычисляющие параметры текущего состояния с учетом фактической истории изменения, режима и условий работы систем. Концепция цифровых двойников, начавшая быть популярной в 2015 году, стала еще более популярной после начала пандемии *COVID-19*, когда многие люди были вынуждены работать из дома [2]. Однако разработка цифровых двойников, несмотря на их большую полезность, представляет собой сложную инженерную задачу. Чтобы уменьшить усилия,

его разработка должна быть лучше связана с инженерными процессами развития центров обработки данных.

Исходя из вышесказанного, тема ВКР посвящённая автоматизации проектирования моделей цифровых подстанций МЭК 61850 (по сути дела, являющихся их цифровым двойником) является актуальной.

Цель работы – разработка методик и программной системы автоматической генерации *MATLAB/SIMULINK*- и функционально-блочных моделей на основе *SCL*-описаний цифровых подстанций МЭК 61850.

Первая методика должна позволять генерировать *MATLAB/SIMULINK* модели, представляющие собой физическую часть системы. А вторая - генерировать функционально-блочные модели, представляющие кибернетическую часть системы, для поддержки распределенного управления.

1 Система автоматической генерации моделей цифровых подстанций МЭК 61850

1.1 Обзор и анализ методов проектирования цифровых подстанций МЭК 61850

1.1.1 Сети электроснабжения *Smart Grid*

Smart Grid - интеллектуальная электрическая сеть, в которой существует связь между всеми участниками энергетического рынка, направленная на предоставление энергетических услуг, снижение затрат и повышение эффективности, а также на интеграцию распределенных источников энергии, в том числе возобновляемых источников энергии.

Это электроэнергетические и коммуникационные сети, которые позволяют регулировать производство и потребление электроэнергии в режиме реального времени как на местном, так и на глобальном уровне. Их принцип - интерактивное двустороннее общение между производственными источниками и приборами или потребителями о текущих возможностях производства и потребления энергии.

По мере увеличения доли возобновляемых источников энергии в электрических сетях возрастают и требования, в частности, к сетям низкого напряжения. Сети низкого напряжения выполняют не только задачу приема и распределения электроэнергии из распределительных сетей, но и все чаще выполняют задачу подачи децентрализованно произведенной электроэнергии обратно в распределительные сети.

Вторжение возобновляемых источников энергии в энергетический ландшафт заметно изменило потоки энергии в электросети: теперь пользователи не только потребляют, но и производят электроэнергию через ту же самую сеть. Следовательно, поток энергии теперь двунаправленный.

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

7

Интеллектуальная сеть отправляет электроэнергию от поставщиков к потребителям, используя двунаправленную цифровую технологию для управления потребностями потребителей. Это помогает сэкономить электроэнергию, снизить затраты, повысить удобство использования и прозрачность.

Характеристики умных сетей электроснабжения:

- полная автоматизация. Она включает в себя цифровую систему мониторинга и управления, встроенные датчики, отслеживающие поведение сети и автоматическое восстановление после сбоев. Обеспечивается доступность информации в режиме реального времени о загрузке сети, качестве электроснабжения, перебоях и т. д;

- полная интеграция клиентов. Ее суть состоит в том, чтобы снабдить потребителей цифровыми счетчиками двусторонним потоком информации в реальном времени, что позволяет создавать ценовые тарифы в соответствии с текущей ситуацией в сети (так называемые «интеллектуальные счетчики электроэнергии»). Это позволяет клиентам эффективно управлять потреблением, таким как нагрев воды, стирка или подзарядка аккумуляторов;

- адаптация к различным способам производства электроэнергии. Развитие интеллектуальных сетей началось в основном в то время, когда к распределительной системе начали подключаться более мелкие источники электроэнергии, поведение которых трудно предсказать. Например, солнечные и ветряные электростанции, газовые микротурбины и другие децентрализованные технологии производства электроэнергии, дающие потребителям возможность вырабатывать электроэнергию из собственных ресурсов и продавать ее излишки в сеть. Солнечный свет или силу ветра очень трудно предсказать везде, где расположены эти децентрализованные источники. Благодаря тому, что интеллектуальные сети регулярно отправляют информацию о потреблении электроэнергии, можно

регулировать мощность, например, на электростанциях, использующих традиционные виды топлива.

Основной особенностью *Smart Grid* является децентрализация системы управления, это обусловлено необходимостью управления сетью в больших масштабах (возобновляемые источники энергии распределены на достаточно большой территории, ввиду чего невозможно их централизованное управление).

Проектирование сетей таких масштабов имеет множество проблем, таких как обеспечение безопасности передачи данных, надежности, масштабируемости и технического обслуживания. Интеллектуальная сеть имеет проблемы не только при передаче данных и при взаимодействии отдельных компонентов сети, но и при обмене информацией между физической и кибернетической частью. Главной их причиной является высокая сложность и постоянно меняющаяся структура основных физических систем в течении жизненного цикла *Smart Grid*, что приводит к перестройке и повторному тестированию кибернетической части, которая является наиболее сложным элементом сети. Для контроля работы сети был введен стандарт МЭК 61850, в котором описаны правила совместимости протоколов систем автоматизации подстанции в виде множества объектно-ориентированного представления данных.

1.1.2 Стандарт МЭК 61850 для спецификации цифровых подстанций

МЭК 61850— это международный стандарт связи между различными типами электрооборудования, который изначально был разработан для работы внутри зоны автоматизации подстанции [3]. Он определяет множество функций, таких как защита, управление и измерение. По сравнению с другими стандартами и протоколами связи МЭК 61850 имеет две отличительные характеристики, которые подчеркивают привлекательность и интерес к этому стандарту, а именно:

- высокие требования ко времени обмена данными;

- высокие требования к надежности обмена данными.

Наиболее важные особенности структуры данных МЭК 61850 включают в себя:

- моделирование данных. Первичные объекты, а также функции защиты и контроля моделируются как стандартные логические узлы, которые могут быть сгруппированы в рамках различных логических устройств;

- схемы отчетности. Существует несколько схем отчетности (*BRCB* и *URCB*) для передачи данных с сервера клиенту посредством взаимодействия между сервером и клиентом, которые можно активировать на основе заранее определенных обстоятельств запуска;

- быстрая передача событий. Общие события подстанции (*GSE*) определены для быстрой передачи данных о событиях в режиме одноранговой связи. *GOOSE* и *GSSE* снова разделены;

- настройка групп. Блоки управления группой настроек (*SGCB*) предназначены для обработки групп настроек, позволяя пользователю переключаться на любую активную группу в зависимости от его потребностей;

- передача выборочных данных. Блоки управления выборочными значениями используются в схемах для обработки передачи выборочных значений (*SVCB*);

- хранение данных. Язык конфигурации подстанции (*SCL*) — это спецификация для хранения всех сконфигурированных данных подстанции в стандартизированном формате.

Одним из важнейших преимуществ использования МЭК 61850 является его каталог логических объектов. Зная тип элемента (например, *VTR*, *CBR* или *CTR*), можно достоверно определить его назначение для дальнейших преобразований. Например, зная, что некоторая система имеет элементы «*CBR*» и «*CON*», можно сделать вывод, что эта система имеет

функцию преобразования энергии (элемент типа «*CON*») и токозащитные функции (элемент типа «*CBR*»).

Еще одним преимуществом использования МЭК 61850 является язык конфигурации подстанции (*SCL*), который описан в формате *XML*. Поскольку это *XML*, вся внутренняя структура файла представлена, как связность нескольких узлов с разными свойствами. Подобно физическому представлению любого элемента, *SCL* стандарт МЭК 61850 имеет специальные параметры и атрибуты, используемые для описания соединений между отдельными компонентами. Например, ключевое слово «*Terminal*» определяет новое физическое соединение с параметром «*connectivityNode*», который используется для определения описания физического порта назначения. В качестве примера используется физическое соединения двух компонентов автоматического выключателя (изображенных на рисунке 1), их *SCL* описание представлено на рисунке 2. Каждый новый элемент структуры в узле «*ConductingEquipment*», имеет параметры «*name*», «*type*», «*sxy:x*» и «*sxy:y*» («*sxy:x*» и «*sxy:y*» несут информацию о позиции элемента). В нем также есть подузел «*Terminal*» с дополнительной информацией о связи между компонентами. Как видно из рисунка 2, объекты МЭК 61850 не имеет идентификации для входных/выходных соединений, но вместо этого они имеют уникальные идентификационные метки для подключенных проводов (например, имя провода «*MFS/J/01/L1*» на рисунке 2).

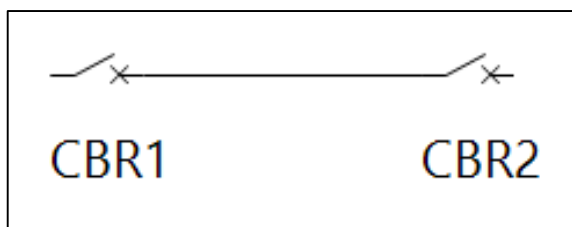


Рисунок 1- Пример цепи из двух выключателей *CBR1* и *CBR2*
(графическое представление)

```

<ConductingEquipment
  name="CBR1"
  sxy:x="22"
  type="CBR">
  <Terminal
    bayName="01"
    cNodeName="L1"
    substationName="MFS"
    voltageLevelName="J"
    connectivityNode="MFS/J/01/L1"/>
  </ConductingEquipment>
<ConductingEquipment
  name="CBR2"
  sxy:x="26"
  type="CBR">
  <Terminal
    bayName="01"
    cNodeName="L1"
    substationName="MFS"
    voltageLevelName="J"
    connectivityNode="MFS/J/01/L1"/>
  </ConductingEquipment>

```

Рисунок 2- Пример цепи из двух выключателей *CBR1* и *CBR2* (XML-представление)

1.1.3 Программа *MATLAB* для имитационного моделирования систем электроснабжения

Моделирование применяется для познания свойств оригинала путем исследования не самого объекта, а его модели. Моделирование оправдано в том случае, если создание модели проще, чем создание оригинала, кроме того, применительно к системам электроснабжения оно позволяет моделировать режимы и ситуации, крайне нежелательные для реальных систем (короткие замыкания, провалы напряжения, перенапряжения и т.п.).

В принципе, в зависимости от способа реализации моделей их можно разделить на физические модели и математические модели.

Физические модели представляют собой идентичные оригиналу устройства уменьшенных габаритов, которые имеют ту же физическую природу. Они наиболее точно описывают поведение реального объекта, но

варьирование их параметрами затруднено, а создание самой модели очень дорого.

Математические модели представляют собой описания объекта в виде математических соотношений. Математическая модель, описывающая поведение объекта во времени, называется имитационной моделью.

Имитационное моделирование является одним из видов компьютерного моделирования. При этом под имитацией понимают проведение на компьютерах различных экспериментов с моделями, которые представлены в качестве некоторого набора компьютерных программ. В настоящее время имеется значительное количество программных комплексов, позволяющих с помощью последовательности вычислений и графического отображения их результатов воспроизводить (имитировать) процессы функционирования объекта при условии воздействия на него различных факторов. Среди систем компьютерного моделирования особо выделяется программа *MATLAB* (матричная лаборатория), ориентированная в первую очередь на научно-технические вычисления и моделирование.

Программа *MATLAB* представляет собой систему и язык программирования высокого уровня для инженерных и научных вычислений и создания средств моделирования для различных устройств и систем [4].

Моделирование систем электроснабжения возможно, как с помощью создания своей программы на языке *MATLAB*, так и путем использования моделей типовых элементов системы электроснабжения из пакета расширения *SIMULINK*.

Пакет расширения *SIMULINK* программы *MATLAB* позволяет выполнить имитационное моделирование объектов, состоящих из графических блоков с заданными параметрами. В свою очередь в пакет *SIMULINK* входит крупный пакет расширения *SimPowerSistems*, предназначенный специально для моделирования электротехнических устройств и энергетических систем большой мощности. В модель

энергетической системы могут включаться блоки источников, элементов, виртуальных регистрирующих приборов, графические средства анимации. Приложение *SIMULINK* является своего рода «виртуальной лабораторией», позволяющей собирать и исследовать работу многих видов электрических цепей и устройств. Получившаяся в результате «виртуальная схема» позволяет проводить все этапы исследования, присущие экспериментальному или опытному образцу.

В настоящее время существует несколько исследовательских работ, где сети *Smart Grid* моделируются с использованием инструментальной среды *MATLAB/SIMULINK*. В работе [5] модель сети *Smart Grid* реализованной с помощью *MATLAB Simulink* используется для динамического исследования новой модели гибридной системы *PV/WT*, подключенной к интеллектуальной сети. В работе [6] представлена реализация системы управления интеллектуальной сетью с использованием языка программирования *MATLAB*, которая демонстрирует передачу данных интеллектуального учета от домов к коммунальным предприятиям и иллюстрирует контроль коммунальных служб над потреблением пользователя. В работе [7] показано, как эти архитектуры в сочетании с другими передовыми технологиями, такими как унифицированный язык моделирования, *SIMULINK* и сетевые системы условий/событий, формируют структуру, которая позволяет выполнять проектирование, моделирование, формальную проверку и развертывание с поддержкой набора программных средств.

1.1.4 Стандарт МЭК 61499 для построения распределенных систем управления

МЭК 61499 — это международный стандарт для систем измерения и управления промышленными процессами [8]. Впервые он был выпущен в 2005 году. МЭК 61499 допускает проект, ориентированный на приложения, при котором одно или несколько приложений создаются для всей системы, а затем развертываются на доступных устройствах, определяемых сетями

взаимосвязанных функциональных блоков. Модель устройства описывает все устройства в системе. Модель системы отражает топологию системы. В рамках модели сопоставления описывается распределенные приложения. В результате системные приложения являются распределенными, но должны поддерживаться вместе.

МЭК 61499 вводит функциональные блоки, которые являются артефактами объектно-ориентированного проектирования (Рисунок 3). Входные события используются для определения логических выражений, которые активируют переходы диаграммы управления выполнением функционального блока (ЕСС) (Рисунок 4). Согласно работе [9] МЭК 61499 значительно упрощает конструкцию и отделяет логику реконфигурации от моделей управления. В МЭК 61499 это было достигнуто с помощью иерархии в конструкции. Используя иерархический подход конечного автомата, функциональные блоки обрабатывает события реконфигурации и немедленно переключается на соответствующую конфигурацию. Последняя представляет собой модель реконфигурации, которая реагирует на изменения в среде, чтобы обнаружить подходящий сценарий реконфигурации для выполнения. Каждый сценарий предоставляет уникальный набор алгоритмов, заключенных в отдельную ведомую схему управления выполнением, представляющую модель управления функциональным блоком.

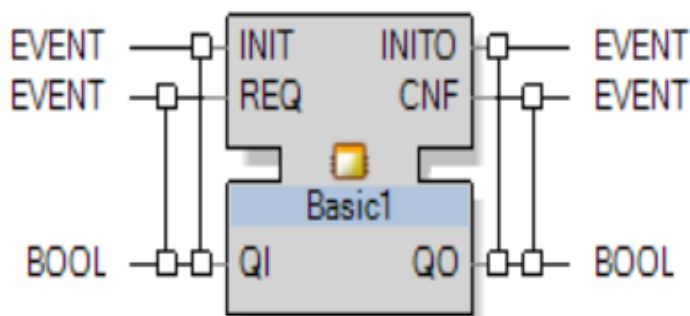


Рисунок 3 - Функциональный блок

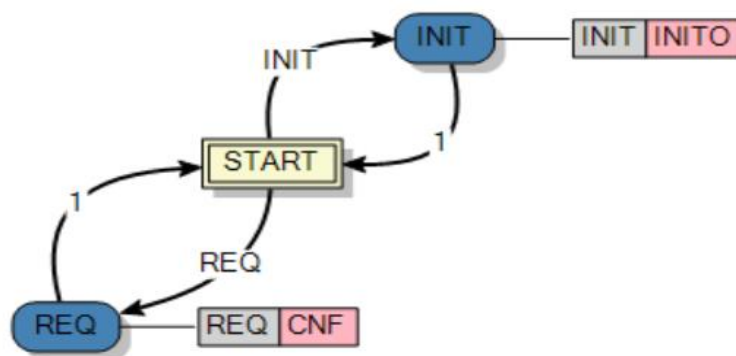


Рисунок 4 - Пример диаграммы ECC

Как видно на рисунке 5, графическое представление всех функциональных блоков можно разделить на 4 части подробное описание которых представлено в таблице 1.

Таблица 1 - Описание структуры интерфейса функционального блока

Часть интерфейса функционального блока	Описание
Входные события	Каждый раз, когда приходит входное событие, оно запускает реализацию некоторых функций, связанных с этим событием
Выходные события	Новое событие срабатывает каждый раз, когда вычисляется какая-то конкретная функция, связанная с этим событием
Входные данные	Входные данные используются для изменения поведения predetermined внутренней архитектуры функционального блока. Входные данные будут обновляться каждый раз, когда наступит новое входное событие, связанное с этим входом (или группой входов).

Часть интерфейса функционального блока	Описание
Выходные данные	Выходные данные используются для взаимодействия с системой вне функционального блока. Выходные данные будут обновляться каждый раз, когда будет срабатывать новое выходное событие, связанное с тем же выходом (или группой выходов).

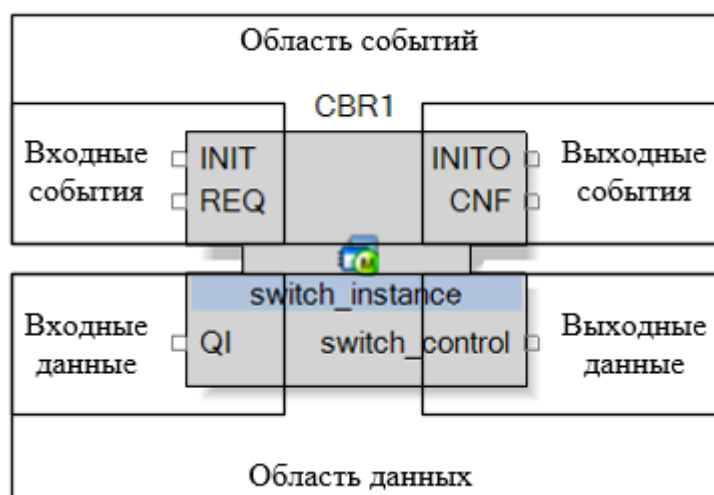


Рисунок 5 - Пример интерфейса функционального блока

1.1.5 Модельно-ориентированный подход к проектированию

Модельно-ориентированный подход к проектированию (*Model-Driven Engineering*) широко используется в разработке программного обеспечения, поскольку она доказала свою эффективность при разработке приложений большого масштаба. Недавно *MDE* нашла свой путь в отрасли автоматизации. *MDE* следует мантре «все это модель», где парадигмы проектирования сосредоточены вокруг модели, будь то модели, зависящие от платформы модели или независимые [10]. МЭК 61850 может быть рассмотрено как решение *MDE* для систем SAS с введением на подстанциях

языка программирования *SCL*. Язык *SCL* представлен как *XML*-документ, и имеется в общей сложности шесть типов документов *SCL*, каждый документ описывающих различную стадию процесса проектирования *MDE*. Например, системная спецификация *System Specication Description (SSD)* используется на начальном этапе проектирования для определения формальной спецификаций системы, которая включает в себя функциональные и физические требования, представленные как логические узлы. Общая конструкция всей системы называется *Substation Conguration Description (SCD)*, которое содержит общую конфигурацию системы, включая конфигурацию связи, логические узлы и *IED*.

Существуют исследовательские работы, в которых также используется язык *SCL* в качестве основы для автоматической генерации систем управления, которая является одним компонентом разработки *MDE*. Архитектура *SysGrid* в [11] является *MDE*-ориентированной архитектурой, которая использует *SCL*-конфигурации в качестве исходных моделей *MDE* для моделирования МЭК 61850/61499.

1.2 Методика автоматической генерации *MATLAB/SIMULINK* моделей цифровых подстанций МЭК 61850

1.2.1 Описание *MATLAB/SIMULINK* моделей

Модель *MATLAB/SIMULINK* в данной работе используется в качестве основы для математических расчетов физических процессов, происходящих в цифровых подстанциях. Конструкцией модели *SIMULINK* можно управлять вручную или удаленно (например, через веб-интерфейс). Процедуры для простого ручного проектирования проекта *SIMULINK* в *MATLAB* следующие:

- построение модели;
- размещение компонентов;
- определите атрибуты компонента;
- установление межкомпонентных связей;

- определение атрибутов «решателя» и сохранение модели.

Используя язык программирования *MATLAB*, идентичные процедуры можно выполнить в сценарии *MATLAB*. Сценарий, изображенный на рисунке 6, создает простую модель (показана на рисунке 7), которая практически не имеет смысла, но в то же время полезна для понимания основных вещей программирования в *MATLAB*.

```
function SCD_2_MATLAB
fname = 'SCD_2_MATLAB';
% Check if the file already exists and delete it if it does
if exist(fname,'file') == 4
    % If it does then check whether it's open
    if bdlsLoaded(fname)
        % If it is then close it (without saving!)
        close_system(fname,0)
    end
    % delete the file
    delete([fname,'.slx']);
end
% Create the system
new_system(fname);
(1) {
(2) {
(3) {
(4) {
(5) {
    add_block('powerlib/Measurements/Three-Phase V-I Measurement', [gcs,'/CT1'],...
    'VoltageMeasurement', 'no',...
    'CurrentMeasurement', 'yes');
    add_block('powerlib/Elements/Three-Phase Breaker', [gcs,'/CB2'],...
    'InitialState', 'open')
    add_line(gcs, 'CB2/RConn1', 'CT1/LConn1');
    add_line(gcs, 'CB2/RConn2', 'CT1/LConn2');
    add_line(gcs, 'CB2/RConn3', 'CT1/LConn3');
    add_line(gcs, 'CT1/RConn1', 'CB2/LConn1');
    add_line(gcs, 'CT1/RConn2', 'CB2/LConn2');
    add_line(gcs, 'CT1/RConn3', 'CB2/LConn3');
    'Solver','FixedStepDiscrete',...
    'FixedStep','0.1');
    % Save the model
    save_system(fname);

```

Рисунок 6 - Пример сценарий создания простой *MATLAB/SIMULINK* модели

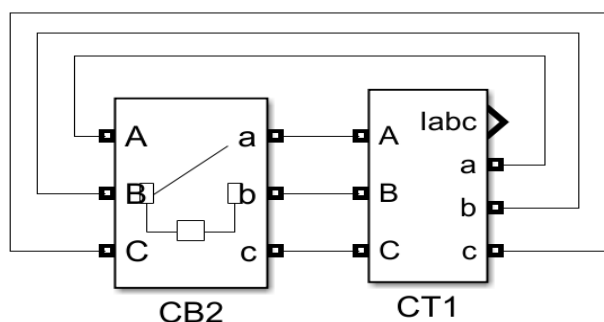


Рисунок 7 - Сгенерированная SIMULINK модель

Рисунок 8 можно использовать как пример графического представления компонента *MATLAB/SIMULINK*. Каждый порт объектов *MATLAB/SIMULINK* имеет уникальное идентификационное имя. Через эти имена можно связаться с каждым портом непосредственно из командного окна *MATLAB*. Компонент трехфазного прерывателя тока, представленный на рисунок 8, имеет 6 контактов (*A-C* и *a-c*). Все контакты имеют уникальное идентификационное имя «*XConnY*», где «*X*» — это определение стороны компонента (а именно левой и правой) и может быть «*L*» или «*R*», а «*Y*» — номер порта (с наименьшим возможным значением, равным 1). Например, порт «*A*» компонента «*CBR1*» на рисунке 8 имеет имя «*RConn1*», порт «*c*» имеет имя «*LConn3*» и т. д. Сценарий *MATLAB* представлен на рисунке 9.

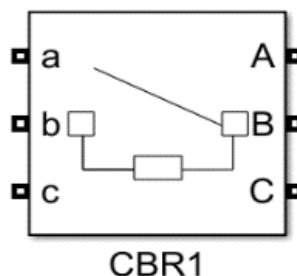


Рисунок 8 - Трехфазный переключатель в системе *MATLAB*

```
% Name of the model
fname = 'SCD_2_MATLAB';

% Create the system
new_system(fname);
add_block(...
'powerlib/Elements/Three-Phase Breaker',...
[gcs, '/CBR1'],...
'InitialState', 'open',...
'Position', [240 100 270 130]);

% Set a couple of model parameters
%to eliminate warning messages
set_param(gcs,...
'Solver', 'FixedStepDiscrete',...
'FixedStep', '0.1');

% Save the model
save_system(fname);
```

Рисунок 9 – Генерация трехфазного переключателя с использование функциональных возможностей сценария *MATLAB*

1.2.2 Основные положения методики

Методика автоматической генерации *MATLAB/SIMULINK* моделей базируется на подробном каталоге описания типов оборудования стандарта МЭК 61850. В ходе генерации *MATLAB/SIMULINK* модели можно выделить два этапа:

- замена оборудования цифровой подстанции МЭК 61850 на соответствующий элемент библиотеки *MATLAB*, который является функциональным аналогом заменяемого оборудования;
- добавление связей между элементами *MATLAB* в соответствии с линиями связи между оборудованием цифровой подстанции. На данном этапе возникают сложности связанные с тем, что стандарт МЭК 61850 не предоставляет достаточных сведений для проведения корректных связей между элементами *MATLAB*.

1.2.3 Разработка правил трансформаций исходной модели в целевую модель

1.2.3.1 Графовые правила трансформации

В данном разделе с помощью графов описаны правила преобразования SCL структуры схемы оборудования цифровых подстанций в *MATLAB* модель.

Схему оборудования цифровой подстанции, описанную в SCL файле можно представить в виде графа, где множество вершин S_i являются тегами «ConnectivityNode», а множество C_j – тегами «ConductingEquipment». Связи между вершинами формируются на основании вложенных тегов “Terminal”, где в атрибутах указано название тега «ConnectivityNode» с которым имеется связь.

Граф *MATLAB* модели, состоит из множества диаграмм *MATLAB* объектов C_j , в которых имеются левосторонние и правосторонние входы. Эти входы соединяются между собой множеством вершин S_k .

Трансформация графа схемы (из *SCL* файла) в граф *MATLAB* модели выполняется с помощью двух правил:

- первое правило описывает преобразование графа, где вершина *S* связывает две вершины *C*;
- второе правило описывает преобразование графа, где вершина *S* связывает более двух вершин *C*.

В первую очередь необходимо искать в графе применение второго правила.

Правило трансформации для двух вершин *C* представлено на рисунке 10. Вершины *C1* и *C2* заменяются на диаграммы *MATLAB* объектов у которых по три левосторонних и правосторонних входов. Вершина *S1* заменяется на три вершины, которые соединяют между собой входы свободных сторон диаграмм *C1* и *C2*.

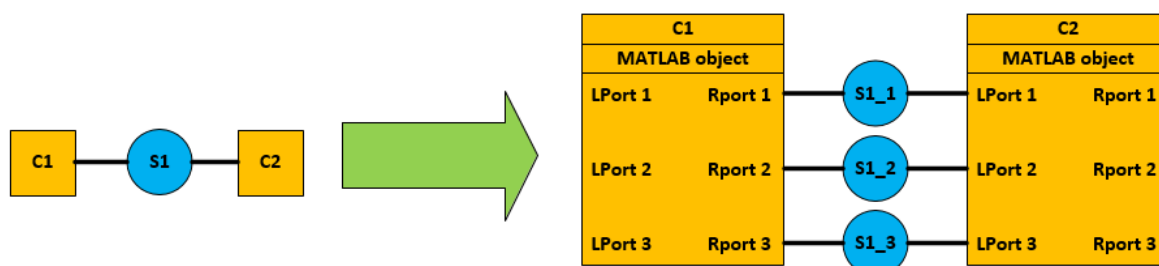


Рисунок 10 - Правило для двух вершин *C*

Правило трансформации для множества вершин *C* изображено на рисунке 11. Множество вершин *C1, C2..Cn* заменяются на диаграммы *MATLAB* объектов. Вершина *S1* заменяется на три вершины, которые соединяют между собой входы свободных сторон множества диаграмм *C1, C2..Cn*.

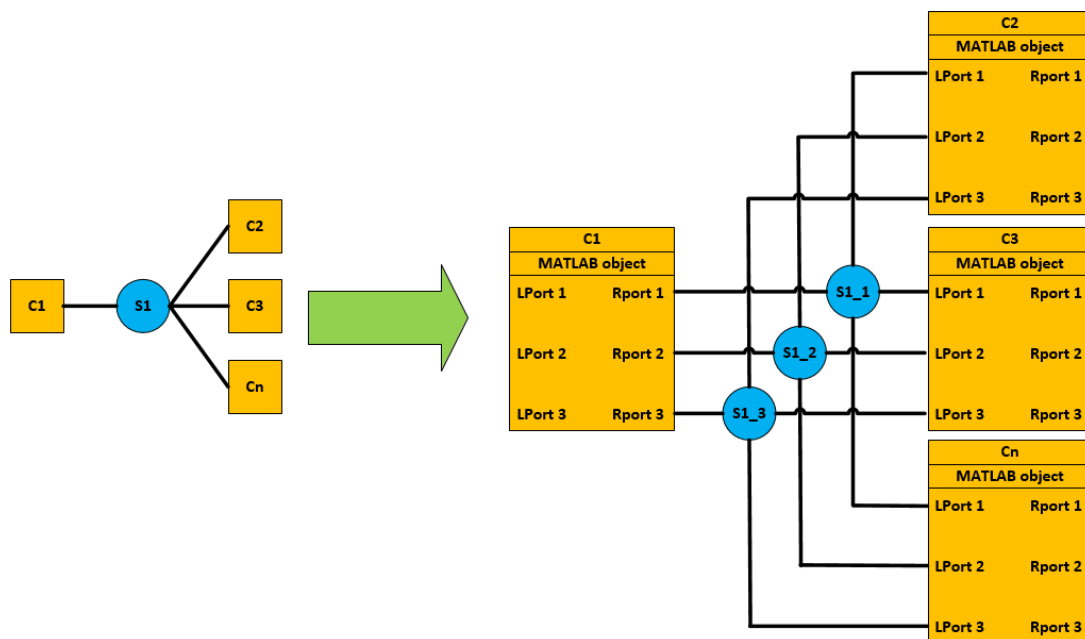


Рисунок 11 - Правило для множества вершин C

1.2.3.2 Правила трансформации SCL описания в сценарий $MATLAB$

Для трансформации типов оборудования стандарта МЭК 61850 в блоки $MATLAB$ модели, которые являются их функциональным аналогом, была составлена таблица соответствий (Таблица 2).

Таблица 2 - Список типов оборудования стандарта МЭК 61850 и соответствующие им объекты $MATLAB$

Тип оборудования МЭК 61850	Значение	$MATLAB$ объект
<i>CBR</i>	защитный выключатель	<i>Three-Phase Breaker</i>
<i>CTR</i>	измеритель тока	<i>Three-Phase V-I Measurement</i>
<i>IFL</i>	заземление	<i>Ground</i>
<i>DIS</i>	выключатель	<i>Three-Phase Breaker</i>
<i>CAP</i>	конденсатор	<i>Three-Phase RLC Branch</i>

Тип оборудования МЭК 61850	Значение	<i>MATLAB</i> объект
<i>BAT</i>	батарея	<i>Battery</i>
<i>CON (AC/DC)</i>	преобразователь переменного тока в постоянный	<i>CCCV Battery Charger</i>
<i>CON (DC/AC)</i>	преобразователь постоянного тока в переменный	<i>Inverter (Three-Phase)</i>
<i>PSH</i>	нагрузка	<i>RLC Brunch</i>

Также был разработан свод правил для проведения связей между объектами *MATLAB/SIMULINK* модели. Подробное описание правил представлено ниже:

- максимальное количество сторон соединения для всех объектов МЭК 61850 и *MATLAB/SIMULINK* равно двум (лево и право);
- одна линия соединения в МЭК 61850 меньше или равна трем в *MATLAB/SIMULINK*;
- нумерация контактов объектов *SIMULINK* и МЭК 61850 одинакова (левый верхний контакт —первый, правый нижний— последний);
- только два компонента соединены соединительными проводами;
- если три или более объектов МЭК 61850 соединены, «первичный» компонент, который напрямую подключен ко всем другим компонентам (так называемая топология «звезда»), будет первым элементом, имеющим эту общую линию связи;
- количество проводов соединения *MATLAB/SIMULINK* между двумя объектами равно минимальному количеству портов (одной из сторон соединения) 2-х взаимосвязанных компонентов.

1.3 Методика автоматической генерации функционально-блочной моделей распределенного управления для цифровых подстанций МЭК 61850

1.3.1 Основные положения методики

Данная методика позволяет генерировать модели распределенного управления в виде человеко-машинного интерфейса для осуществления контроля над *MATLAB/SIMULINK* моделью. Методика включает в себя два этапа:

- разработка шаблонного проекта в системе *nxtStudio*, в котором будут создана библиотека функциональных блоков для каждого из типов оборудования;
- трансляция оборудования из *SCL* представления цифровой подстанции на устройство в проекте *nxtStudio* с использованием разработанной библиотеки функциональных блоков.

1.3.2 Разработка функциональных блоков

Стандарт МЭК 61499 относится к области аппаратной автоматизации, что означает, что все устройства, поддерживающие этот стандарт, имеют функции управления некоторой механической, электрической, или комбинированной системой. Любая аппаратура управления большую часть времени выполняет две основные функции:

- чтение входа (это может быть простой аналоговый/цифровой сигнал или любой протокол связи);
- отправка сигнала (мигание/вызов/рассылка и т.д.).

Были разработаны два типа функциональных блока стандарта МЭК 61499, реализующие две функции, о которых говорилось выше. Первый тип функциональных блоков осуществляет процесс получения данных и их отображения на человеко-машинном интерфейсе. На рисунках 12 и 13 в качестве примера представлен функциональный блок и его человеко-

машинный интерфейс для конденсатора (CAP) из оборудования стандарта МЭК 61850. У функционального блока имеются следующие событийные входы и выходы:

- *INIT*, событие инициализации функционального блока;
- *REQ*, событие-запрос;
- *INTO*, событие завершения инициализации функционального блока.

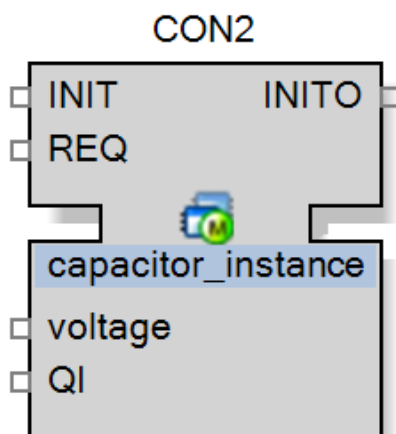


Рисунок 12- Интерфейс функционального блока чтения

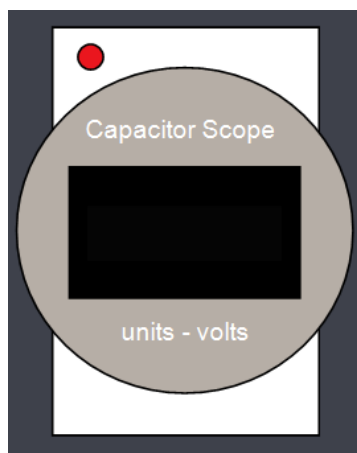


Рисунок 13 - Человеко-машинный интерфейс функционального блока чтения

Также имеется параметризованный вход «*voltage*» типа *INT* с помощью которого осуществляется чтение данных. Он связан с событийными входами *INIT* и *REQ*. Человеко-машинный интерфейс функционального блока

представляет собой дисплей, в котором отображается считываемая информация, и подписи, позволяющие понять, какие данные показываются.

Второй тип функциональных блоков осуществляет процесс отправки данных. На рисунках 14 и 15 в качестве примера представлен функциональный блок и его человеко-машинный интерфейс для переключателя (CBR) из стандарта МЭК 61850.

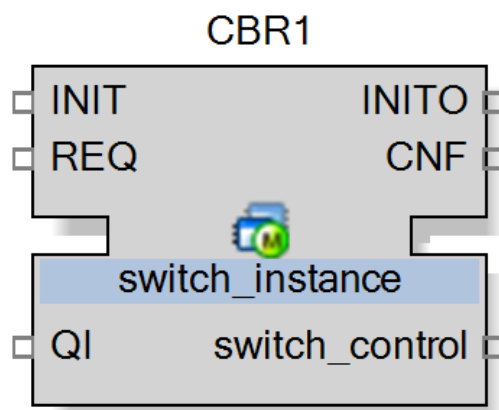


Рисунок 14 - Интерфейс функционального блока отправки

Человеко-машинный интерфейс функционального блока представляет собой кнопку, которая изменяет свой цвет по нажатию с красного на зеленый и на оборот.

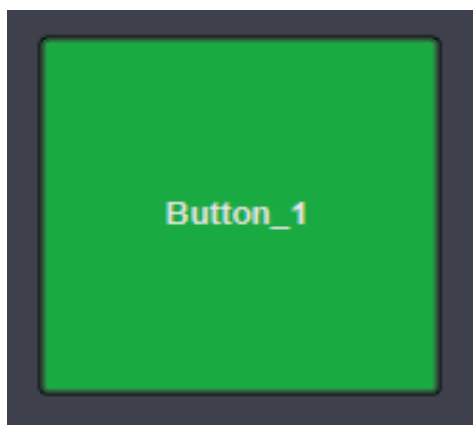


Рисунок 15- Человеко-машинный интерфейс функционального блока отправки

1.3.3 Алгоритм генерации функционально-блочной модели из *SCL* представления цифровой подстанции МЭК 61850

Цель алгоритма генерации функционально-блочной модели из *SCL* представления цифровой подстанции МЭК 61850 – получение проекта системы *nxtStudio*, где на устройстве будет размещена функционально-блочная модель управления стандарта МЭК 61499, где каждому элементу модели цифровой подстанции соответствует определенный тип функционального блока (список соответствия представлен в таблице 3).

Таблица 3 - Список соответствия типов оборудования МЭК 61850 и функциональных блоков.

Тип оборудования МЭК 61850	Тип функционального блока
<i>CBR</i>	<i>switch_instance</i>
<i>CTR</i>	<i>current_measurement_instance</i>
<i>VTR</i>	<i>volt_measurement_instance</i>
<i>DIS</i>	<i>switch_instance</i>
<i>CAP</i>	<i>capacitor_instance</i>
<i>BAT</i>	<i>battery_instance</i>

Для получения готовой рабочей функционально-блочной модели необходимо описать ее в проекте МЭК 61499. Файлы проекта имеют структуру *XML* (как и IEC61850). Необходимо изменить следующий *XML*-файл проекта МЭК 61499:

- *System.sys*;
- *System.cfg*;
- *IEC61499.dfbproj*;
- *HMI.csproj*.

Файл «System.sys» (Рисунок 16) описывает сеть функциональных блоков на устройствах. Для каждого экземпляра оборудования в файле SCL, имеющего функциональный блок в шаблонном проекте, создается запись в теге «FBNetwork».

```
<FB
  ID="5C6K38RDXR0IDR34"
  Name="VT1"
  Type="volt_measurment_instance"
  x="12"
  y="16"
  Namespace="Main"
  Loaded="true">
</FB>
```

Рисунок 16– Структура тэга «FB» в XML файле «System.sys»

Входное событие «INIT» добавляемого блока всегда связывается с выходным событием «COLD» функционального блока «START» (Рисунок 17).

```
<EventConnections>
  <Connection
    Source="START.COLD"
    Destination="VT1.INIT" />
</EventConnections>
```

Рисунок 17– Структура тэг «EventConnections» в XML файле «System.sys»

Файл «System.cfg» описывает привязку функционального блока к системным ресурсам (Рисунок 18). Атрибут «Map» тега «ResInst» получается в результате объединения идентификатора устройства и идентификатора ресурса из файла «System.sys».

В файле «IEC61499.dfbproj» описывается порядок компиляции функционального блока. А в файле «HMI.csproj» описывает порядок

компиляции человеко-машинного интерфейса. Поэтому в них добавляются необходимые записи для всех функциональных блоков, используемых в проекте.

```
<System Name="System" Dir="System">
  <DeviceItem>
    <DeviceItem Name="DEV0" />
  </DeviceItem>
  <CATType
    Name="volt_measurment_instance"
    Namespace="Main">
    <ResInst
      ID="5C6K38RDXR0IDR34"
      Name="Res-DEV0-RES0-VT1"
      Map="2371D82C743A513C.28A29A086E15ACB3;" />
    </ResInst>
  </CATType>
</System>
```

Рисунок 18– Структура тэг «System» в XML файле «System.cfg»

1.4 Разработка программных средств для поддержки методик автоматической генерации

1.4.1 Основные структуры данных

Разработанное приложение получает на вход следующие данные:

- *SCL* файл, описывающий цифровую подстанцию МЭК 61850;
- *XML* файл конфигурации, в котором описана таблица соответствия типов оборудования МЭК 61850 и *MATLAB* объектов;
- проект *nxtStudio* стандарта МЭК 61499, в котором имеется разработанная библиотека функциональных блоков.

На выход приложение генерирует:

- сценарий *MATLAB*, который используется для создания модели *SIMULINK*;
- проект *nxtStudio* стандарта 61499, в котором размещена функционально-блочная модель, полученная из *SCL* файла.

Для хранения данных из *SCL* файла был разработан класс объекта МЭК 61850 параметры которого представлены в таблице 4. Также был разработан класс для хранения информации о *MATLAB* объекте (Таблица 5).

Таблица 4 - Параметры объекта МЭК 61850

Имя параметра	Описание
<i>type</i>	тип объекта МЭК 61850 (например « <i>CBR</i> », « <i>CTR</i> » и т.д)
<i>name</i>	имя объекта МЭК 61850
<i>X</i>	позиция объекта на координатной оси <i>X</i>
<i>Y</i>	позиция объекта на координатной оси <i>Y</i>
<i>connection1</i> <i>connectionN</i>	- список имен соединений между элементами (атрибут « <i>connectivityNode</i> » из тега « <i>Terminal</i> »)

Таблица 5 - Параметры объекта MATLAB

Имя параметра	Описание
<i>path</i>	путь к объекту в библиотеке <i>MATLAB</i> (пример <i>electricdrivelib/Extra Sources/Battery</i>)
<i>name</i>	имя <i>MATLAB</i> объекта
<i>count_num_R</i>	число правосторонних портов <i>MATLAB</i> объекта
<i>count_num_L</i>	число левосторонних портов <i>MATLAB</i> объекта

Данные из XML файла конфигурации имеют вид, представленный на рисунке 9.

```
<SCDElement
  name="CBR"
  path="powerlib/Elements/Three-Phase Breaker"
  LPorts = "3" RPorts = "3">
  <parameter>
    'InitialState', 'open'
  </parameter>
</SCDElement>
```

Рисунок 19 - Пример описания файла конфигурации

1.4.2 Алгоритмы программы

В разработанном приложении можно выделить три основных алгоритма: чтение SCL файла с выбором необходимых данных, генерация сценария MATLAB, генерация проекта nxtStudio. Для данных алгоритмов были построены блок-схемы они представлены в приложение Б.

Алгоритм чтение SCL файла, выполняет обход XML документа по тегам. Во время обхода формируются списки объектов МЭК 61850 и MATLAB. Псевдокод алгоритма представлен ниже.

```
ЕСЛИ тег имеет вложенные теги, ТО
  ДЛЯ i = 0 ДО количества вложенных тегов ШАГ 1
    ЕСЛИ данные элемента заполнены, ТО
      Добавить элемент в список;
      Создать новый элемент;
    Данные элемента не заполнены;
    ЕСЛИ вложенный тег == "ConductingEquipment", ТО
      Данные элемента заполнены;
      Записать атрибут name вложенного тега в поле имя;
      Записать атрибут type вложенного тега в поле
тип;
      Записать атрибуты sxy вложенного тега в поля X и Y;
      Добавить в список MATLAB объектов новый элемент
```

В соответствии с типом;
 ДЛЯ $j = 0$ ДО количества вложенных тегов во вложенном теге
 ШАГ 1

ЕСЛИ вложенный тег == "Terminal", ТО
 Добавить в список соединений элемента новое соединение;
 ВСЕ_ДЛЯ_ j
 ПОВТОРИТ алгоритм для вложенного тега
 ВСЕ_ДЛЯ_ i

В алгоритме генерация сценария *MATLAB* особый интерес представляет алгоритм проведения связей между *MATLAB* объектами, его псевдокод представлен ниже.

Создать список соединений;
 ПОКА список элементов имеет следующий элемент
 ЕСЛИ список соединений элемента не пуст, ТО
 Добавить элемент в список соединений;
 ПОКА список элементов имеет следующий элемент
 ЕСЛИ имя элемента из ЦИКЛ1 не равно имени элемента из ЦИКЛ2, ТО
 ЕСЛИ элемент из ЦИКЛ2 есть в списке соединений элемента ЦИКЛ1, ТО
 Добавить элемент ЦИКЛ2 в список соединений;
 Удалить из списка соединений элемента ЦИКЛ2 элемент ЦИКЛ1;
 Перейти к следующему элементу;
 ВСЕ_ЦИКЛ2
 ЕСЛИ количество элементов в списке соединений > 1 , ТО
 Базовый элемент равен первому элементу в списке соединений;
 ПОКА список соединений имеет следующий элемент
 ЕСЛИ базовый элемент не равен элементу списка, ТО
 Стартовый объект равен *MATLAB* объекту базового элемента;
 Конечный объект равен *MATLAB* объекту элемента списка;
 ЕСЛИ левая сторона стартового объекта не подключена, ТО
 ЕСЛИ левая сторона конечного объекта не подключена, ТО
 Соединить левые стороны стартового и конечного объектов;

ИНАЧЕ ЕСЛИ правая сторона конечного объекта не подключена,
ТО

Соединить левую сторону стартового объекта и правую сторону
конечного объекта;

ИНАЧЕ ошибка соединения;

ИНАЧЕ ЕСЛИ правая сторона стартового объекта не подключена,
ТО

ЕСЛИ левое сторона конечного объекта не подключена, ТО

Соединить правую сторону стартового объекта и левую сторону
конечного объекта;

ИНАЧЕ ЕСЛИ правая сторона конечного объекта не подключена,
ТО

Соединить правые стороны стартового и конечного объектов;

ИНАЧЕ ошибка соединения;

ИНАЧЕ ошибка соединения;

Перейти к следующему элементу;

ВСЕ_ЦИКЛ3

Перейти к следующему элементу;

ВСЕ_ЦИКЛ1

1.4.3 Архитектура программных средств

В качестве инструмента разработки был выбран язык программирования C#. Язык программирования C# — это типизированный, объектно-ориентированный, простой и в то же время мощный язык программирования, который позволяет разработчикам создавать многофункциональные приложения. C# поддерживает много полезных функций:

- инкапсуляция,
- наследование,
- полиморфизм,
- перегрузка операторов,
- статическая типизация.

Разработчики языка C# включили в него библиотеки для работы с XML, которые необходимы в данной работе для чтения SCL файлов формата XML.

Инструментарий C# позволяет решать широкий круг задач, язык действительно очень мощный и универсальный. На нём часто разрабатывают:

- веб-приложения,
- игры,
- мобильные приложения для *Android* или *iOS*,
- программы под *Windows*.

Перечень возможностей разработки практически не имеет ограничений благодаря широчайшему набору инструментов и средств. Конечно, всё это можно реализовать при помощи других языков. Но некоторые из них узкоспециализированные, а в некоторых придётся использовать дополнительные инструменты сторонних разработчиков. В C# решить широкий круг задач возможно быстрее, проще и с меньшими затратами времени и ресурсов.

1.4.4 Руководство пользователя

При разработке любого программного продукта необходимо составить руководство пользователя для того, что у пользователя не возникало никаких проблем при работе с ним.

После запуска программы появляется её основное окно (Рисунок 20).

Пользователь с помощью элементов меню (Рисунок 21) может выполнить следующие действия:

- «File-> Open SCD», вызвать диалоговое для открытия SCL файла. После открытие файла выполняется автоматическая генерация сценария MATLAB;
- «File-> Save script», вызвать диалоговое окно для сохранения сценария MATLAB;

- «File-> *Generate 61499 project*», сгенерировать проект стандарта МЭК 61499 для открытого в данный момент *SCL* файла;
- «Settings», открыть окно для редактирования файла конфигурации.

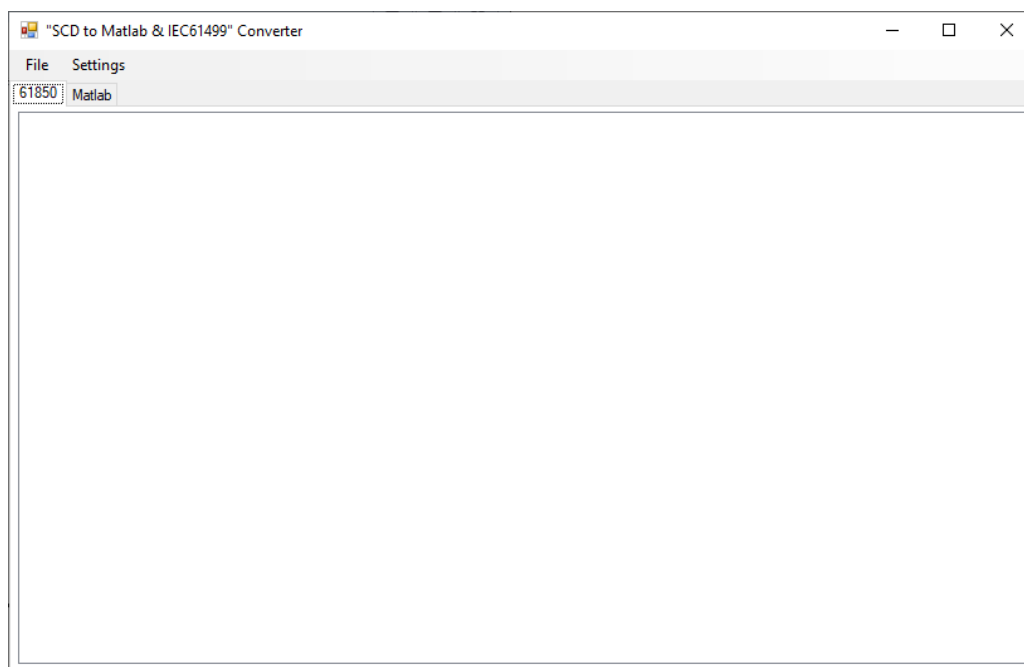


Рисунок 20 - Главное окно приложения

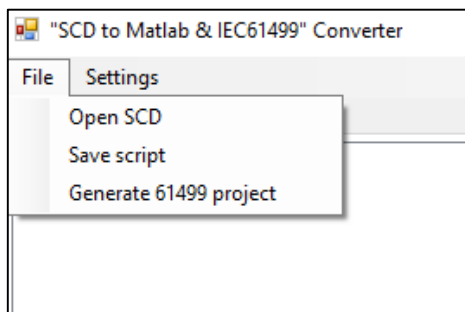


Рисунок 21 - Элементы меню

В окне настроек (Рисунок 22) пользователь может добавить/удалить конфигурации для типов МЭК 61850. В конфигурации можно изменить путь к компоненту в библиотеке *MATLAB*, выбрать тип организации подключения к компоненту, добавить/изменить/удалить параметры компонента.

После открытия, выбранного *SCL* файла на вкладке «61850» (Рисунок 23) отобразится содержимое этого файла в виде дерева. А на вкладке

«Matlab» (Рисунок 24) будет отображен сценарий *MALAB* с помощью которого можно сгенерировать модель *MATLAB/SIMULINK*.

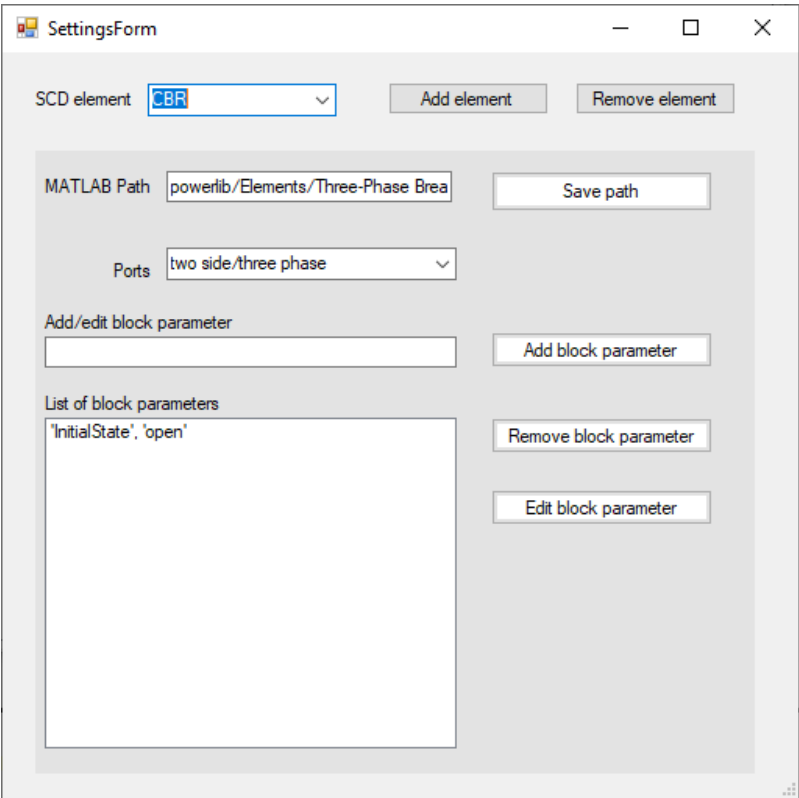


Рисунок 22 - Окно настроек

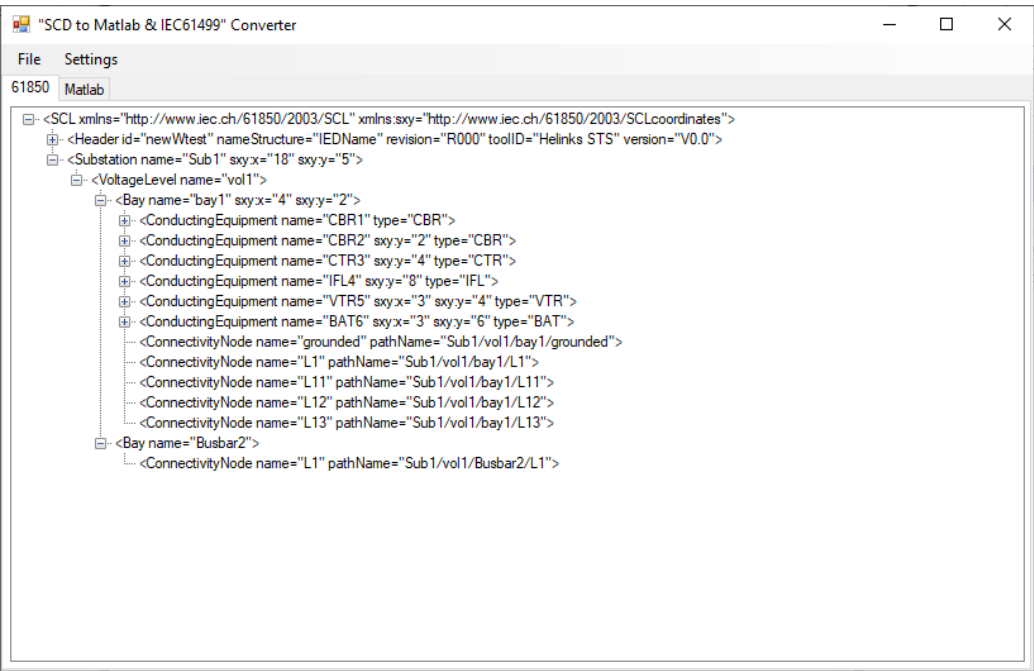


Рисунок 23 - Главное окно с видом на содержимое *SCL* файла

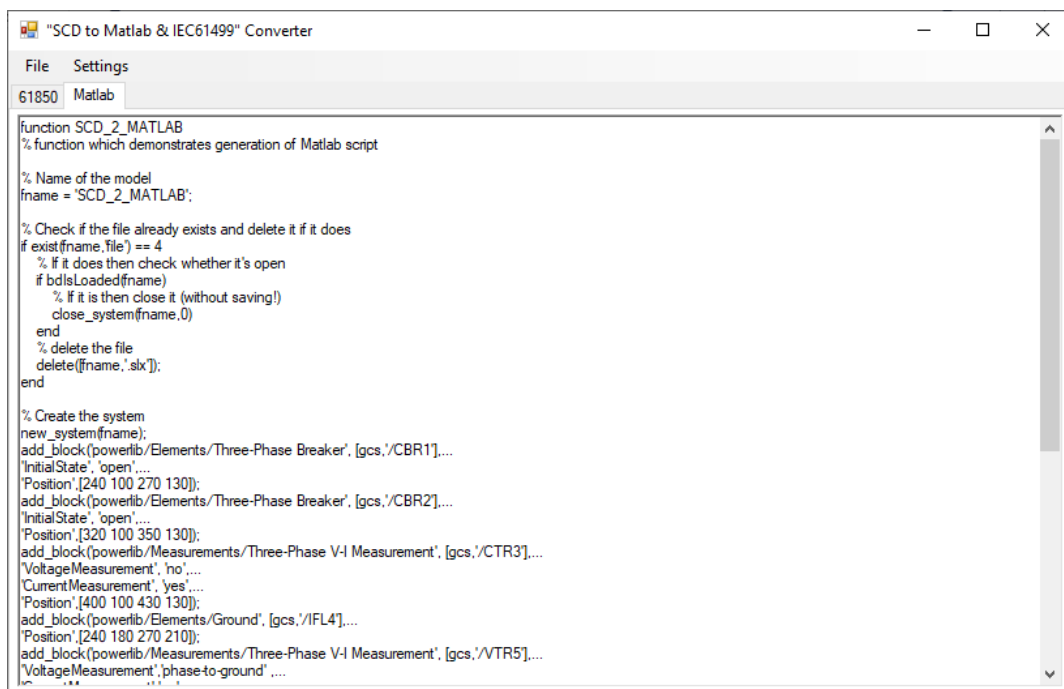


Рисунок 24 - Главное окно с видом на сгенерированный сценарий
MATLAB

1.4.5 Тестирование и отладка

Для тестирования разработанного приложения в системе *Helinks STS*, предназначенной для проектирования цифровых подстанций, был построен фрагмент электроэнергетической сети (Рисунок 25). *SCL* описание данного фрагмента сети представлено ниже.

```
<?xml version="1.0" encoding="UTF-8"?>
<SCL
xmlns="http://www.iec.ch/61850/2003/SCL"
xmlns:sxy="http://www.iec.ch/61850/2003/SCLcoordinates">
  <Header
    id="newWtest"
    nameStructure="IEDName"
    revision="R000" toolID="Helinks STS" version="V0.0">
    <History/>
  </Header>
  <Substation name="Substation1" sxy:x="16" sxy:y="2">
    <VoltageLevel name="vol1">
      <Bay name="bay1" sxy:x="1" sxy:y="1">
        <ConductingEquipment name="CBR1" type="CBR">
```

```

        <Terminal bayName="Power bus" cNodeName="L1"
name="T1" substationName="Substation1" voltageLevelName="vol1"
connectivityNode="Substation1/vol1/Power bus/L1"/>
        <Terminal bayName="bay1" cNodeName="L1" name="T2"
substationName="Substation1" voltageLevelName="vol1"
connectivityNode="Substation1/vol1/bay1/L1"/>
    </ConductingEquipment>
    <ConductingEquipment name="CTR2" sxy:y="3"
type="CTR">
        <Terminal bayName="bay1" cNodeName="L11"
name="T2" substationName="Substation1" voltageLevelName="vol1"
connectivityNode="Substation1/vol1/bay1/L11"/>
        <Terminal bayName="bay1" cNodeName="L12"
name="T2" substationName="Substation1" voltageLevelName="vol1"
connectivityNode="Substation1/vol1/bay1/L12"/>
    </ConductingEquipment>
    <ConductingEquipment name="IFL3" sxy:y="4"
type="IFL">
        <Terminal bayName="bay1" cNodeName="L11"
name="T1" substationName="Substation1" voltageLevelName="vol1"
connectivityNode="Substation1/vol1/bay1/L11"/>
    </ConductingEquipment>
    <ConductingEquipment name="VTR4" sxy:x="2"
sxy:y="1" type="VTR">
        <Terminal bayName="bay1" cNodeName="L1" name="T1"
substationName="Substation1" voltageLevelName="vol1"
connectivityNode="Substation1/vol1/bay1/L1"/>
    </ConductingEquipment>
    <ConductingEquipment name="CBR5" sxy:y="1"
type="CBR">
        <Terminal bayName="bay1" cNodeName="L1"
name="T1" substationName="Substation1" voltageLevelName="vol1"
connectivityNode="Substation1/vol1/bay1/L1"/>

```

					<p style="text-align: center;"><i>ПГУ 1.09.03.01.35.001 ПЗ</i></p>	Лист
Изм.	Лист	№ докум	Подпись	Дата		39

```

        <Terminal      bayName="bay1"      cNodeName="L12"
name="T2"  substationName="Substation1"  voltageLevelName="vol1"
connectivityNode="Substation1/vol1/bay1/L12"/>
    </ConductingEquipment>
    <ConnectivityNode      name="grounded"
pathName="Substation1/vol1/bay1/grounded"/>
    <ConnectivityNode      name="L11"
pathName="Substation1/vol1/bay1/L11"/>
    <ConnectivityNode      name="L1"
pathName="Substation1/vol1/bay1/L1"/>
    <ConnectivityNode      name="L12"
pathName="Substation1/vol1/bay1/L12"/>
    </Bay>
    <Bay name="Power bus">
        <ConnectivityNode      name="L1"
pathName="Substation1/vol1/Power bus/L1"/>
    </Bay>
</VoltageLevel>
</Substation>
</SCL>

```

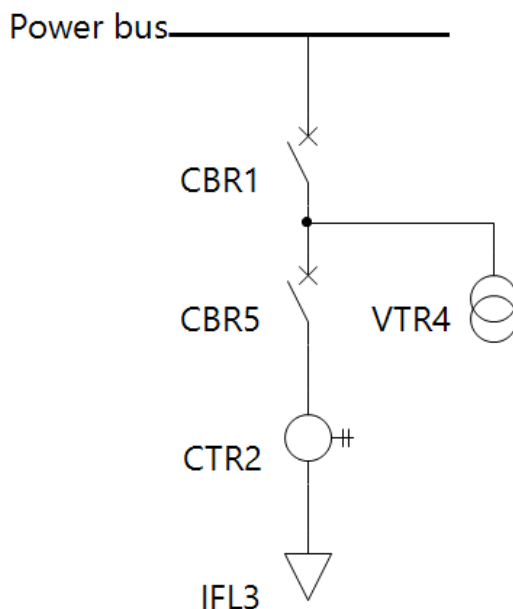


Рисунок 25 - Фрагмент электроэнергетической сети

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

40

С помощью приложения был получен сценарий *MATLAB*, представленный ниже.

```
function SCD_2_MATLAB
% function which demonstrates generation of Matlab script
% Name of the model
fname = 'SCD_2_MATLAB';
% Check if the file already exists and delete it if it does
if exist(fname,'file') == 4
    % If it does then check whether it's open
    if bdIsLoaded(fname)
        % If it is then close it (without saving!)
        close_system(fname,0)
    end
    % delete the file
    delete([fname, '.slx']);
end
% Create the system
new_system(fname);
add_block('powerlib/Elements/Three-Phase Breaker',
[gcs, '/CBR1'],...
'InitialState', 'open',...
'Position',[200 200 300 250]);
add_block('powerlib/Measurements/Three-Phase V-I
Measurement', [gcs, '/CTR2'],...
'VoltageMeasurement', 'no',...
'CurrentMeasurement', 'yes',...
'Position',[200 440 300 490]);
add_block('powerlib/Elements/Ground', [gcs, '/IFL3'],...
'Position',[200 520 300 570]);
add_block('powerlib/Measurements/Three-Phase V-I
Measurement', [gcs, '/VTR4'],...
'VoltageMeasurement','phase-to-ground' ,...
'CurrentMeasurement','no' ,...
'Position',[360 280 460 330]);
```

```

        add_block('powerlib/Elements/Three-Phase Breaker',
[gcs, '/CBR5'],...
    'InitialState', 'open',...
    'Position',[200 280 300 330]);
add_line(gcs,'CTR2/LConn1','IFL3/LConn1');
add_line(gcs,'CTR2/LConn2','IFL3/LConn1');
add_line(gcs, 'CTR2/LConn3', 'IFL3/LConn1');
add_line(gcs,'VTR4/LConn1','CBR1/LConn1');
add_line(gcs,'VTR4/LConn2','CBR1/LConn2');
add_line(gcs, 'VTR4/LConn3', 'CBR1/LConn3');
add_line(gcs,'VTR4/LConn1','CBR5/LConn1');
add_line(gcs,'VTR4/LConn2','CBR5/LConn2');
add_line(gcs, 'VTR4/LConn3', 'CBR5/LConn3');
add_line(gcs,'CBR5/RConn1','CTR2/RConn1');
add_line(gcs,'CBR5/RConn2','CTR2/RConn2');
add_line(gcs, 'CBR5/RConn3', 'CTR2/RConn3');

% Set a couple of model parameters to eliminate warning
messages

set_param(gcs,...
    'Solver','FixedStepDiscrete',...
    'FixedStep','0.1');

% Save the model
save_system(fname);

```

В системе *MATLAB/SIMULINK* данный скрипт формирует модель, представленную на рисунке 26. После ручного перемещения блоков можно получить модель, представленную на рисунке 27. Название, тип и соединение блоков полученной модели соответствует *SCL* представлению электроэнергетической сети, следовательно, приложение работает правильно.

При генерации проекта МЭК 61499 была получена система функциональных блоков, представленная на рисунке 28. Человеко-машинный интерфейс данной системы имеет вид, изображенный на рисунках 29 и 30.

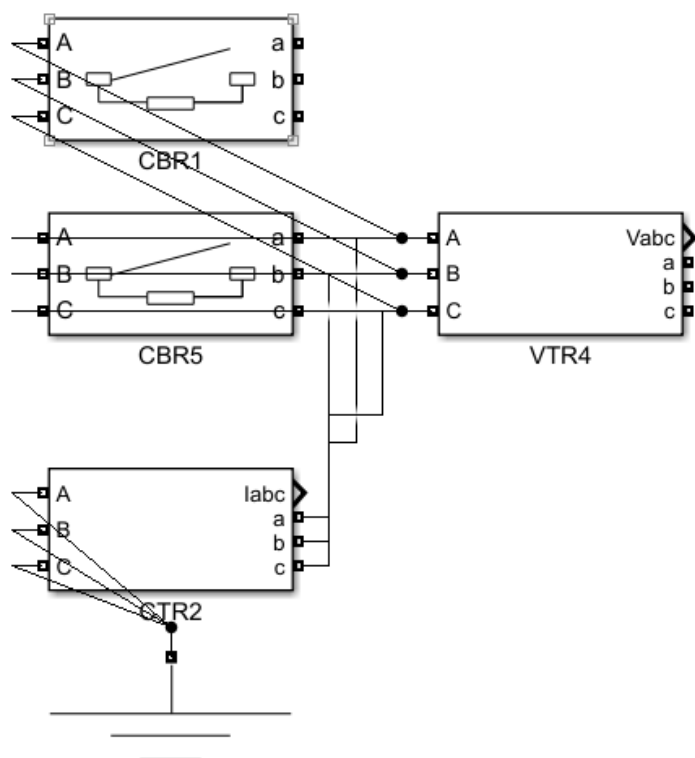


Рисунок 26 - Сгенерированная модель *MATLAB/SIMULINK*

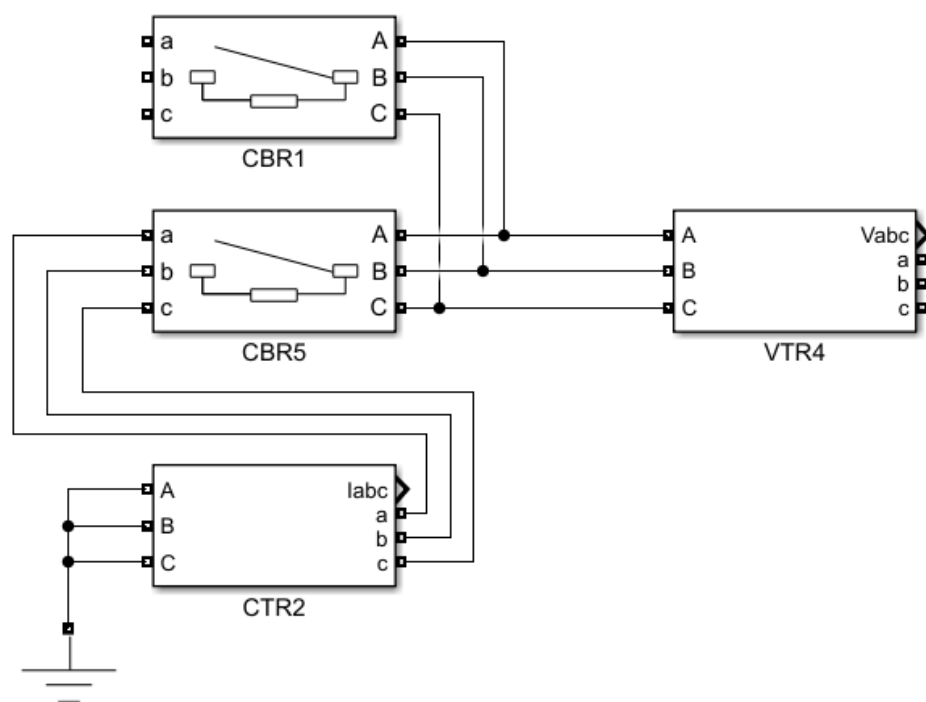


Рисунок 27 - Сгенерированная модель *MATLAB/SIMULINK*

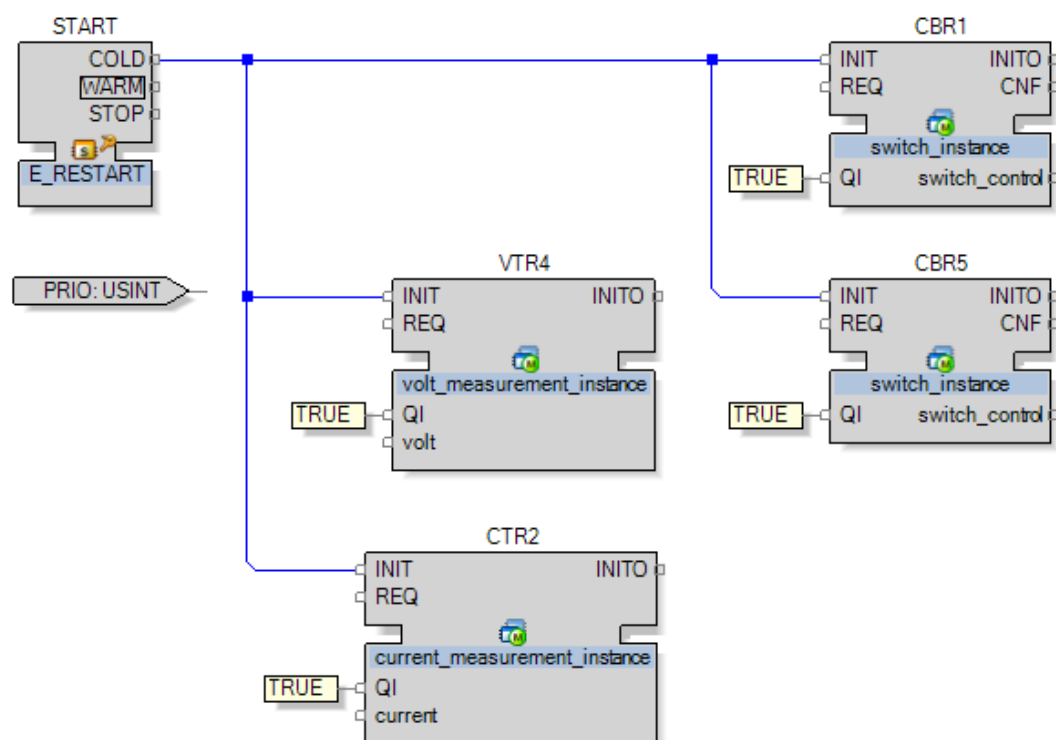


Рисунок 28 - Сгенерированная функционально блочная модель

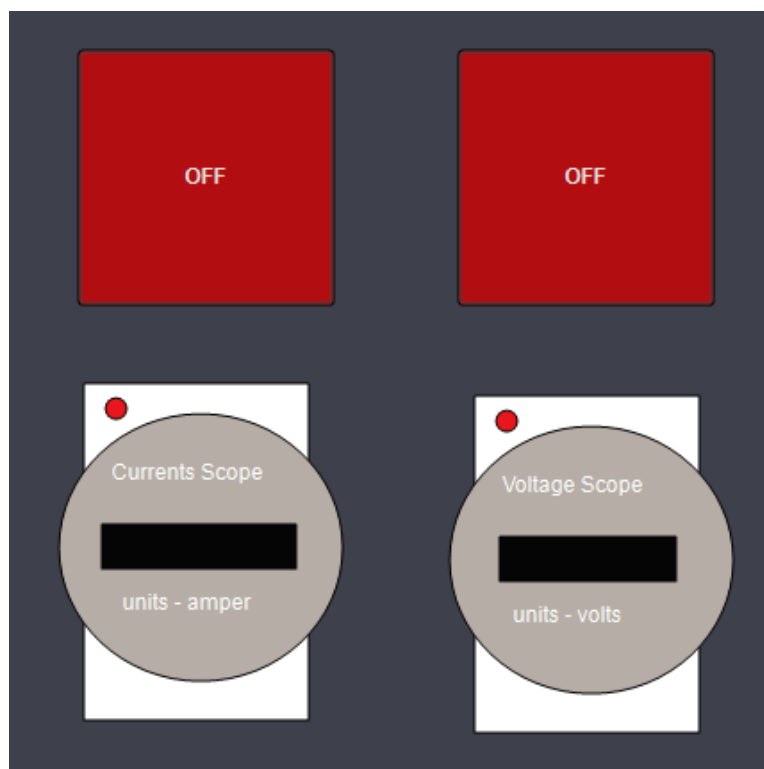


Рисунок 29 – Человеко-машинный интерфейс сгенерированной функционально блочная модели



Рисунок 30– Человеко-машинный интерфейс сгенерированной функционально блочная модели

2 Безопасность жизнедеятельности

2.1 Введение

Освещение рабочего места – важнейший фактор создания нормальных условий труда. Практически возникает необходимость освещения как естественным, так и искусственным светом. Первый случай характерен для светлого времени суток и при работе в помещениях, в которых имеются проемы в стенах и крыше здания, во втором случае применяются соответствующие осветительные установки искусственного света.

Естественное освещение по своему спектральному составу является наиболее приемлемым. Искусственное же, наоборот, отличается относительной сложностью восприятия его зрительным органом человека. Это связано с тем, что суточные переходные режимы естественной освещенности имеют малую частоту при достаточно высокой (днем) или очень низкой (ночью) интенсивности светового потока, а искусственные – довольно большую частоту при недостаточной в целом освещенности. Поэтому при искусственном освещении начинают возникать неустойчивые зрительные процессы, которые из-за большой частоты сменяемости световых условий накладываются друг на друга, не давая глазу времени адаптироваться к новым условиям. От усиленной деятельности приспособительных механизмов глаза быстро утомляются, что вызывает физическую усталость организма.

Несмотря на это, искусственное освещение необходимо, как важнейший фактор для приближения ночных условий труда к дневным. Основное отличие ночных условий труда от дневных состоит в том, что при ночных условиях труда отсутствует достаточная освещенность поля зрения работающих равномерно распределенным световым потоком. Стимулирующее действие света на организм при недостаточной освещенности снижается, поэтому ночные условия труда более тяжелые с

физиологической точки зрения. Однако основа естественного и искусственного света общая – энергетическая, поэтому их разделение вызвано разницей в спектре и интенсивности. Последнее предположение можно принять только в первом приближении, поскольку спектр естественного света полностью не выяснен и не воспроизведен. Более реально создание световой среды, обеспечивающей психофизиологический комфорт и заданное эмоционально-эстетическое воздействие с учетом цветоцветовой доминанты в поле зрения.

По функциональному назначению искусственное освещение подразделяют на рабочее, аварийное и специальное, которое может быть охранным, дежурным, эвакуационным, бактерицидным и др. Нормы освещенности для каждого из видов установлены в СНиП 23-05-95 «Естественное и искусственное освещение» от 01.01.1996.

Рабочее освещение предназначено для обеспечения нормального выполнения производственного процесса, прохода людей, движения транспорта и является обязательным для всех производственных помещений.

Аварийное освещение устраивают для продолжения работы в тех случаях, когда внезапное отключение рабочего освещения (при авариях) и связанное с этим нарушение нормального обслуживания оборудования могут вызвать взрыв, пожар, отравление людей, нарушение технологического процесса и т.д. Минимальная освещенность рабочих поверхностей при аварийном освещении должна составлять 5% нормируемой освещенности рабочего освещения, но не менее 2 лк.

Эвакуационное освещение предназначено для обеспечения эвакуации людей из производственного помещения при авариях и отключении рабочего освещения; организуется в местах, опасных для прохода людей: на лестничных клетках, вдоль основных проходов производственных помещений, в которых работают более 50 чел. Минимальная освещенность

на полу основных проходов и на ступеньках при эвакуационном освещении должна быть не менее 0,5 лк, на открытых территориях – не менее 0,2 лк.

Охранное освещение устраивают вдоль границ территорий, охраняемых специальным персоналом. Наименьшая освещенность в ночное время 0,5 лк.

Сигнальное освещение применяют для фиксации границ опасных зон; оно указывает на наличие опасности, либо на безопасный путь эвакуации.

2.2 Требования к помещениям для работы с ПЭВМ

Требования к помещениям для работы с ПЭВМ определены в СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы» и содержат следующие положения:

- помещения для эксплуатации ПЭВМ должны иметь естественное и искусственное освещение. Эксплуатация ПЭВМ в помещениях без естественного освещения допускается только при соответствующем обосновании и наличии положительного санитарно-эпидемиологического заключения, выданного в установленном порядке;

- естественное и искусственное освещение должно соответствовать требованиям действующей нормативной документации. Окна в помещениях, где эксплуатируется вычислительная техника, преимущественно должны быть ориентированы на север и северо-восток. Оконные проемы должны быть оборудованы регулируемыми устройствами типа: жалюзи, занавесей, внешних козырьков и др;

- не допускается размещение мест пользователей ПЭВМ во всех образовательных и культурно-развлекательных учреждениях для детей и подростков в цокольных и подвальных помещениях;

- площадь на одно рабочее место пользователей ПЭВМ с ВДТ на базе электронно-лучевой трубки (ЭЛТ) должна составлять не менее 6 м², в

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

48

помещениях культурно-развлекательных учреждений и с ВДТ на базе плоских дискретных экранов (жидкокристаллические, плазменные) - 4,5 м². При использовании ПЭВМ с ВДТ на базе ЭЛТ (без вспомогательных устройств - принтер, сканер и др.), отвечающих требованиям международных стандартов безопасности компьютеров, с продолжительностью работы менее 4-х часов в день допускается минимальная площадь 4,5 м² на одно рабочее место пользователя (взрослого и учащегося высшего профессионального образования);

- для внутренней отделки интерьера помещений, где расположены ПЭВМ, должны использоваться диффузно отражающие материалы с коэффициентом отражения для потолка - 0,7 - 0,8; для стен - 0,5 - 0,6; для пола - 0,3 - 0,5.

- полимерные материалы используются для внутренней отделки интерьера помещений с ПЭВМ при наличии санитарно-эпидемиологического заключения;

- помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации;

- не следует размещать рабочие места с ПЭВМ вблизи силовых кабелей и вводов, высоковольтных трансформаторов, технологического оборудования, создающего помехи в работе ПЭВМ.

2.3 Требования к освещению на рабочих местах, оборудованных ПЭВМ

Требования к освещению на рабочих местах, оборудованных ПЭВМ, определены в СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы» и содержат следующие положения:

- рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева;

- искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения. В производственных и административно-общественных помещениях, в случаях преимущественной работы с документами, следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов);

- освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 - 500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк;

- следует ограничивать прямую блескость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, должна быть не более 200 кд/м²;

- следует ограничивать отраженную блескость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране ПЭВМ не должна превышать 40 кд/м² и яркость потолка не должна превышать 200 кд/м²;

- показатель ослепленности для источников общего искусственного освещения в производственных помещениях должен быть не более 20. Показатель дискомфорта в административно-общественных помещениях - не более 40, в дошкольных и учебных помещениях - не более 15;

- яркость светильников общего освещения в зоне углов излучения от 50 до 90 градусов с вертикалью в продольной и поперечной плоскостях должна

составлять не более 200 кд/м², защитный угол светильников должен быть не менее 40 градусов;

- светильники местного освещения должны иметь не просвечивающий отражатель с защитным углом не менее 40 градусов;

- следует ограничивать неравномерность распределения яркости в поле зрения пользователя ПЭВМ, при этом соотношение яркости между рабочими поверхностями не должно превышать 3:1 - 5:1, а между рабочими поверхностями и поверхностями стен и оборудования - 10:1;

- в качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ). При устройстве отраженного освещения в производственных и административно-общественных помещениях допускается применение металлогалогенных ламп. В светильниках местного освещения допускается применение ламп накаливания, в том числе галогенных;

- для освещения помещений с ПЭВМ следует применять светильники с зеркальными параболическими решетками, укомплектованными электронными пускорегулирующими аппаратами (ЭПРА). Допускается использование многоламповых светильников с электромагнитными пускорегулирующими аппаратами (ЭПРА), состоящими из равного числа опережающих и отстающих ветвей. Применение светильников без рассеивателей и экранирующих решеток не допускается. При отсутствии светильников с ЭПРА лампы многоламповых светильников или рядом расположенные светильники общего освещения следует включать на разные фазы трехфазной сети;

- общее освещение при использовании люминесцентных светильников следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядном расположении видеодисплейных терминалов. При

периметральном расположении компьютеров линии светильников должны располагаться локализовано над рабочим столом ближе к его переднему краю, обращенному к оператору;

- коэффициент запаса для осветительных установок общего освещения должен приниматься равным 1,4;
- коэффициент пульсации не должен превышать 5%;
- для обеспечения нормируемых значений освещенности в помещениях для использования ПЭВМ следует проводить чистку стекол оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

2.4 Определение нормативных уровней освещенности

Каждый из разрядов работ и нормативные уровни освещенности, установленные для них, определены в СНиП 23-05-95 «Естественное и искусственное освещение» от 01.01.1996.

Работы, проводимые инженером-программистом, относятся к разряду работ высокой точности (III разряд), с наименьшими размерами объекта различения 0,3...0,5 мм. Подразряд зрительных работ типа (в) – работа со светящимися объектами при среднем контрасте объекта различения с фоном и при среднем фоне. Зрительная работа выполняется более половины рабочего дня. Работа ведется преимущественно с электронными, а не бумажными документами. На основании этих данных заключаем, что минимальная освещенность, создаваемая общим искусственным освещением, составляет 300 лк. Нормативная величина показателя ослепленности $P=40\%$, уровня пульсаций освещенности $K_n=15\%$.

2.5 Проект электрического освещения помещения при эксплуатации ПЭВМ

Рабочее помещение имеет следующие геометрические параметры:

					ПГУ 1.09.03.01.35.001 ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		52

- длина помещения 6 м.;
- ширина помещения 4 м.;
- высота 2,7 м.

В помещении, где находится рабочее место оператора используется смешанное освещение, т.е. сочетание естественного и искусственного освещения.

Естественное освещение – это боковое освещение через окна. Искусственное – используется при недостаточном естественном освещении. Поскольку работа оператора ЭВМ не относится к категории особо точных, выбираем систему общего освещения.

Расчет его осуществляется по методу светового потока с учетом потока, отраженного от стен, пола и потолка.

Нормами для данных работ установлена необходимая освещенность рабочего места $E_n=300$ лк. Общий световой поток $F_{общ}$ определяется по формуле:

$$F_{общ} = \frac{E_n * S * k_1 * k_2}{\eta},$$

где E_n – нормированная освещенность ($E_n=300$ лк.),

S – площадь помещения;

k_1 – коэффициент, учитывающий старение ламп и загрязнение светильников ($k_1=1,5$),

k_2 – коэффициент учитывающий неравномерность освещения помещения ($k_2=1,1$),

η – коэффициент использования светового потока. Определяется в зависимости от коэффициентов отражения от стен, потолка, рабочих поверхностей, типов светильников и геометрии помещения.

Расчет необходимых компонентов представлен ниже.

Площадь помещения S :

$$S = A * B = 6 * 4 = 24 \text{ м}^2,$$

где A – длина помещения,

B – ширина помещения.

Определим коэффициент использования светового потока по следующим данным:

- коэффициент отражения побеленного потолка (70%);
- коэффициент отражения от стен, окрашенных в светлую краску (50%);
- коэффициент отражения от пола, покрытого линолеумом темного цвета (10%);
- индекс помещения i :

$$i = \frac{A * B}{h * (A + B)} = \frac{6 * 4}{1,9 * (6 + 4)} = 1,26 ,$$

где A – длина помещения,

B – ширина помещения,

h – высота помещения.

Коэффициент светового потока $\eta = 0,34$.

Определим световой поток $F_{\text{общ}}$:

$$F_{\text{общ}} = \frac{300 * 24 * 1,26 * 1,5}{0,34} = 40023 \text{ лм.}$$

Для организации общего искусственного освещения из применяемых на практике ламп накаливания и газоразрядных (люминесцентных) отдадим предпочтение последним, т.к. они обеспечивают меньшее утомление органов зрения и организма в целом и способствуют повышению работоспособности и производительности труда. Кроме того, люминесцентные лампы имеют высокую световую отдачу, большой срок службы, малую яркость светящейся поверхности, спектр излучения близкий к естественному (солнечному). Следует учесть, что эксплуатация люминесцентных ламп обходится существенно дешевле, чем ламп накаливания.

Выбирая систему освещения, необходимо учитывать, что более эффективной является система комбинированного освещения, но система общего освещения более гигиенична, т.к. обеспечивает большую равномерность освещенности рабочих поверхностей. Используя локализованное общее освещение, можно наиболее просто добиться высоких уровней освещенностей на рабочих местах без значительных затрат. При повышенных требованиях к освещенности отдельных рабочих мест используют комбинированную систему освещения.

Для организации общего искусственного освещения выберем лампы типа ЛБ40. Световой поток одной лампы ЛБ40 составляет не менее $F_{\text{л}}=2810$ лм. Число N ламп, необходимых для организации общего освещения, определяется по формуле:

$$N = \frac{F_{\text{общ}}}{F_{\text{л}}} = \frac{40023}{2810} = 14$$

В качестве светильников выбираем ПВЛ-1, 2x40 Вт. Таким образом, чтобы обеспечить световой поток $F_{\text{общ}}=40023$ лм, надо использовать 7 светильников по 2 лампы ЛБ40 в каждом.

Электрическая мощность одной лампы ЛБ40 $W_{\text{л}}=40$ Вт.

Мощность всей осветительной системы:

$$W_{\text{общ}} = W_{\text{л}} * N = 40 * 14 = 560 \text{ Вт.}$$

С целью снижения уровня засветки экранов мониторов прямыми световыми потоками светильников общего назначения, последние следует располагать сбоку от рабочих мест, параллельно главной линии зрения операторов ПЭВМ и стене помещения с окнами. На рисунке 31 приведена возможная схема расположения светильников и ориентации рабочих мест операторов.

Из произведенного в данном разделе расчета следует, что для нормальной работы пользователя рабочего места с видеотерминальным устройством необходимо общее освещение помещения со световым потоком

40023 лм, для чего необходимо наличие 7 светильников типа ПВЛ-1 с двумя лампами типа ЛБ40. Кроме того, рекомендуется использовать ряд специальных мер по защите оператора от вредных факторов экрана дисплея, например, использование занавесок на окнах.

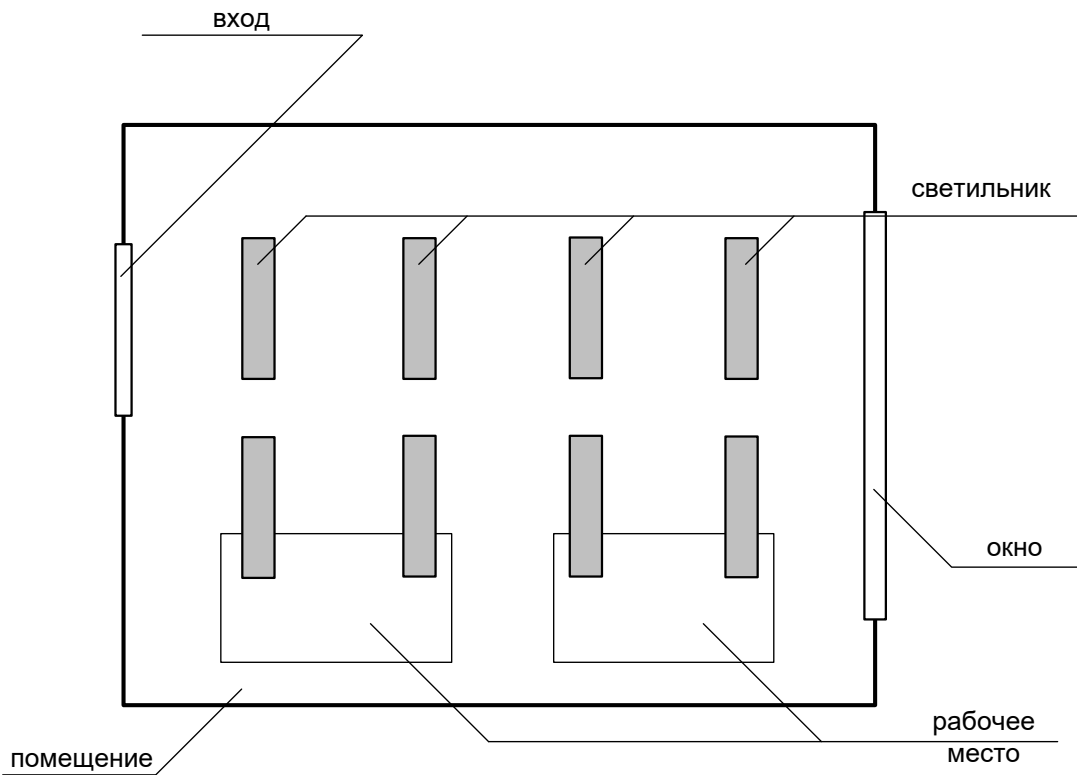


Рисунок 31 - Возможная схема расположения светильников и ориентации рабочих мест операторов

Заключение

В ходе выполнения выпускной квалификационной работы была разработана программная система для автоматической генерации моделей цифровых подстанций МЭК 61850, которая поддерживает предложенные методики автоматической генерации *MATLAB/SIMULINK*-моделей и функционально-блочных моделей распределенного управления МЭК 61499 на основе *SCL*-описаний цифровых подстанций. Достоинством системы является её простота в использовании, а также возможность расширения.

В результате выполнения выпускной квалификационной работы были достигнуты следующие результаты:

- разработана методика автоматической генерации *MATLAB/ SIMULINK* моделей цифровых подстанций МЭК 61850;
- разработана методика автоматической генерации функционально-блочных моделей распределенного управления цифровых подстанций МЭК 61850 на основе стандарта МЭК 61499;
- разработана программная система для поддержки предложенных методик автоматической генерации.

Направление дальнейших исследований - расширить систему автоматической генерации моделей следующим образом:

- добавить все типы оборудования стандарта МЭК 61850 в таблицу генерации *MATLAB/SIMULINK* моделей;
- расширить генерацию функционально-блочных моделей добавлением функциональных блоков логических узлов стандарта МЭК61850;
- разработка метода автоматической генерации системы взаимодействия *MATLAB/SIMULINK* моделей и функционально-блочных моделей через коммуникационную сеть.

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

57

Список использованных источников

1. Кобец Б. Б., Волкова И. О. Инновационное развитие электроэнергетики на базе концепции Smart Grid. – М.: ИАЦ Энергия, 2010. – 208 с.
2. Yang C.-H., Vyatkin V., Zhabelova G., Yang C-W. Co-Simulation Environment for Distributed Controls in SmartGrid // IEEE Transactions on Industrial Informatics. – 2013. – Vol. 9, iss. 3. – P. 1423 – 1435.
3. МЭК 61850: Communication networks and systems for power utility automation, I. E. Commission. – 2009.
4. Документация MATLAB. [Электронный ресурс]: Режим доступа: <https://www.mathworks.com/products/matlab.html>
5. Natsheh E., Albarbar A., Yazdani J. Modeling and control for smart grid integration of solar/wind energy conversion system // 2011 IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies. – 2011. – P. 1-8
6. Shlebig T., Fadel A., Mhereeg M., Shlebig M. The development of a simulation-based smart grid communication management system using MATLAB // 2017 IEEE International Conference on Green Energy Conversion Systems (GECS). – 2017. – P. 1 – 7.
7. Vyatkin V., Hanisch H., Pang C., Yang C. Closed-Loop Modeling in Future Automation System Engineering and Validation// IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). – 2009. – Vol. 39, Issue 1. – P. 17 – 28.
8. Vyatkin V. IEC 61499 as Enabler of Distributed and Intelligent Automation: State-of-the-Art Review // IEEE Transactions on Industrial Informatics. – 2011. – Vol. 7, Issue 4. – P. 768–781.
9. Guellouz S., Benzina A., Khalgui M., Frey G., Li Z., Vyatkin V. Designing Efficient Reconfigurable Control Systems Using IEC61499 and

Symbolic Model Checking // IEEE Transactions on Automation Science and Engineering. – 2019. – Vol. 16, Issue 3. – P. 1110–1124.

10. Vyatkin V. // Guest editorial: Special section on software engineering in industrial automation," IEEE Transactions on Industrial Informatics. – 2013. – Vol. 9, Issue 4. – P. 2337–2339.

11. Yang C.-W., Zhabelova G., Vyatkin V. // Sysgrid: МЭК 61850 and МЭК 61499 standard based engineering tool for smart grid automation design // EAI Endorsed Transactions on Energy Web. – 2014. – Vol. 3. – P. 1–8.

Приложение А

(Обязательное)

Система автоматической генерации моделей цифровых подстанций

МЭК61850

Листинг программы

					<i>ПГУ 1.09.03.01.35.001 ПЗ</i>	Лист
						60
Изм.	Лист	№ докум	Подпись	Дата		

Файл Program.cs

```
using System;
using System.Windows.Forms;

namespace ConverterAppllycation
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();

            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

61

Файл Form1.cs

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Xml;
using System.IO;
using System.IO.Compression;

namespace ConverterAppllycation
{
    public partial class Form1 : Form
    {
        List<Object_61850> _61850_objects;
        List<Object_matlab> MATLAB_objects;
        Object_61850 simple_object;
        List<string> fB_instances;

        XmlDocument xmldoc;

        public Form1()
        {
            InitializeComponent();

            _61850_objects = new List<Object_61850>();
            MATLAB_objects = new List<Object_matlab>();

            simple_object = new Object_61850(string.Empty,
string.Empty);

            fB_instances = new List<string>() {
"switch_instance",
"capacitor_instance","battery_instance","current_measurement_ins
tance", "volt_measurement_instance", "load_instance" };

```

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

62

```

    }

    private string TagString(XmlNode xNode) {
        string tag = "<" + xNode.Name;
        if (xNode.Attributes != null)
        {
            foreach (XmlAttribute atr in xNode.Attributes)
            {
                tag += " " + atr.Name + "=\"" + atr.Value +
"\\";
            }
        }
        tag += ">";
        return tag;
    }

    private void RenderXMLFile(string filepath)
    {
        try
        {
            xmldoc = new XmlDocument();

            string xmlString_1 =
File.ReadAllText("settingsdata.xml", Encoding.UTF8);

            xmldoc.LoadXml(xmlString_1);

            string xmlString = File.ReadAllText(filepath,
Encoding.UTF8);

            XmlDocument dom = new XmlDocument();
            dom.LoadXml(xmlString);
            treeView1.Nodes.Clear();

            treeView1.Nodes.Add(new
TreeNode(TagString(dom.DocumentElement)));

            TreeNode tNode = new TreeNode();
            tNode = treeView1.Nodes[0];

            this.AddNode(dom.DocumentElement, tNode);
        }
    }

```

```

        catch (XmlException xmlEx)
        {
            MessageBox.Show(xmlEx.Message);
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void AddNode(XmlNode inXmlNode, TreeNode
inTreeNode)
    {
        XmlNode xNode;
        TreeNode tNode;
        XmlNodeList nodeList;
        int i;
        bool has_initiated = false;
        if (inXmlNode.HasChildNodes)
        {
            nodeList = inXmlNode.ChildNodes;
            for (i = 0; i <= nodeList.Count - 1; i++)
            {
                xNode = inXmlNode.ChildNodes[i];
                if (has_initiated)
                {
                    _61850_objects.Add(simple_object);
                    simple_object = new
Object_61850(string.Empty, string.Empty);
                    has_initiated = false;
                }
                if (xNode.Name == "ConductingEquipment")
                {

```

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

64

```

        has_initiated = true;

        simple_object.Type =
xNode.Attributes.GetNamedItem("type").Value;

        simple_object.Name =
xNode.Attributes.GetNamedItem("name").Value;

        if
(xNode.Attributes.GetNamedItem("sxy:x") != null)

            simple_object.Xpos =
Convert.ToInt32(xNode.Attributes.GetNamedItem("sxy:x").Value);

            if
(xNode.Attributes.GetNamedItem("sxy:y") != null)

                simple_object.Ypos =
Convert.ToInt32(xNode.Attributes.GetNamedItem("sxy:y").Value);

MATLAB_objects.Add(ConfigurateMatlabObject(simple_object.Type,
simple_object.Name));

        foreach (XmlNode childnode in
xNode.ChildNodes)

            {

                if (childnode.Name ==

"Terminal")

                    simple_object.Connection
= childnode.Attributes.GetNamedItem("connectivityNode").Value;

            }

        }

        inTreeNode.Nodes.Add(new
TreeNode(TagString(xNode)));

        tNode = inTreeNode.Nodes[i];

        this.AddNode(xNode, tNode);

    }

}

else {

    inTreeNode.Text = TagString(inXmlNode);

}

}

```

```

        private void GenerateMatlabScript(List<Object_61850>
objects_61850)
        {
            string generated_file_path =
Directory.GetCurrentDirectory() + @" \SCD_2_MATLAB.m";
            string script = "";
            script = InitializeMatlabScript();
            script = AddObjectsToScript(script, objects_61850);
            script = AddConnectionsToScript(script,
objects_61850);
            script = InitializeSolverInScript(script);

            richTextBox1.Text = script;
        }

        private Object_matlab ConfigurateMatlabObject(string
object_type, string name)
        {
            Object_matlab local_matlab_obj = null;

            XmlNode NodeGen =
xmldoc.SelectSingleNode("configuration");

            foreach (XmlNode childNode in NodeGen.ChildNodes)
            {
                if
(childNode.Attributes.GetNamedItem("name").Value == object_type)
                {
                    if
(childNode.Attributes.GetNamedItem("ports").Value == "one side")

                        local_matlab_obj = new
Object_matlab(childNode.Attributes.GetNamedItem("path").Value,
name, "/LConn1", "/LConn1", "/LConn1", "/LConn1", "/LConn1",
"/LConn1", true, true);
                }
            }
        }
    }
}

```

```

        else if
            (childNodes.Attributes.GetNamedItem("ports").Value == "two
side/three phase")

                local_matlab_obj = new
Object_matlab(childNode.Attributes.GetNamedItem("path").Value,
name, "/LConn1", "/LConn2", "/LConn3", "/RConn1", "/RConn2",
"/RConn3", true, true);

            else

                local_matlab_obj = new
Object_matlab(childNode.Attributes.GetNamedItem("path").Value,
name, "/LConn1", "/LConn1", "/LConn1", "/LConn2", "/LConn2",
"/LConn2", true, true);

            }

        }

        return local_matlab_obj;

    }

    private string InitializeMatlabScript()
    {

        string script =

@"function SCD_2_MATLAB
% function which demonstrates generation of Matlab script

% Name of the model
fname = 'SCD_2_MATLAB';

% Check if the file already exists and delete it if it does
if exist(fname,'file') == 4

    % If it does then check whether it's open
    if bdIsLoaded(fname)

        % If it is then close it (without saving!)
        close_system(fname,0)

    end

    % delete the file
    delete([fname, '.slx']);

end

```

```

% Create the system
new_system(fname);" + "\n";

        return script;
    }

    private string InitializeSolverInScript(string _script)
    {
        _script +=
@"% Set a couple of model parameters to eliminate warning
messages
set_param(gcs,...
    'Solver','FixedStepDiscrete',...
    'FixedStep','0.1');

% Save the model
save_system(fname);"

        return _script;
    }

    private string AddObjectsToScript(string _script,
List<Object_61850> objects_61850)
    {
        int block_Y = 50;
        int block_X = 100;
        int Yoffset = 80;
        int Xoffset = 80;
        int StartX = 200;
        int StartY = 200;

```

```

        foreach (Object_61850 _61850_object in
_61850_objects)
        {
            XmlNode NodeGen =
xmldoc.SelectSingleNode("configuration");

            foreach (XmlNode childNode in
NodeGen.ChildNodes)
            {
                if
                (childNode.Attributes.GetNamedItem("name").Value ==
_61850_object.Type)
                {
                    _script += "add_block('" +
childNode.Attributes.GetNamedItem("path").Value + "', [gcs, '/' +
_61850_object.Name + "'], ... \n";

                    foreach (XmlNode childNode_1 in
childNode.ChildNodes)
                    {
                        _script += childNode_1 .InnerText+
", ... \n";
                    }

                    int countStartY = StartY + (Yoffset *
_61850_object.Ypos);

                    int countStartYOffset = countStartY +
block_Y;

                    int countStartX = StartX + (Xoffset *
_61850_object.Xpos);

                    int countStartXOffset = countStartX +
block_X;

                    _script += "'Position', [" + countStartX
+ " " + countStartY + " " + countStartXOffset + " " +
countStartYOffset + "]); \n";

                    break;
                }
            }
        }

```

```

    }

    return _script;
}

private string AddConnectionsToScript(string _script,
List<Object_61850> objects_61850)
{
    List<Object_61850> connection_list = new
List<Object_61850>();

    string component_side = string.Empty;
    Object_matlab start_obj = null;
    Object_matlab end_obj = null;

    foreach (Object_61850 _61850_object in
_61850_objects)
    {
        if (_61850_object.Connection != null)
        {
            connection_list.Add(_61850_object);
            foreach (Object_61850 _61850_object_compare
in _61850_objects)
            {
                if (_61850_object.Name !=
_61850_object_compare.Name)
                {
                    if
(_61850_object_compare.ConnectionFind(_61850_object.Connection))
                    {
                        connection_list.Add(_61850_object_compare);

_61850_object_compare.ConnectionRemove(_61850_object.Connection)
;
                    }
                }
            }
        }
    }
}

```

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

70

```

_61850_object.ConnectionRemove(_61850_object.Connection);

        if (connection_list.Count > 1)
        {
            Object_61850 basic_node =
connection_list.First();

            foreach (Object_61850 _61850_object_name
in connection_list)
            {
                if (basic_node.Name !=
_61850_object_name.Name)
                {
                    start_obj =
MATLAB_objects.Find(x => x.Name.Contains(basic_node.Name));

                    end_obj = MATLAB_objects.Find(x
=> x.Name.Contains(_61850_object_name.Name));

                    if (start_obj.LConnStatus)
                    {
                        if (end_obj.LConnStatus)
                        {
                            component_side = "LL";

                            if (start_obj.Port1 ==
start_obj.Port2 && start_obj.Port2 == start_obj.Port3 &&
end_obj.Port1 == end_obj.Port2 && end_obj.Port2 ==
end_obj.Port3)

                                _script +=

@" add_line(gcs, '"' + start_obj.Name + start_obj.Port2 + "', '"
+ end_obj.Name + end_obj.Port2 + @''); " + "\n";

                                else

                                    _script +=

@"add_line(gcs, '"' + start_obj.Name + start_obj.Port1 + "', '"
+ end_obj.Name + end_obj.Port1 + @'');

add_line(gcs, '"' + start_obj.Name + start_obj.Port2 + "', '"
+ end_obj.Name + end_obj.Port2 + @'');

```

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

71

```

add_line(gcs, '"' + start_obj.Name + start_obj.Port3 + "', '" +
end_obj.Name + end_obj.Port3 + @'');" + "\n";

    }

    else if
(end_obj.RConnStatus)

    {

        component_side = "LR";

        if (start_obj.Port1 ==
start_obj.Port2 && start_obj.Port2 == start_obj.Port3 &&
end_obj.Port4 == end_obj.Port5 && end_obj.Port5 ==
end_obj.Port6)

            _script +=

@" add_line(gcs, '"' + start_obj.Name + start_obj.Port2 + "', '"
+ end_obj.Name + end_obj.Port5 + @''); " + "\n";

            else

                _script +=

@"add_line(gcs,'" + start_obj.Name + start_obj.Port1 + "', '" +
end_obj.Name + end_obj.Port4 + @'');

add_line(gcs,'" + start_obj.Name + start_obj.Port2 + "', '" +
end_obj.Name + end_obj.Port5 + @'');

add_line(gcs, '"' + start_obj.Name + start_obj.Port3 + "', '" +
end_obj.Name + end_obj.Port6 + @'');" + "\n";

            }

    }

    else if (start_obj.RConnStatus)

    {

        if (end_obj.LConnStatus)

        {

            component_side = "RL";

            if (start_obj.Port4 ==
start_obj.Port5 && start_obj.Port5 == start_obj.Port6 &&
end_obj.Port1 == end_obj.Port2 && end_obj.Port2 ==
end_obj.Port3)

                _script +=

@" add_line(gcs, '"' + start_obj.Name + start_obj.Port5 + "', '"
+ end_obj.Name + end_obj.Port2 + @''); " + "\n";

            else

```

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

72

```

                                _script +=

@"add_line(gcs,'" + start_obj.Name + start_obj.Port4 + "',' +
end_obj.Name + end_obj.Port1 + @'');

add_line(gcs,'" + start_obj.Name + start_obj.Port5 + "',' +
end_obj.Name + end_obj.Port2 + @'');

add_line(gcs, '" + start_obj.Name + start_obj.Port6 + "',' +
end_obj.Name + end_obj.Port3 + @'');" + "\n";

                                }

                                else if

(end_obj.RConnStatus)

                                {

                                component_side = "RR";

                                if (start_obj.Port4 ==
start_obj.Port5 && start_obj.Port5 == start_obj.Port6 &&
end_obj.Port4 == end_obj.Port5 && end_obj.Port5 ==
end_obj.Port6)

                                _script +=

@" add_line(gcs, '" + start_obj.Name + start_obj.Port5 + "',' +
+ end_obj.Name + end_obj.Port5 + @'');" + "\n";

                                else

                                _script +=

@"add_line(gcs,'" + start_obj.Name + start_obj.Port4 + "',' +
end_obj.Name + end_obj.Port4 + @'');

add_line(gcs,'" + start_obj.Name + start_obj.Port5 + "',' +
end_obj.Name + end_obj.Port5 + @'');

add_line(gcs, '" + start_obj.Name + start_obj.Port6 + "',' +
end_obj.Name + end_obj.Port6 + @'');" + "\n";

                                }

                                }

                                }

                                }

                                switch (component_side)

                                {

                                case "LL":

                                start_obj.LConnStatus = false;

```

```

        end_obj.LConnStatus = false;
        component_side = string.Empty;
        break;
    case "LR":
        start_obj.LConnStatus = false;
        end_obj.RConnStatus = false;
        component_side = string.Empty;
        break;
    case "RL":
        start_obj.RConnStatus = false;
        end_obj.LConnStatus = false;
        component_side = string.Empty;
        break;
    case "RR":
        start_obj.RConnStatus = false;
        end_obj.RConnStatus = false;
        component_side = string.Empty;
        break;
    default:
        MessageBox.Show("Error with
block connection");

        break;
    }
}
connection_list.Clear();
}
}
return _script;
}

```

```

private void openSCDToolStripMenuItem_Click(object
sender, EventArgs e)

```

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

74

```

{
    string destination_filename = string.Empty;
    using (var fbd = new OpenFileDialog())
    {
        fbd.Filter = "SCD Files (*.scd)|*.scd";
        DialogResult result = fbd.ShowDialog();
        if (result == DialogResult.OK)
        {
            _61850_objects.Clear();
            MATLAB_objects.Clear();
            destination_filename =
fbd.FileName.ToString();
            this.RenderXMLFile(destination_filename);
            GenerateMatlabScript(_61850_objects);
        }
    }
}

private void saveScriptToolStripMenuItem_Click(object
sender, EventArgs e)
{
    SaveFileDialog saveFileDialog1 = new
SaveFileDialog();

    saveFileDialog1.FileName = "SCD_2_MATLAB";

    saveFileDialog1.DefaultExt = ".m";
    saveFileDialog1.Filter = "Model files (*.m)|*.m";
    saveFileDialog1.FilterIndex = 2;
    saveFileDialog1.RestoreDirectory = true;

    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        string filename = saveFileDialog1.FileName;

```

```

        File.WriteAllText(filename, richTextBox1.Text);
    }
}

private void settingsToolStripMenuItem_Click(object
sender, EventArgs e)
{
    Form ifrm = new SettingsForm();
    ifrm.ShowDialog();
}

private void Initialize_61499_project(string
extractPath, string project_name)
{
    ZipFile.ExtractToDirectory(Directory.GetCurrentDirectory() +
@"\IEC61499.zip", extractPath);

    File.Move(extractPath + @"no_name.nxtsln",
extractPath + @"\" + project_name + ".nxtsln");

    File.Move(extractPath + @"no_name.sln", extractPath
+ @"\" + project_name + ".sln");
}

private void Initialize_61499_DFBPROJ(string
local_file_path)
{
    foreach (var fb_instance in fb_instances)
    {
        XmlDocument IEC61499_dfbproj = new
XmlDocument();

        IEC61499_dfbproj.Load(local_file_path +
@"\IEC61499\IEC61499.dfbproj");

        var namespaceManager = new
XmlNamespaceManager(IEC61499_dfbproj.NameTable);

```

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

76

```

        namespaceManager.AddNamespace("ns",
"http://schemas.microsoft.com/developer/msbuild/2003");

        XmlElement ItemGroup =
(XmlElement)IEC61499_dfbproj.SelectSingleNode("//ns:Project//ns:
ItemGroup//ns:None", namespaceManager).ParentNode;

        if (ItemGroup == null)

            MessageBox.Show("Ups!");

        XmlElement None_object =
IEC61499_dfbproj.CreateElement("None",
"http://schemas.microsoft.com/developer/msbuild/2003");

        None_object.SetAttribute("Include",
fb_instance+@"\"+ fb_instance + ".cfg");

        XmlElement dependentUpon_1 =
IEC61499_dfbproj.CreateElement("DependentUpon",
"http://schemas.microsoft.com/developer/msbuild/2003");

        dependentUpon_1.InnerText = fb_instance+".fbt";

        XmlElement IEC61499Type_1 =
IEC61499_dfbproj.CreateElement("IEC61499Type",
"http://schemas.microsoft.com/developer/msbuild/2003");

        IEC61499Type_1.InnerText = "CAT";

        None_object.AppendChild(dependentUpon_1);

        None_object.AppendChild(IEC61499Type_1);

        ItemGroup.AppendChild(None_object);


        XmlElement compile_object =
IEC61499_dfbproj.CreateElement("Compile",
"http://schemas.microsoft.com/developer/msbuild/2003");

        compile_object.SetAttribute("Include",
fb_instance + @"\" + fb_instance + ".fbt");

        XmlElement IEC61499Type_2 =
IEC61499_dfbproj.CreateElement("IEC61499Type",
"http://schemas.microsoft.com/developer/msbuild/2003");

        IEC61499Type_2.InnerText = "CAT";

        compile_object.AppendChild(IEC61499Type_2);

        ItemGroup.AppendChild(compile_object);

```

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

77

```

        XmlElement compile_object_2 =
IEC61499_dfbproj.CreateElement("Compile",
"http://schemas.microsoft.com/developer/msbuild/2003");

        compile_object_2.SetAttribute("Include",
fb_instance + @"\" + fb_instance + "_HMI.fbt");

        XmlElement IEC61499Type_3 =
IEC61499_dfbproj.CreateElement("IEC61499Type",
"http://schemas.microsoft.com/developer/msbuild/2003");

        IEC61499Type_3.InnerText = "CAT";

        compile_object_2.AppendChild(IEC61499Type_3);

        XmlElement usage =
IEC61499_dfbproj.CreateElement("Usage",
"http://schemas.microsoft.com/developer/msbuild/2003");

        usage.InnerText = "Private";

        compile_object_2.AppendChild(usage);

        XmlElement dependentUpon_2 =
IEC61499_dfbproj.CreateElement("DependentUpon",
"http://schemas.microsoft.com/developer/msbuild/2003");

        dependentUpon_2.InnerText = fb_instance+".fbt";

        compile_object_2.AppendChild(dependentUpon_2);

        XmlElement hmi =
IEC61499_dfbproj.CreateElement("HMI",
"http://schemas.microsoft.com/developer/msbuild/2003");

        hmi.InnerText = @"..\HMI\"+fb_instance+@"\"+
fb_instance + "_sDefault.cnv.cs";

        compile_object_2.AppendChild(hmi);

        ItemGroup.AppendChild(compile_object_2);

        IEC61499_dfbproj.Save(local_file_path +
@"\IEC61499\IEC61499.dfbproj");

    }

}

private void Initialize_HMI_CSPROJ(string
local_file_path)

{

    XmlDocument HMI_csproj = new XmlDocument();

```

```

        HMI_csproj.Load(local_file_path +
@"\HMI\HMI.csproj");

        var namespaceManager = new
XmlNamespaceManager(HMI_csproj.NameTable);

        namespaceManager.AddNamespace("ns",
"http://schemas.microsoft.com/developer/msbuild/2003");

        XmlElement ItemGroup =
(XmlElement)HMI_csproj.SelectSingleNode("//ns:Project//ns:ItemGr
oup//ns:None", namespaceManager).ParentNode;

        foreach (var fb_instance in fB_instances)
        {

            XmlElement compile =
HMI_csproj.CreateElement("Compile",
"http://schemas.microsoft.com/developer/msbuild/2003");

            compile.SetAttribute("Include", fb_instance +
@"\"+ fb_instance + ".def.cs");

            XmlElement compile_1 =
HMI_csproj.CreateElement("Compile",
"http://schemas.microsoft.com/developer/msbuild/2003");

            compile_1.SetAttribute("Include",
fb_instance+@"\"+ fb_instance + ".event.cs");

            XmlElement compile_2 =
HMI_csproj.CreateElement("Compile",
"http://schemas.microsoft.com/developer/msbuild/2003");

            compile_2.SetAttribute("Include",
fb_instance+@"\"+ fb_instance + "_sDefault.cnv.cs");

            XmlElement compile_3 =
HMI_csproj.CreateElement("Compile",
"http://schemas.microsoft.com/developer/msbuild/2003");

            compile_3.SetAttribute("Include",
fb_instance+@"\"+ fb_instance + "_sDefault.cnv.Designer.cs");

            XmlElement DependentUpon_1 =
HMI_csproj.CreateElement("DependentUpon",
"http://schemas.microsoft.com/developer/msbuild/2003");

            DependentUpon_1.InnerText =
fb_instance+"_sDefault.cnv.cs";

            compile_3.AppendChild(DependentUpon_1);

            ItemGroup.AppendChild(compile);

```

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

79

```

        ItemGroup.AppendChild(compile_1);
        ItemGroup.AppendChild(compile_2);
        ItemGroup.AppendChild(compile_3);

        XmlElement None_object_1 =
HMI_csproj.CreateElement("None",
"http://schemas.microsoft.com/developer/msbuild/2003");

        None_object_1.SetAttribute("Include",
fb_instance+@"\"+ fb_instance + ".Design.resx");

        ItemGroup.AppendChild(None_object_1);

        XmlElement embeddedResource_1 =
HMI_csproj.CreateElement("EmbeddedResource",
"http://schemas.microsoft.com/developer/msbuild/2003");

        embeddedResource_1.SetAttribute("Include",
fb_instance+@"\"+ fb_instance + "_sDefault.cnv.resx");

        XmlElement DependentUpon_2 =
HMI_csproj.CreateElement("DependentUpon",
"http://schemas.microsoft.com/developer/msbuild/2003");

        DependentUpon_2.InnerText =
fb_instance+"_sDefault.cnv.cs";

        embeddedResource_1.AppendChild(DependentUpon_2);
        ItemGroup.AppendChild(embeddedResource_1);

        XmlElement embeddedResource_2 =
HMI_csproj.CreateElement("EmbeddedResource",
"http://schemas.microsoft.com/developer/msbuild/2003");

        embeddedResource_2.SetAttribute("Include",
fb_instance+@"\"+ fb_instance + "_sDefault.cnv.xml");

        XmlElement DependentUpon_3 =
HMI_csproj.CreateElement("DependentUpon",
"http://schemas.microsoft.com/developer/msbuild/2003");

        DependentUpon_3.InnerText =
fb_instance+"_sDefault.cnv.cs";

        embeddedResource_2.AppendChild(DependentUpon_3);
        ItemGroup.AppendChild(embeddedResource_2);
    }

```

```

        XmlElement compile_4 =
HMI_csproj.CreateElement("Compile",
"http://schemas.microsoft.com/developer/msbuild/2003");

        compile_4.SetAttribute("Include",
@"Canvas1.cnv.cs");

        XmlElement canvas =
HMI_csproj.CreateElement("Canvas",
"http://schemas.microsoft.com/developer/msbuild/2003");

        canvas.InnerText = "true";

        compile_4.AppendChild(canvas);

        ItemGroup.AppendChild(compile_4);


        XmlElement compile_5 =
HMI_csproj.CreateElement("Compile",
"http://schemas.microsoft.com/developer/msbuild/2003");

        compile_5.SetAttribute("Include",
@"Canvas1.cnv.Designer.cs");

        XmlElement DependentUpon_4 =
HMI_csproj.CreateElement("DependentUpon",
"http://schemas.microsoft.com/developer/msbuild/2003");

        DependentUpon_4.InnerText = "Canvas1.cnv.cs";

        compile_5.AppendChild(DependentUpon_4);

        ItemGroup.AppendChild(compile_5);


        XmlElement embeddedResource_3 =
HMI_csproj.CreateElement("EmbeddedResource",
"http://schemas.microsoft.com/developer/msbuild/2003");

        embeddedResource_3.SetAttribute("Include",
@"Canvas1.cnv.resx");

        XmlElement DependentUpon_5 =
HMI_csproj.CreateElement("DependentUpon",
"http://schemas.microsoft.com/developer/msbuild/2003");

        DependentUpon_5.InnerText = "Canvas1.cnv.cs";

        embeddedResource_3.AppendChild(DependentUpon_5);

        ItemGroup.AppendChild(embeddedResource_3);

```

```

        HMI_csproj.Save(local_file_path +
@"\HMI\HMI.csproj");
    }

    private void
Initialize_SYSTEM_FOLDER_according_to_SCD_object(string
local_file_path)
    {
        XmlDocument system_sys = new XmlDocument();
        XmlDocument system_cfg = new XmlDocument();

        system_sys.Load(local_file_path +
@"\IEC61499\System\System.sys");

        system_cfg.Load(local_file_path +
@"\IEC61499\System\System.cfg");

        var namespaceManager = new
XmlNamespaceManager(system_cfg.NameTable);

        namespaceManager.AddNamespace("ns",
"http://www.nxtcontrol.com/IEC61499.xsd");

        Random random = new Random();

        const string chars =
"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";

        const int id_length = 16;

        string generated_ID = "";

        foreach (Object_61850 _61850_object in
_61850_objects)
        {
            if (_61850_object.Type == "CBR")
            {
                generated_ID = new
string(Enumerable.Repeat(chars, id_length).Select(s =>
s[random.Next(s.Length)]).ToArray());

                Initialize_SYSTEM_FOLDER(system_sys,
system_cfg, namespaceManager, "switch_instance",
_61850_object.Name, generated_ID);
            }
        }
    }

```

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

82

```

else if (_61850_object.Type == "CAP")
{
    generated_ID = new
string(Enumerable.Repeat(chars, id_length).Select(s =>
s[random.Next(s.Length)]).ToArray());

    Initialize_SYSTEM_FOLDER(system_sys,
system_cfg, namespaceManager, "capacitor_instance",
_61850_object.Name, generated_ID);
}

else if (_61850_object.Type == "BAT")
{
    generated_ID = new
string(Enumerable.Repeat(chars, id_length).Select(s =>
s[random.Next(s.Length)]).ToArray());

    Initialize_SYSTEM_FOLDER(system_sys,
system_cfg, namespaceManager, "battery_instance",
_61850_object.Name, generated_ID);
}

else if (_61850_object.Type == "CTR")
{
    generated_ID = new
string(Enumerable.Repeat(chars, id_length).Select(s =>
s[random.Next(s.Length)]).ToArray());

    Initialize_SYSTEM_FOLDER(system_sys,
system_cfg, namespaceManager, "current_measurement_instance",
_61850_object.Name, generated_ID);
}

else if (_61850_object.Type == "VTR")
{
    generated_ID = new
string(Enumerable.Repeat(chars, id_length).Select(s =>
s[random.Next(s.Length)]).ToArray());

    Initialize_SYSTEM_FOLDER(system_sys,
system_cfg, namespaceManager, "volt_measurement_instance",
_61850_object.Name, generated_ID);
}

else if (_61850_object.Type == "PSH")
{

```

```

        generated_ID = new
string(Enumerable.Repeat(chars, id_length).Select(s =>
s[random.Next(s.Length)]).ToArray());

        Initialize_SYSTEM_FOLDER(system_sys,
system_cfg, namespaceManager, "load_instance",
_61850_object.Name, generated_ID);
    }
}

    system_sys.Save(local_file_path +
@"\IEC61499\System\System.sys");

    system_cfg.Save(local_file_path +
@"\IEC61499\System\System.cfg");
}

private void Initialize_SYSTEM_FOLDER(XmlDocument
system_sys, XmlDocument system_cfg, XmlNamespaceManager
namespaceManager, string fb_instance_name, string
scd_object_name, string object_id)
{

    XmlElement FBNetwork =
(XmlElement)system_sys.SelectSingleNode("//Resource/FBNetwork");

    XmlElement System =
(XmlElement)system_cfg.SelectSingleNode("//ns:SystemDatabase//ns
:System", namespaceManager);

    XmlElement add_CATType = null;

    XmlElement new_ResInst = null;

    XmlElement FB = system_sys.CreateElement("FB");
    FB.SetAttribute("ID", object_id);
    FB.SetAttribute("Name", scd_object_name);
    FB.SetAttribute("Type", fb_instance_name);
    FB.SetAttribute("x", "12");
    FB.SetAttribute("y", "16");
    FB.SetAttribute("Namespace", "Main");

```

```

        FB.SetAttribute("Loaded", "true");

        XmlElement param =
system_sys.CreateElement("Parameter");

        param.SetAttribute("Name", "QI");
        param.SetAttribute("Value", "TRUE");

        FB.AppendChild(param);

        FBNetwork.AppendChild(FB);

        if
((XmlElement)system_sys.SelectSingleNode("//Resource/FBNetwork/E
ventConnections") == null)
        {

            XmlElement event_connections =
system_sys.CreateElement("EventConnections");

            XmlElement connection =
system_sys.CreateElement("Connection");

            connection.SetAttribute("Source", "START.COLD");

            connection.SetAttribute("Destination",
scd_object_name + ".INIT");

            event_connections.AppendChild(connection);

            FBNetwork.AppendChild(event_connections);

        }
        else
        {

            XmlElement connection =
system_sys.CreateElement("Connection");

            connection.SetAttribute("Source", "START.COLD");

            connection.SetAttribute("Destination",
scd_object_name + ".INIT");

            ((XmlElement)system_sys.SelectSingleNode("//Resource/FBNetwork/E
ventConnections")).AppendChild(connection);

        }

```

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

85

```

        if
        ((XmlElement)system_cfg.SelectSingleNode("//ns:SystemDatabase//n
s:System//ns:CATType", namespaceManager) == null)

        {

            add_CATType =
system_cfg.CreateElement("CATType",
"http://www.nxtcontrol.com/IEC61499.xsd");

            add_CATType.SetAttribute("Name",
fb_instance_name);

            add_CATType.SetAttribute("Namespace", "Main");

            new_ResInst =
system_cfg.CreateElement("ResInst",
"http://www.nxtcontrol.com/IEC61499.xsd");

            new_ResInst.SetAttribute("ID", object_id);

            new_ResInst.SetAttribute("Name", "Res-DEV0-RES0-
" + scd_object_name);

            new_ResInst.SetAttribute("Map",
"2371D82C743A513C.28A29A086E15ACB3;");

            add_CATType.AppendChild(new_ResInst);

            System.AppendChild(add_CATType);

        }

        else

        {

            new_ResInst =
system_cfg.CreateElement("ResInst",
"http://www.nxtcontrol.com/IEC61499.xsd");

            new_ResInst.SetAttribute("ID", object_id);

            new_ResInst.SetAttribute("Name", "Res-DEV0-RES0-
" + scd_object_name);

            new_ResInst.SetAttribute("Map",
"2371D82C743A513C.28A29A086E15ACB3;");

            ((XmlElement)system_cfg.SelectSingleNode("//ns:SystemDatabase//n
s:System//ns:CATType",
namespaceManager)).AppendChild(new_ResInst);

        }

    }

```

```

        private void
create61499ProjectToolStripMenuItem1_Click(object sender,
EventArgs e)
    {
        string local_file_path =
Directory.GetCurrentDirectory();

        string project_name = "hello_61499";

        string directory_name = local_file_path + @"\" +
project_name;

        if (!Directory.Exists(directory_name))
        {
            Directory.CreateDirectory(directory_name);
            local_file_path = directory_name;
            Initialize_61499_project(local_file_path,
project_name);

            Initialize_61499_DFBPROJ(local_file_path);
            Initialize_HMI_CSPROJ(local_file_path);

            Initialize_SYSTEM_FOLDER_according_to_SCD_object(local_file_path
);

        }
        else
        {
            MessageBox.Show("Directory with such name is
already exist. Please, specify a different name.");
        }
    }
}

```

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

87

Файл Object_matlab.cs

```

namespace ConverterAppllycation
{
    class Object_matlab
    {
private string name;
        private bool LConn_status;
        private bool RConn_status;
        public List<string> LPorts;
        public List<string> RPorts;

        public Object_matlab( string name)
        {
            this.name = name;

            this.LConn_status = false;
            this.RConn_status = false;
            LPorts = new List<string>();
            RPorts = new List<string>();

        }
        public int LPortsCount
        {
            set {
                LPorts = new List<string>();
                if (value > 0) {
                    LConn_status = true;
                    for (int i = 1; i < value + 1; i++)
                    {
                        LPorts.Add("/LConn"+i);
                    }
                }
            }
            get {
                return LPorts.Count;
            }
        }

        public int RPortsCount
        {
            set
            {
                RPorts = new List<string>();
                RConn_status = true;
                if (value > 0)
                {
                    for (int i = 1; i < value + 1; i++)
                    {
                        RPorts.Add("/RConn" + i);
                    }
                }
            }
        }
    }
}

```

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

88

```

        get
        {
            return LPorts.Count;
        }
    }
    public string Name
    {
        get { return name; }
        set { name = value; }
    }

    public bool LConnStatus
    {
        get { return LConn_status; }
        set { LConn_status = value; }
    }
    public bool RConnStatus
    {
        get { return RConn_status; }
        set { RConn_status = value; }
    }
}
}

```

Файл Object_61850.cs

```
using System.Collections.Generic;
using System.Linq;

namespace ConverterAppllycation
{
    class Object_61850
    {
        private string name;
        private string type;
        int x;
        int y;
        private List<string> con;

        public Object_61850(string type, string name)
        {
            this.type = type;
            this.name = name;
            x = 0;
            y = 0;
            this.con = new List<string>();
        }

        public int Xpos
        {
            get { return x; }
            set { x = value; }
        }

        public int Ypos
        {
            get { return y; }
            set { y = value; }
        }

        public string Type
        {
            get { return type; }
            set { type = value; }
        }

        public string Name
        {
            get { return name; }
            set { name = value; }
        }

        public string Connection
        {
            get {
                if (con.Count > 0)
                    return con.First();
                else
                    return null;
            }
        }
    }
}
```

Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

90

```

        set { con.Add(value); }
    }

    public void ConnectionRemove(string Connection)
    {
        con.Remove(Connection);
    }

    public bool ConnectionFind(string Connection)
    {
        return con.Contains(Connection);
    }
}
}

```

Файл SettingsForm.cs

```
using System;
using System.Text;
using System.Windows.Forms;
using System.Xml;
using System.IO;

namespace ConverterAppllycation
{
    public partial class SettingsForm : Form
    {
        XmlDocument xmldoc;
        public SettingsForm()
        {
            InitializeComponent();
            xmldoc = new XmlDocument();
            string xmlString =
File.ReadAllText("settingsdata.xml", Encoding.UTF8);
            xmldoc.LoadXml(xmlString);
            XmlNode NodeGen =
xmldoc.SelectSingleNode("configuration");
            foreach(XmlNode childNode in NodeGen.ChildNodes)

comboBox1.Items.Add(childNode.Attributes.GetNamedItem("name").Value);
        }

        private void button2_Click(object sender, EventArgs e)
        {
            bool checker = true;
            string selected_element = comboBox1.Text;
            if (selected_element != string.Empty)
            {
                XmlNode NodeGen =
xmldoc.SelectSingleNode("configuration");
                foreach (XmlNode childNode in
NodeGen.ChildNodes)
                {
                    if
(childNode.Attributes.GetNamedItem("name").Value ==
selected_element)
                    {
                        MessageBox.Show("Element exist");
                        checker = false;
                        break;
                    }
                }
                if (checker)
                {
                    XmlNode SCDElement =
xmldoc.CreateNode(XmlNodeType.Element, "SCDElement", null);
```

```

        XmlAttribute xa =
xmlldoc.CreateAttribute("name");
        xa.Value = selected_element;
        SCDElement.Attributes.Append(xa);
        xa = xmlldoc.CreateAttribute("path");
        xa.Value = string.Empty;
        SCDElement.Attributes.Append(xa);
        xa = xmlldoc.CreateAttribute("ports");
        xa.Value = "one side";
        SCDElement.Attributes.Append(xa);
        NodeGen.AppendChild(SCDElement);
        xmlldoc.Save("settingsdata.xml");
        comboBox1.Items.Clear();
        foreach (XmlNode childNode in
NodeGen.ChildNodes)

comboBox1.Items.Add(childNode.Attributes.GetNamedItem("name").Value);

        comboBox1.SelectedItem = selected_element;
    }
}
else
    MessageBox.Show("Empty field!!!");
}

private void comboBox1_SelectedIndexChanged(object
sender, EventArgs e)
{
    string selected_element =
(string)comboBox1.SelectedItem;
    listBox1.Items.Clear();
    XmlNode NodeGen =
xmlldoc.SelectSingleNode("configuration");
    foreach (XmlNode childNode in NodeGen.ChildNodes)
    {
        if
(childNode.Attributes.GetNamedItem("name").Value ==
selected_element)
        {
            textBox1.Text =
childNode.Attributes.GetNamedItem("path").Value;
            if
(childNode.Attributes.GetNamedItem("ports").Value == "one side")
                comboBox2.SelectedIndex = 0;
            else if
(childNode.Attributes.GetNamedItem("ports").Value == "two side")
                comboBox2.SelectedIndex = 1;
            else
                comboBox2.SelectedIndex = 2;
            foreach (XmlNode childNode_1 in
childNode.ChildNodes)
            {

```

```

listBox1.Items.Add(childNode_1.InnerText);
        }
    }
}

private void comboBox2_SelectedIndexChanged(object
sender, EventArgs e)
{
    string selected_element =
(string)comboBox1.SelectedItem;
    XmlNode NodeGen =
xmldoc.SelectSingleNode("configuration");
    foreach (XmlNode childNode in NodeGen.ChildNodes)
    {
        if
(childNode.Attributes.GetNamedItem("name").Value ==
selected_element)
        {
            childNode.Attributes["ports"].Value =
(string)comboBox2.SelectedItem;
            xmldoc.Save("settingsdata.xml");
        }
    }

private void button5_Click(object sender, EventArgs e)
{
    //кнопка save
    string selected_element =
(string)comboBox1.SelectedItem;
    XmlNode NodeGen =
xmldoc.SelectSingleNode("configuration");
    foreach (XmlNode childNode in NodeGen.ChildNodes)
    {
        if
(childNode.Attributes.GetNamedItem("name").Value ==
selected_element)
        {
            childNode.Attributes["path"].Value =
textBox1.Text;
            xmldoc.Save("settingsdata.xml");
        }
    }

private void button1_Click(object sender, EventArgs e)
{
    //добавить элемент списка
    listBox1.Items.Add(textBox2.Text);
    string selected_element =
(string)comboBox1.SelectedItem;

```

```

        XmlNode NodeGen =
xmlldoc.SelectSingleNode("configuration");
        foreach (XmlNode childNode in NodeGen.ChildNodes)
        {
            if
(xmlldoc.SelectSingleNode("configuration").ChildNodes.Count ==
selected_element)
            {
                XmlNode New_Node =
xmlldoc.CreateNode(XmlNodeType.Element, "parameter", null);
                New_Node.InnerText = textBox2.Text;
                childNode.AppendChild(New_Node);
                xmlldoc.Save("settingsdata.xml");
            }
        }
    }

private void button4_Click(object sender, EventArgs e)
{
    //удалить элемент списка
    if (listBox1.SelectedItems.Count != 0)
    {
        string selected_element =
(string)comboBox1.SelectedItem;
        XmlNode NodeGen =
xmlldoc.SelectSingleNode("configuration");
        foreach (XmlNode childNode in
NodeGen.ChildNodes)
        {
            if
(xmlldoc.SelectSingleNode("configuration").ChildNodes.Count ==
selected_element)
            {
                foreach (XmlNode childNode_1 in
childNode.ChildNodes)
                {
                    if (childNode_1.InnerText ==
(string)listBox1.SelectedItem)
                    {
                        childNode.RemoveChild(childNode_1);
                        xmlldoc.Save("settingsdata.xml");
                    }
                }
            }
        }
        listBox1.Items.Remove(listBox1.SelectedItem);
    }
}

private void button3_Click(object sender, EventArgs e)
{
    //удаление элемента настройки
    string selected_element =
(string)comboBox1.SelectedItem;

```

```

        XmlNode NodeGen =
xmldoc.SelectSingleNode("configuration");
        foreach (XmlNode childNode in NodeGen.ChildNodes)
            if
(selected_element)
            {
                NodeGen.RemoveChild(childNode);
                xmldoc.Save("settingsdata.xml");
            }
        comboBox1.Items.Clear();
        foreach (XmlNode childNode in NodeGen.ChildNodes)

comboBox1.Items.Add(childNode.Attributes.GetNamedItem("name").Value);
    }

    private void button6_Click(object sender, EventArgs e)
    {
        if (listBox1.SelectedItems.Count != 0)
        {
            string selected_element =
(string)comboBox1.SelectedItem;

            XmlNode NodeGen =
xmldoc.SelectSingleNode("configuration");
            foreach (XmlNode childNode in
NodeGen.ChildNodes)
            {
                if
(selected_element)
                {
                    foreach (XmlNode childNode_1 in
childNode.ChildNodes)
                        if (childNode_1.InnerText ==
(string)listBox1.SelectedItem)
                        {
                            childNode.RemoveChild(childNode_1);
                            xmldoc.Save("settingsdata.xml");
                        }
                }
            }
            textBox2.Text = (string)listBox1.SelectedItem;
            listBox1.Items.Remove(listBox1.SelectedItem);
        }
    }
}

```

Приложение Б

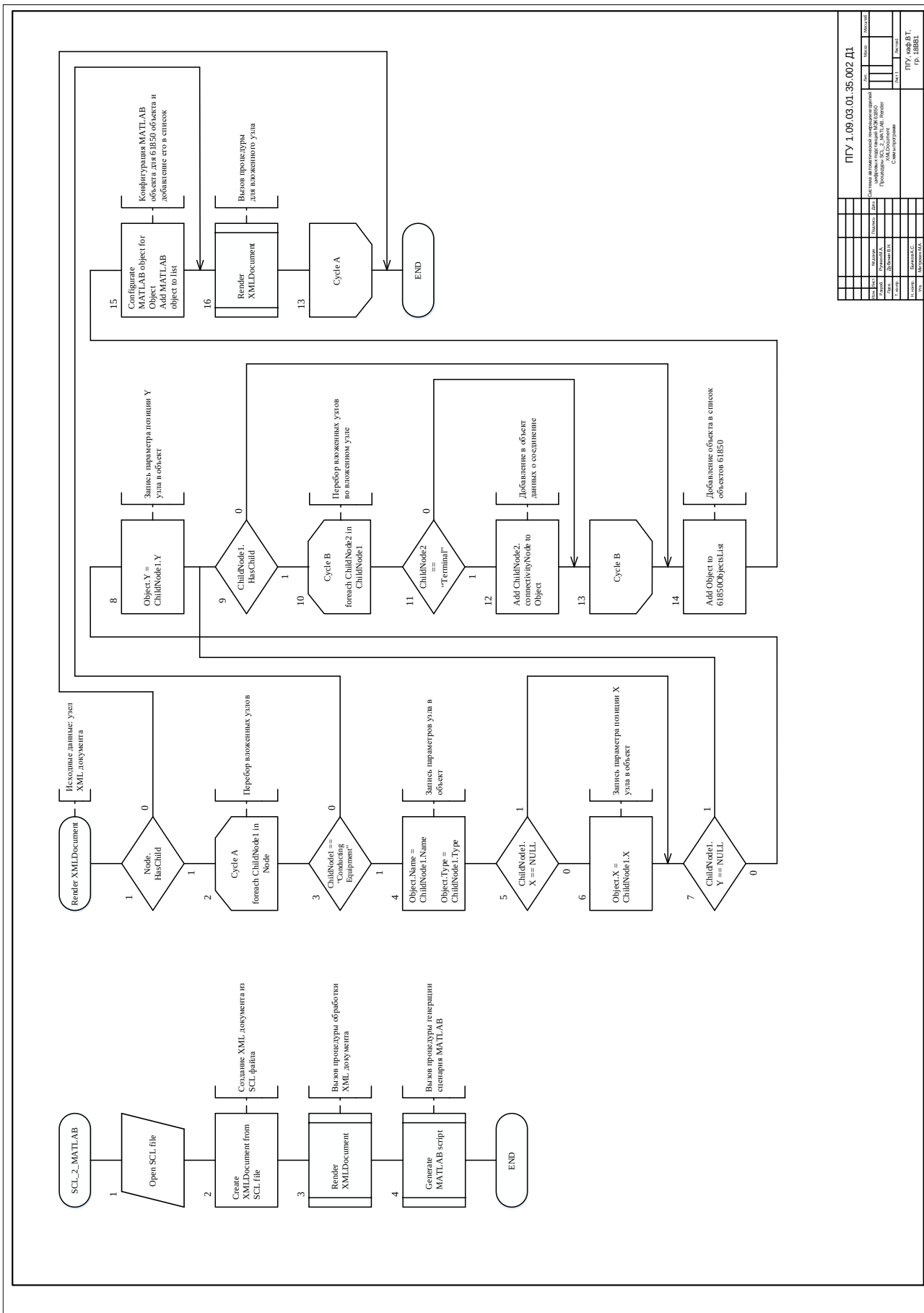
(Обязательное)

Система автоматической генерации моделей цифровых подстанций

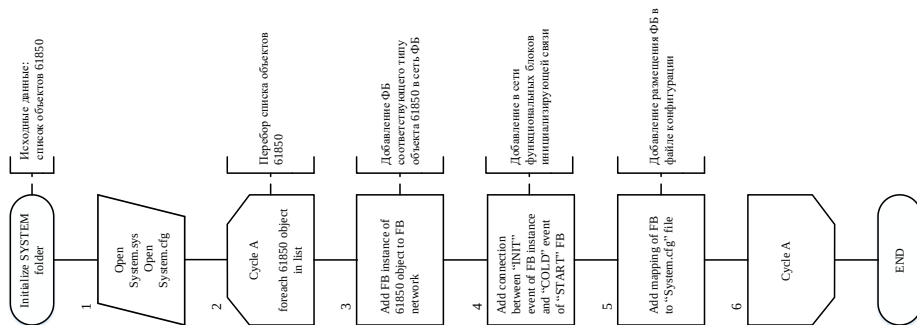
МЭК61850

Схемы программ

					<i>ПГУ 1.09.03.01.35.001 ПЗ</i>	Лист
Изм.	Лист	№ докум	Подпись	Дата		97



ПГУ 1.09.03.01.35.002 Д1									
№ п/п	№ документа	Дата	Исполнитель	Проверенный	Содержание	Содержание	Содержание	Содержание	Содержание
1	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
2	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
3	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
4	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
5	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
6	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
7	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
8	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
9	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
10	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
11	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
12	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
13	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
14	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
15	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
16	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
17	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
18	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
19	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.
20	1.09.03.01.35.002 Д1	2019.09.03	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.	И.И.И.



ПГУ 1.09.03.01.35.005 Д1									
Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.
Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.
Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.
Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.
Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.
Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.
Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.
Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.
Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.	Исполн.

Приложение В

(Обязательное)

Система автоматической генерации моделей цифровых подстанций

МЭК61850

Презентация

					<i>ПГУ 1.09.03.01.35.001 ПЗ</i>	Лист
Изм.	Лист	№ докум	Подпись	Дата		102

СИСТЕМА АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ МОДЕЛЕЙ ЦИФРОВЫХ ПОДСТАНЦИЙ МЭК 61850

Выполнил: студент группы 18BB1

Ручкин М.А.

Руководитель: Дубинин В.Н.



АКТУАЛЬНОСТЬ И МОТИВАЦИЯ

- Цифровые подстанции МЭК 61850 – компонент интеллектуальных систем электроснабжения Smart Grid, представляющий собой киберфизические системы
- Киберфизические системы являются атрибутом Индустрии 4.0
- Разработка, ввод в эксплуатацию и эксплуатацию таких сложных систем облегчается с помощью цифровых двойников, которые представляют собой имитационные модели систем, вычисляющие параметры текущего состояния с учетом фактической истории изменения, режима и условий работы систем
- Разработка цифровых двойников, несмотря на их большую полезность, представляет собой сложную инженерную задачу
- Программных систем для генерации цифровых двойников цифровых подстанций МЭК61850 не существует
- Цель работы – разработка методик и программной системы автоматической генерации MATLAB/SIMULINK- и функционально-блочных моделей на основе SCL-описаний цифровых подстанций МЭК 61850



Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

103

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

- разработана методика автоматической генерации MATLAB/SIMULINK моделей цифровых подстанций МЭК61850
- разработана методика автоматической генерации функционально-блочных моделей распределенного управления цифровых подстанций МЭК 61850 на основе стандарта МЭК 61499
- разработана программная система для поддержки методик автоматической генерации



SCL - ОТПРАВНАЯ ТОЧКА ПРОЕКТИРОВАНИЯ ЦИФРОВЫХ ПОДСТАНЦИЙ МЭК 61850

SCL описание цифровой подстанции представлено в исходном файле с расширением scd (файл для описания конфигурации подстанции). Данный файл представляет собой XML файл, но с главным тегом SCL, внутри которого и производится описание конфигурации подстанции.



```
<ConductingEquipment
  name="CBR1"
  xxy:x="22"
  type="CBR">
  <Terminal
    bayName="01"
    cNodeName="L1"
    substationName="МПС"
    voltageLevelName="3"
    connectivityNode="МПС/3/01/L1"/>
  </ConductingEquipment>
<ConductingEquipment
  name="CBR2"
  xxy:x="26"
  type="CBR">
  <Terminal
    bayName="01"
    cNodeName="L1"
    substationName="МПС"
    voltageLevelName="3"
    connectivityNode="МПС/3/01/L1"/>
  </ConductingEquipment>
```



Изм.	Лист	№ докум	Подпись	Дата

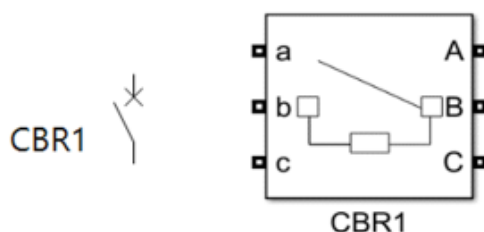
ПГУ 1.09.03.01.35.001 ПЗ

Лист

104

ПРЕДСТАВЛЕНИЕ ЭЛЕМЕНТОВ ЦИФРОВЫХ ПОДСТАНЦИЙ В ВИДЕ MATLAB/SIMULINK МОДЕЛЕЙ

Выключатель CBR из стандарта МЭК 61850 – трехфазный выключатель из библиотеки MATLAB



```
% Name of the model
fname = 'SCD_2_MATLAB';

% Create the system
new_system(fname);
add_block(...
'powerlib/Elements/Three-Phase Breaker',...
[gcs,'/CBR1'],...
'InitialState','open',...
'Position',[240 100 270 130]);

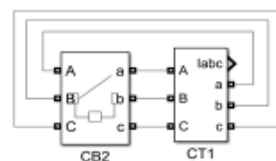
% Set a couple of model parameters
%to eliminate warning messages
set_param(gcs,...
'Solver','FixedStepDiscrete',...
'FixedStep','0.1');

% Save the model
save_system(fname);
```

МЕТОДИКА ГЕНЕРАЦИИ MATLAB/SIMULINK МОДЕЛИ

1. Построение модели
2. Размещение компонентов
3. Определение атрибутов компонента
4. Установление межкомпонентных связей
5. Определение атрибутов "Решателя" и сохранение модели

```
function SCD_2_MATLAB
fname = 'SCD_2_MATLAB';
% Check if the file already exists and delete it if it does
if exist(fname,'file')==4
% If it does then check whether it's open
if bIsLoaded(fname)
% If it is then close it (without saving)
close_system(fname,0)
end
% delete the file
delete(fname,'s');
end
% Create the system
new_system(fname);
% Add blocks
add_block('powerlib/Measurements/Three-Phase V-I Measurement',[gcs,'/CT1']...
'VoltageMeasurement','no',...
'CurrentMeasurement','yes');
add_block('powerlib/Elements/Three-Phase Breaker',[gcs,'/CB2']...
'InitialState','open');
add_line(gcs,'CB2/RConn1','CT1/LConn1');
add_line(gcs,'CB2/RConn2','CT1/LConn2');
add_line(gcs,'CB2/RConn3','CT1/LConn3');
add_line(gcs,'CT1/RConn1','CB2/LConn1');
add_line(gcs,'CT1/RConn2','CB2/LConn2');
add_line(gcs,'CT1/RConn3','CB2/LConn3');
% Solver, FixedStepDiscrete,...
'Solver','FixedStepDiscrete',...
'FixedStep','0.1');
% Save the model
save_system(fname);
```



СООТВЕТСТВИЯ ТИПОВ ОБОРУДОВАНИЯ СТАНДАРТА МЭК 61850 И ОБЪЕКТОВ MATLAB

Тип оборудования МЭК 61850	Значение	MATLAB объект
CBR	защитный выключатель	Three-Phase Breaker
CTR	измеритель тока	Three-Phase V-I Measurement
IFL	заземление	Ground
DIS	выключатель	Three-Phase Breaker
CAP	конденсатор	Three-Phase RLC Branch
BAT	батарея	Battery
CON (AC/DC)	преобразователь переменного тока в постоянный	CCCV Battery Charger
CON (DC/AC)	преобразователь постоянного тока в переменный	Inverter (Three-Phase)
PSH	нагрузка	RLC Branch

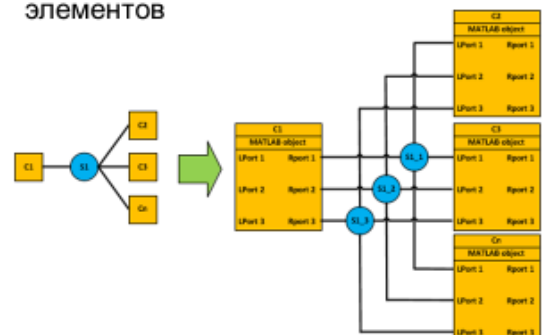


ПРАВИЛА ГРАФОВОЙ ТРАНСФОРМАЦИИ МОДЕЛЕЙ ЦИФРОВЫХ ПОДСТАНЦИЙ В МОДЕЛИ MATLAB/SIMULINK

Правило для двух связанных элементов



Правило для множества связанных элементов



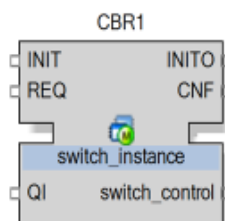
ПРЕДСТАВЛЕНИЕ ЭЛЕМЕНТОВ ЦИФРОВЫХ ПОДСТАНЦИЙ В ВИДЕ ФУНКЦИОНАЛЬНЫХ БЛОКОВ СТАНДАРТА МЭК 61499

Выключатель МЭК61850

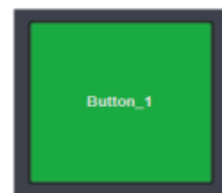
CBR1



Интерфейс ФБ
выключателя



Человеко – машинный
интерфейс ФБ выключателя



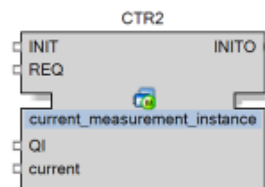
ПРЕДСТАВЛЕНИЕ ЭЛЕМЕНТОВ ЦИФРОВЫХ ПОДСТАНЦИЙ В ВИДЕ ФУНКЦИОНАЛЬНЫХ БЛОКОВ СТАНДАРТА МЭК 61499

Измеритель тока
МЭК 61850

CTR2



Интерфейс ФБ
измерителя тока



Человеко – машинный
интерфейс ФБ измерителя
тока



Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

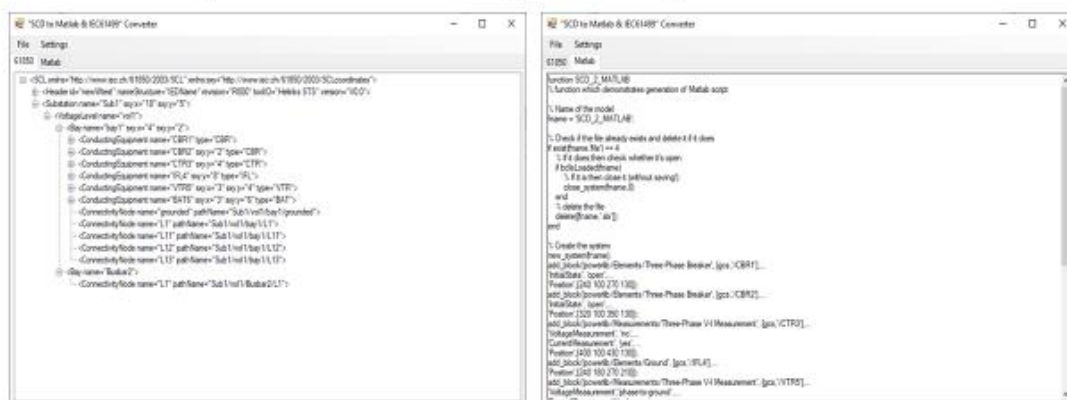
107

МЕТОДИКА ГЕНЕРАЦИИ ФУНКЦИОНАЛЬНО БЛОЧНОЙ МОДЕЛИ

1. Создание функциональных блоков для каждого типа оборудования МЭК61850
2. Создание человеко-машинного интерфейса функциональных блоков
2. Размещение экземпляров функциональных блоков в системе ФБ в соответствии с моделью цифровой подстанции МЭК 61850



ИНТЕРФЕЙС ПРОГРАММНОЙ СИСТЕМЫ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ МОДЕЛЕЙ ЦИФРОВЫХ ПОДСТАНЦИЙ МЭК 61850



Изм.	Лист	№ докум	Подпись	Дата

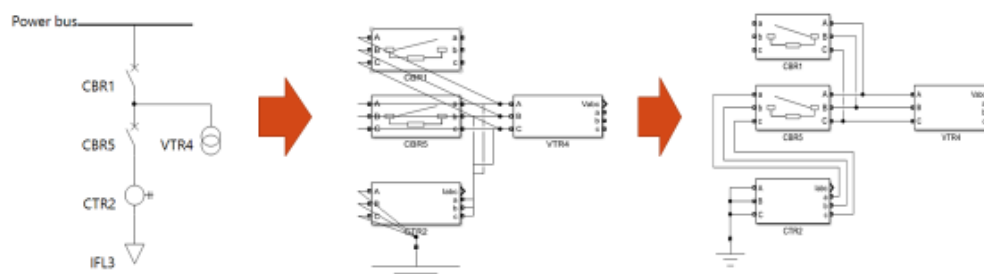
ПГУ 1.09.03.01.35.001 ПЗ

Лист

108

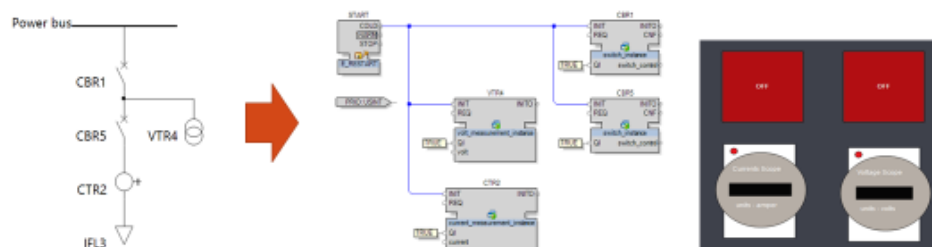
ТЕСТИРОВАНИЕ ГЕНЕРАЦИИ MATLAB/SIMULINK МОДЕЛИ ПРОГРАММНОЙ СИСТЕМОЙ

На основе фрагмента электроэнергетической сети стандарта
МЭК61850 была сгенерирована MATLAB/SIMULINK модель



ТЕСТИРОВАНИЕ ГЕНЕРАЦИИ ФУНКЦИОНАЛЬНО БЛОЧНОЙ МОДЕЛИ ПРОГРАММНОЙ СИСТЕМОЙ

На основе фрагмента электроэнергетической сети стандарта
МЭК61850 была сгенерирована функционально блочная модель
и человеко-машинный интерфейс для взаимодействия с ней



Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

109

СПАСИБО ЗА ВНИМАНИЕ



Изм.	Лист	№ докум	Подпись	Дата

ПГУ 1.09.03.01.35.001 ПЗ

Лист

110